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

DEPARTMENT OF ELECTRONICS AND COMMMUNICATION ENGINEERING

INDUSTRIAL INTERNET OF THINGS – SECA4005

UNIT – I Introduction to Industrial Internet

INTRODUCTION TO INDUSTRIAL INTERNET

Innovation and IIoT – Intelligent Devices – Industrial Internet – Health care –Oil and Gas Industry – Smart Office – Logistics – IoT Innovations in Retail.

The industrial internet of things (IIoT) is the use of smart sensors and actuators to enhance manufacturing and industrial processes. Also known as the industrial internet or Industry 4.0, IIoT uses the power of smart machines and real-time analytics to take advantage of the data that "dumb machines" have produced in industrial settings for years. The driving philosophy behind IIoT is that smart machines are not only better than humans at capturing and analyzing data in real time, but they're also better at communicating important information that can be used to drive business decisions faster and more accurately.

Connected sensors and actuators enable companies to pick up on inefficiencies and problems sooner and save time and money, while supporting business intelligence efforts. In manufacturing, specifically, IIoT holds great potential for quality control, sustainable and green practices, supply chain traceability, and overall supply chain efficiency. In an industrial setting, IIoT is key to processes such as predictive maintenance (PdM), enhanced field service, energy management and asset tracking.

Working of IIoT

IIoT is a network of intelligent devices connected to form systems that monitor, collect, exchange and analyze data. Each industrial IoT ecosystem consists of:

- connected devices that can sense, communicate and store information about themselves;
- public and/or private data communications infrastructure;
- analytics and applications that generate business information from raw data;
- storage for the data that is generated by the IIoT devices; and
- people.

These edge devices and intelligent assets transmit information directly to the data communications infrastructure, where it's converted into actionable information on how a certain piece of machinery is operating. This information can be used for predictive maintenance, as well as to optimize business processes.

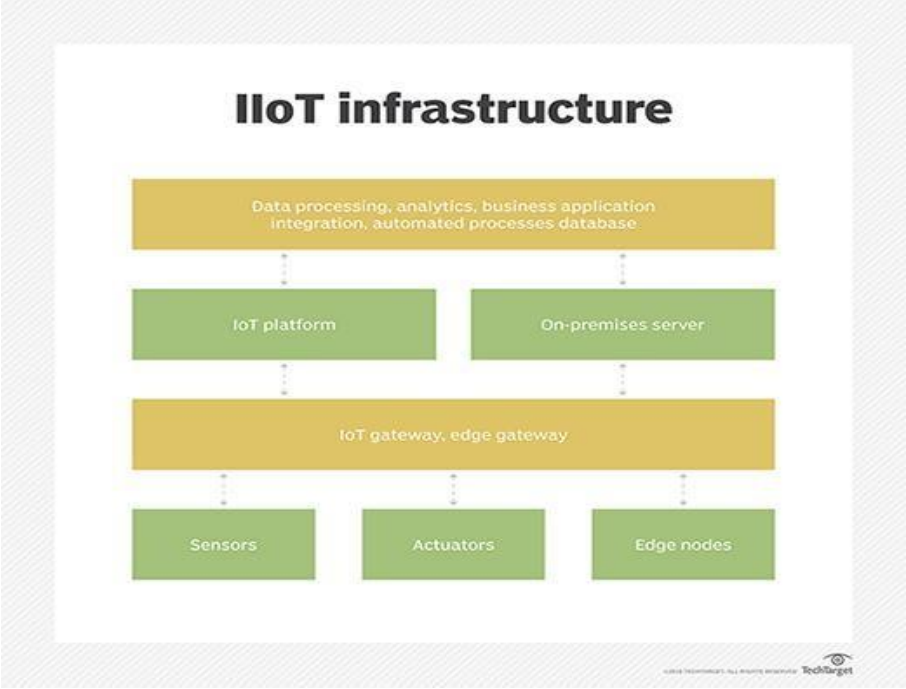


Fig.1.1 : IIoT infrastructure

IIoT utilization in Industry

There are countless industries that make use of IIoT. One example is the automotive industry, which uses IIoT devices in the manufacturing process. The automotive industry extensively uses industrial robots, and IIoT can help proactively maintain these systems and spot potential problems before they can disrupt production.

The agriculture industry makes extensive use of IIoT devices, too. Industrial sensors collect data about soil nutrients, moisture and more, enabling farmers to produce an optimal crop.

The oil and gas industry also uses industrial IoT devices. Some oil companies maintain a fleet of autonomous aircraft that can use visual and thermal imaging to detect potential problems in pipelines. This information is combined with data from other types of sensors to ensure safe operations.

Benefits of IIoT

One of the top touted benefits of IIoT devices used in the manufacturing industry is that they enable predictive maintenance. Organizations can use real-time data generated from IIoT systems to predict when a machine will need to be serviced. That way, the necessary maintenance can be performed before a failure occurs. This can be especially beneficial on a production line, where the failure of a machine might result in a work stoppage and huge costs. By proactively addressing maintenance issues, an organization can achieve better operational efficiency.

Another benefit is more efficient field service. IIoT technologies help field service technicians identify potential issues in customer equipment before they become major issues, enabling techs to fix the problems before they inconvenience customers. These technologies might also provide field service technicians with information about which parts they need to make a repair. That way, the technician has the necessary parts with them when making a service call.

Asset tracking is another IIoT perk. Suppliers, manufacturers and customers can use asset management systems to track the location, status and condition of products throughout the supply chain. The system sends instant alerts to stakeholders if the goods are damaged or at risk of being damaged, giving them the chance to take immediate or preventive action to remedy the situation.

IIoT also allows for enhanced customer satisfaction. When products are connected to the internet of things, the manufacturer can capture and analyze data about how customers use their products, enabling manufacturers and product designers to build more customer-centric product roadmaps.

IIoT also improves facility management. Manufacturing equipment is susceptible to wear and tear, which can be exacerbated by certain conditions in a factory. Sensors can monitor vibrations, temperature and other factors that might lead to suboptimal operating conditions.

IIoT security

Early on, manufacturers created IoT devices with little regard for security, resulting in a perception that IoT devices are inherently insecure. Given the similarities between IoT and IIoT devices, it's worth considering whether it's safe to use IIoT devices.

As with any other connected device, IIoT devices must be evaluated on a device-by-device basis. It's entirely possible that one manufacturer's device is secure while another isn't. Even so, security is a bigger priority among device manufacturers than ever before.

In 2014, several technology companies including AT&T, Cisco, General Electric, IBM and Intel came together to form the Industrial Internet Consortium (IIC). Although this group's primary objective is to accelerate the adoption of IIoT and related technologies, it's making security a priority, even going so far as to form a security working group. The IIC's other working groups include Technology, Liaison, Marketing, Industry and Digital Transformation.

Risks and challenges of IIoT

The biggest risks associated with IIoT use pertain to security. It's relatively common for IIoT devices to continue using default passwords, even after they have been placed into production. Similarly, many IIoT devices transmit data as clear text. These conditions would make it relatively easy for an attacker to intercept the data coming from an IIoT device. Similarly, an attacker could take over an insecure IIoT device and use it as a platform for launching an attack against other network resources. Security is a big challenge for those who are responsible for an organization's IIoT devices, but so, too, is device management. As an organization adopts more and more IIoT devices, it will become increasingly important to adopt an effective device management strategy. More specifically, organizations must be able to positively identify IIoT devices to prevent the use of rogue devices. Establishing a means of identifying each individual device is also crucial for tasks such as replacing a failed device or performing a device refresh.

Patch management presents another big challenge regarding IIoT devices. It's becoming increasingly common for device manufacturers to issue periodic firmware updates. Organizations must have an efficient means of checking devices to see if they have the latest

firmware installed and deploying new firmware if necessary. Additionally, such a tool must adhere to the organization's established maintenance schedule so as to not disrupt operations.

Difference between IoT and IIoT

Although IoT and IIoT have many technologies in common, including cloud platforms, sensors, connectivity, machine-to-machine communications and data analytics, they are used for different purposes.

IoT applications connect devices across multiple verticals, including agriculture, healthcare, enterprise, consumer and utilities, as well as government and cities. IoT devices include smart appliances, fitness bands and other applications that generally don't create emergency situations if something goes amiss.

IIoT applications, on the other hand, connect machines and devices in such industries as oil and gas, utilities and manufacturing. System failures and downtime in IIoT deployments can result in high-risk situations, or even life-threatening ones. IIoT applications are also more concerned with improving efficiency and improving health or safety, versus the user-centric nature of IoT applications.

IIoT applications and examples

In a real-world IIoT deployment of smart robotics, ABB, a power and robotics firm, uses connected sensors to monitor the maintenance needs of its robots to prompt repairs before parts break.

Likewise, commercial jetliner maker Airbus has launched what it calls the *factory of the future*, a digital manufacturing initiative to streamline operations and boost production. Airbus has integrated sensors into machines and tools on the shop floor and outfitted employees with wearable tech -- e.g., industrial smart glasses -- aimed at cutting down on errors and enhancing workplace safety.

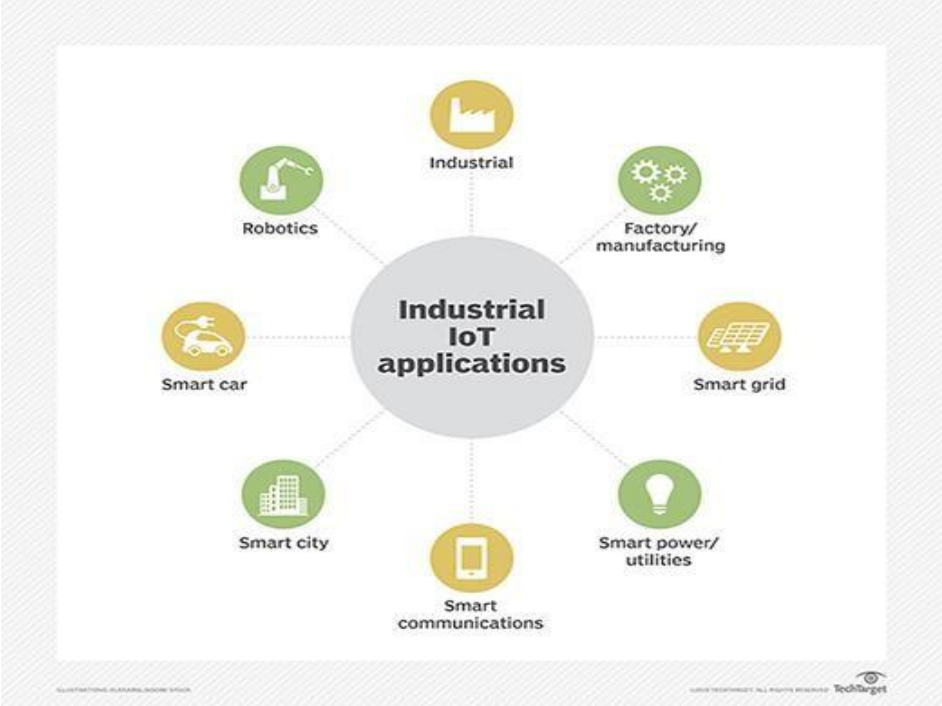


Fig. 1.2 : IIoT Applications

IIoT vendors

There are several vendors with IIoT platforms, including:

- ABB Ability. An IIoT company specializing in connectivity, software and machine intelligence.
- Aveva Wonderware. A company that develops human-machine interface (HMI) and IoT edge platforms for OEMs (original equipment manufacturers) and end users.
- Axzon. An IIoT company focusing on smart automotive manufacturing, predictive maintenance and cold chain.
- Cisco IoT. A networking company offering platforms for network connectivity, connectivity management, data control and exchange, and edge computing.
- Fanuc Field System. A company that has developed a platform for connecting various generations, makes and models of industrial IoT equipment.
- Linx Global Manufacturing. A product development and manufacturing company offering custom IIoT, application and data management platforms.

- MindSphere by Siemens. An industrial IoT solution based around artificial intelligence (AI) and advanced analytics.
- Plataine. An IIoT company specializing in using AI to generate actionable insights in manufacturing.
- Predix by GE. A platform for connecting, optimizing and scaling digital industrial applications.

IIoT and 5G

5G is the emerging standard for mobile networks. It has been specifically designed to deliver fast data throughput speeds with low latency. 5G will support download speeds of up to 20 Gbps (gigabits per second) with sub-millisecond latency.

The emergence of 5G will likely affect the use of IIoT devices in two main ways. First, 5G's high throughput and low latency will make it possible for devices to share data in real time. Previously, this was only possible when the devices were located on private networks with high-speed connectivity. This real-time connectivity will support use cases such as driverless cars and smart cities.

The other way 5G will affect IIoT adoption is that it will likely result in device proliferation. Industrial operations might use thousands of 5G connected devices. 5G's high speed and low latency also means we'll likely see IIoT devices used in remote sites whose lack of high-speed connectivity previously made IIoT use impractical.

Future of IIoT

The future of IIoT is tightly coupled with a trend known as Industry 4.0. Industry 4.0 is, essentially, the fourth Industrial Revolution.

Industry 1.0 was the first Industrial Revolution and occurred in the late 1700s as companies began to use water-powered or steam-powered machines in manufacturing. Industry 2.0 started at the beginning of the 20th century and was brought about by the introduction of electricity and assembly lines. Industry 3.0 occurred in the latter part of the 20th century and was tied to the use of computers in the manufacturing process.

IoT Intelligent Devices

IoT devices are the nonstandard computing devices that connect wirelessly to a network and have the ability to transmit data, such as the many devices on the internet of things Connected devices are part of an ecosystem in which every device talks to other related devices in an environment to automate home and industry tasks. The devices can be categorized into three main groups: consumer, enterprise and industrial. Consumer connected devices include smart TVs, smart speakers, toys, wearables and smart appliances.

Enterprise IoT devices are edge devices designed to be used by a business. There are a huge variety of enterprise IoT devices available. These devices vary in capability but tend to be geared toward maintaining a facility or improving operational efficiency. Some options include smart locks, smart thermostats, smart lighting and smart security. Consumer versions of these technologies exist as well.

In the enterprise, smart devices can help with meetings. Smart sensors located in a conference room can help an employee locate and schedule an available room for a meeting, ensuring the proper room type, size and features are available. When meeting attendees enter the room, the temperature will adjust according to the occupancy, the lights will dim as the appropriate PowerPoint loads on the screen and the speaker begins his or her presentation.

Industrial IoT (IIoT) devices are designed to be used in factories or other industrial environments. Most IIoT devices are sensors used to monitor an assembly line or other manufacturing process. Data from various types of sensors is transmitted to monitoring applications that ensure key processes are running optimally. These same sensors can also prevent unexpected downtime by predicting when parts will need to be replaced.

If a problem occurs, the system might be able to send a notification to a service technician informing them what is wrong and what parts they will need to fix the problem. This can save the technician from coming on site to diagnose the problem and then having to travel to a warehouse to get the part needed to fix the problem.

Working of IoT devices

IoT devices vary in terms of functionality, but IoT devices have some similarities in how they work. First, IoT devices are physical objects designed to interact with the real world in some way. The device might be a sensor on an assembly line or an intelligent security camera. In either case, the device is sensing what's happening in the physical world.

The device itself includes an integrated CPU, network adapter and firmware, which is usually built on an open source platform. In most cases, IoT devices connect to a Dynamic Host Configuration Protocol server and acquire an IP address that the device can use to function on the network. Some IoT devices are directly accessible over the public internet, but most are designed to operate exclusively on private networks.

Although not an absolute requirement, many IoT devices are configured and managed through a software application. Some devices, however, have integrated web servers, thus eliminating the need for an external application.

Once an IoT device has been configured and begins to operate, most of its traffic is outbound. A security camera, for example, streams video data. Likewise, an industrial sensor streams sensor data. Some IoT devices such as smart lights, however, do accept inputs.

Bitdefender BOX - IoT Security Solution

Security solution which blocks incoming threats and can scan all your devices for vulnerabilities. It protects all your IoT devices, even when you go out! Can act as a wireless router or go alongside your current one.

Google Home- Voice controller

Google Home, the connected voice controller from Google. Besides controlling your home it also comes with Google Assistant, helping with lists, translation, news, music, calendar and many many more.

Nest Cam- Indoor camera

Nest Cam Outdoor is the monitoring tool you've been waiting for. It brings all the benefits of modern streaming technology and a sleek design so you can watch your home from anywhere.

Mr. Coffee - Smart Coffeemaker

Mr. Coffee 10-Cup Smart Optimal Brew Coffeemaker makes it easy to schedule, monitor, and modify your brew from anywhere.

SmartMat - Intelligent Yoga Mat

The interactive yoga utility that helps you perfect your yoga training through real time pressure sensing technology and smart mobile interfacing.

TrackR bravo - Tracking Device

TrackR bravo is the coin-sized tracking device that locates your belongings in real time & notifies you their location, whether they are lost or misplaced. Great for wallets, keys, phones or pets.

Linquet - Bluetooth tracking sensor

Linquet is the cloud powered tracking device you can attach to anything. Link them to the app and share their location with those interested so nothing gets left behind.

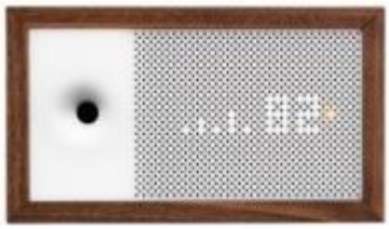
Nest Thermostat - Smart Thermostat

The Nest Thermostat learns what temperature you like and builds a schedule around yours. Also, it will send you an alert when the temperatures are threatening to ruin your belongings and appliances.

Portable Wi-Fi video camera

The smart home security camera that helps you keep home safe and stay connected - all from your smartphone. Comes with AES encryption and free private 24-hour secure cloud storage for your total peace of mind.

Smart Air Quality Monitor



Awair is the first complete device to let you communicate with your air. Awair analyzes your indoor air quality, learns your routines and can communicate with other home devices to help you achieve optimal air quality.

Navdy - Smart navigation system

Navdy combines a high quality projection display with voice and gesture controls to create a safer driving experience. Drivers no longer need to use their phone to navigate, communicate or control their music.

Smart Irrigation Controller

Manage your irrigation controller and save water with predictive schedules from anywhere using your smart device or web browser with Hydrawise web-based software.

Intelligent oven

- modern oven prepared to fit in every kitchen in order to satisfy even the most exquisite tastes. It saves a lot of the time you'd normally spend cooking when connected to your phone.

Blood Pressure Monitor

Wireless Blood Pressure Monitor makes it easy to check your blood pressure & heartrate, anytime and anywhere with an instant feedback and access to all your readings.

Smart baby monitoring

Smart Baby Movement Monitor uses a smart, washable crib sheet to show parents their baby's sleep activity and movement on their smartphone or tablet.

Home Intelligence Sensors

sensors can be thermostats, burglar alarms, leak detectors - or anything else you can think of.

Gas and CO detector

No matter where you are, if Kepler detects danger, it alerts you on your smartphone, while simultaneously flashing its lights and sounding an alarm.

IoT for Healthcare

Before Internet of Things, patients' interactions with doctors were limited to visits, and tele and text communications. There was no way doctors or hospitals could monitor patients' health continuously and make recommendations accordingly.

Internet of Things (IoT)-enabled devices have made remote monitoring in the healthcare sector possible, unleashing the potential to keep patients safe and healthy, and empowering physicians to deliver superlative care. It has also increased patient engagement and satisfaction as interactions with doctors have become easier and more efficient. Furthermore, remote monitoring of patient's health helps in reducing the length of hospital stay and prevents re-admissions. IoT also has a major impact on reducing healthcare costs significantly and improving treatment outcomes.

IoT is undoubtedly transforming the healthcare industry by redefining the space of devices and people interaction in delivering healthcare solutions. IoT has applications in healthcare that benefit patients, families, physicians, hospitals and insurance companies.

IoT for Patients - Devices in the form of wearables like fitness bands and other wirelessly connected devices like blood pressure and heart rate monitoring cuffs, glucometer etc. give patients access to personalized attention. These devices can be tuned to remind calorie count, exercise check, appointments, blood pressure variations and much more.

IoT has changed people's lives, especially elderly patients, by enabling constant tracking of health conditions. This has a major impact on people living alone and their families. On any disturbance or changes in the routine activities of a person, alert mechanism sends signals to family members and concerned health providers.

IoT for Physicians - By using wearables and other home monitoring equipment embedded with IoT, physicians can keep track of patients' health more effectively. They can track patients' adherence to treatment plans or any need for immediate medical attention. IoT enables healthcare professionals to be more watchful and connect with the patients proactively. Data collected from IoT devices can help physicians identify the best treatment process for patients and reach the expected outcomes.

IoT for Hospitals - Apart from monitoring patients' health, there are many other areas where IoT devices are very useful in hospitals. IoT devices tagged with sensors are used for tracking real time location of medical equipment like wheelchairs, defibrillators, nebulizers, oxygen pumps and other monitoring equipment. Deployment of medical staff at different locations can also be analyzed real time.

The spread of infections is a major concern for patients in hospitals. IoT-enabled hygiene monitoring devices help in preventing patients from getting infected. IoT devices also help in asset management like pharmacy inventory control, and environmental monitoring, for instance, checking refrigerator temperature, and humidity and temperature control.

IoT for Health Insurance Companies – There are numerous opportunities for health insurers with IoT-connected intelligent devices. Insurance companies can leverage data captured through health monitoring devices for their underwriting and claims operations. This data will enable them to detect fraud claims and identify prospects for underwriting. IoT devices bring transparency between insurers and customers in the underwriting, pricing, claims handling, and risk assessment processes. In the light of IoT-captured data-driven decisions in all operation processes, customers will have adequate visibility into underlying thought behind every decision made and process outcomes.

Insurers may offer incentives to their customers for using and sharing health data generated by IoT devices. They can reward customers for using IoT devices to keep track of their routine activities and adherence to treatment plans and precautionary health measures. This will help insurers to reduce claims significantly. IoT devices can also enable insurance companies to validate claims through the data captured by these devices.

Redefining Healthcare

The proliferation of healthcare-specific IoT products opens up immense opportunities. And the huge amount of data generated by these connected devices hold the potential to transform healthcare.

IoT has a four-step architecture that are basically stages in a process (See Figure 1). All four stages are connected in a manner that data is captured or processed at one stage and yields the value to the next stage. Integrated values in the process brings intuitions and deliver dynamic business prospects.

Step 1: First step consists of deployment of interconnected devices that includes sensors, actuators, monitors, detectors, camera systems etc. These devices collect the data.

Step 2: Usually, data received from sensors and other devices are in analog form, which need to be aggregated and converted to the digital form for further data processing.

Step 3: Once the data is digitized and aggregated, this is pre-processed, standardized and moved to the data center or Cloud.

Step 4: Final data is managed and analyzed at the required level. Advanced Analytics, applied to this data, brings actionable business insights for effective decision-making.

IoT is redefining healthcare by ensuring better care, improved treatment outcomes and reduced costs for patients, and better processes and workflows, improved performance and patient experience for healthcare providers.

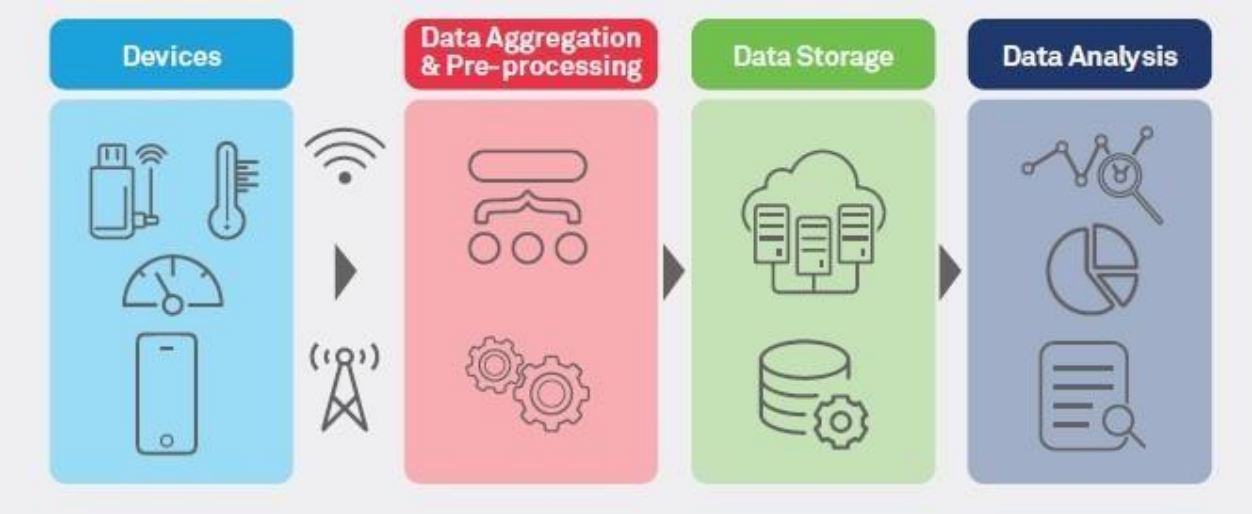


Fig.1.3 : IoT for Healthcare

The major advantages of IoT in healthcare include:

- Cost Reduction: IoT enables patient monitoring in real time, thus significantly cutting down unnecessary visits to doctors, hospital stays and re-admissions
- Improved Treatment: It enables physicians to make evidence-based informed decisions and brings absolute transparency
- Faster Disease Diagnosis: Continuous patient monitoring and real time data helps in diagnosing diseases at an early stage or even before the disease develops based on symptoms

- Proactive Treatment: Continuous health monitoring opens the doors for providing proactive medical treatment
- Drugs and Equipment Management: Management of drugs and medical equipment is a major challenge in a healthcare industry. Through connected devices, these are managed and utilized efficiently with reduced costs
- Error Reduction: Data generated through IoT devices not only help in effective decision making but also ensure smooth healthcare operations with reduced errors, waste and system costs

Healthcare IoT is not without challenges. IoT-enabled connected devices capture huge amounts of data, including sensitive information, giving rise to concerns about data security.

Implementing apt security measures is crucial. IoT explores new dimensions of patient care through real-time health monitoring and access to patients' health data. This data is a goldmine for healthcare stakeholders to improve patient's health and experiences while making revenue opportunities and improving healthcare operations. Being prepared to harness this digital power would prove to be the differentiator in the increasingly connected world.

IoT for OIL and Gas Industry

This high-level overview and architecture focuses what IoT can bring to the oil and gas industry, or most any field, and how to get started.

Deployment of IoT-based smart energy solutions results in better field communication, reduced costs of maintenance, real-time monitoring, digital oil field infrastructure, reduced power consumption, mine automation, greater safety and security of assets, and thus higher productivity.

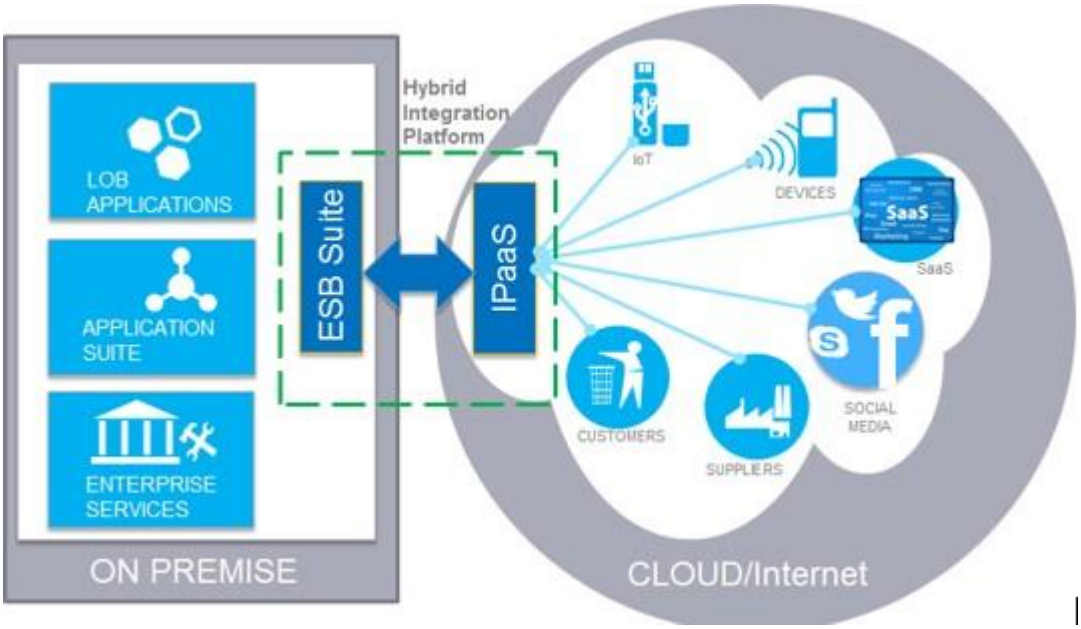


Fig.1.4 : IoT for Oil and Gas Industry

IoT will improve energy efficiency, remote monitoring and control of physical assets, and productivity through applications as diverse as home security to condition monitoring on the factory floor.

Operational Excellence

- Predictive maintenance
- Pipeline and equipment monitoring
- Location Intelligence
- Emissions monitoring and control and release management

Operations

- Real-time machine and sensor integration
- Real-time alerts
- Link to enterprise resource planning data to trigger maintenance workflow
- Plant dashboards and trend analysis
- Asset information network
- Fleet operations

MonitoringGlobal Reach

- IoT is removing the physical barriers so O&G companies helping reach broader target audiences and opening up new global business opportunities.

A typical reference architecture is depicted here for Oil and Gas Industry leveraging Microsoft Azure:

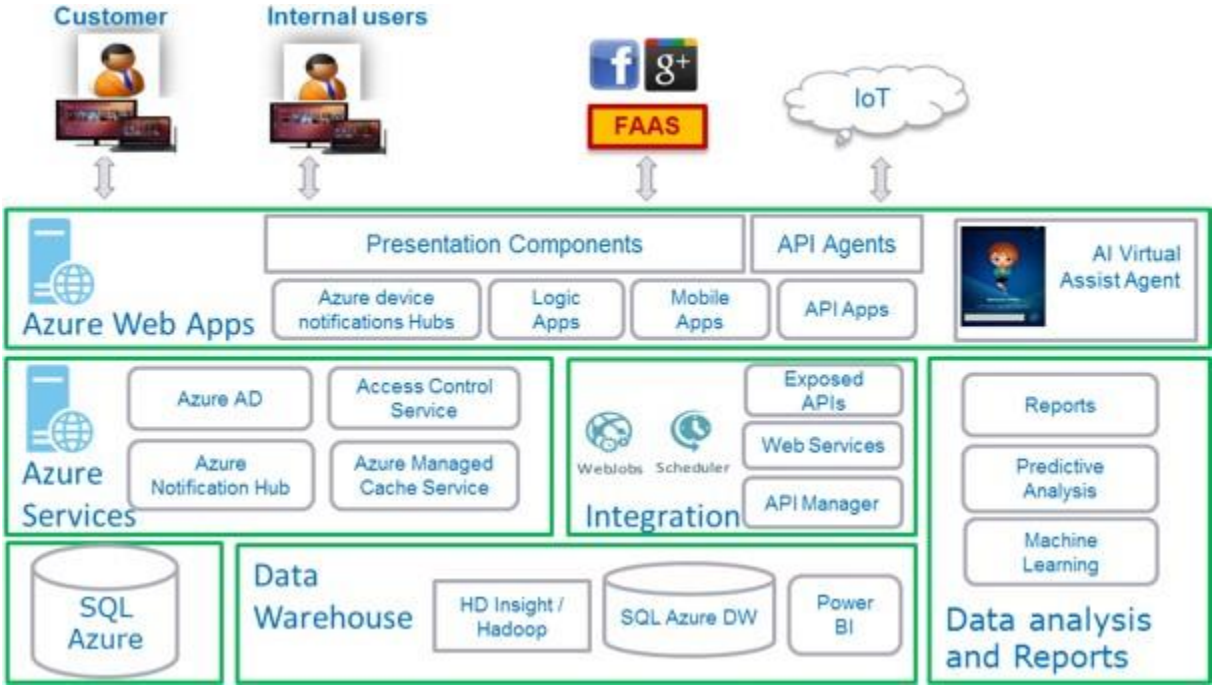


Fig.1.5 : Reference architecture for Oil and Gas Industry

Two important aspects that need to be considered when architecting an IoT solution are scalability and security. The IoT solution should be scalable to support unpredictable traffic surge while security is important at the device level to ensure it is hack-proofed. Azure IoT Hub

provides the reliability to secure the connection between device and cloud and vice-versa, but scalability has to be implemented at the architecture level.

- Device Management: After the device registers with the cloud gateway, it can send and receive the data to and from the hubs. It should have the device management feature to add, activate, deactivate, remove the device, and update the attributes of the device.
- Device Connectivity: There will be a huge amount of data that needs to be managed, with multiple messages being received in a second from a huge number of devices, which would result in 10s of thousands or possibly millions of messages a day. The platform should provide high-volume message ingestion using a single logical endpoint.
- Transformation and Storage: Once the messages arrive, the platform should provide a mechanism to select, transform, and route messages to various storage media for the purpose of archiving and staging for downstream processing.
- Analytics and Data Visualization: The value of collecting data in a continuous fashion is to build up a historical record for the purpose of performing analytics to gain business insights.
- Presentation and Action: The cloud solution should provide the ability to visualize the status of the messages in real time through tabular or graphical UI components. In addition, some messages may contain information of an alert status so the IoT solution must provide a mechanism for real-time notifications to actionable operation.
- Microsoft Azure IoT Suite is an enterprise-grade solution that lets you get started quickly through a set of extensible, preconfigured solutions that address common IoT scenarios, such as remote monitoring and predictive maintenance. These solutions are implementations of the IoT solution architecture described previously.
- The preconfigured solutions are complete, working, end-to-end solutions that include simulated devices to get you started, preconfigured Azure services such as Azure IoT Hub, Azure Event Hubs, Azure Stream Analytics, Azure Machine Learning, and Azure storage, and solution-specific management consoles. The preconfigured solutions contain proven, production-ready code that you can customize and extend to implement your own specific IoT scenarios.

Key Highlights

- HTML5-based UI to support a wider range of devices
- Modernized, highly intuitive, and easy to use UI with internationalization in compliance with customer branding guidelines
- Cloud-based, highly available, and scalable architecture to extend support for additional features in the future
- Cloud-based architecture to eliminate the constraints with increasing users and data growth in the future
- Standard REST-based integration architecture for future extensions and integrations with other on-premise or third-party systems
- Support for event notification and online monitoring
- Real-time/near real-time reporting and move towards self-service business intelligence
- Predictive analytics to improve business process efficiency
- Artificial Intelligence-based agents for learning and knowledge management
- Standards (SAML)-based federated authentication of users to accept the user identities from social networks, like Google, Microsoft, etc.

IoT Revolution in Oil and Gas Industry

- Oil and gas industry is a tech and asset-heavy sector. Malfunctions, incorrect measurements and even tiny mistakes in this field result in billions of dollars in losses and, sometimes, tragic events like Deepwater Horizon.
- Development of IoT for oil and gas industry helps solve different challenges in this field — reduce costly downtime, increase efficiency and safety on the premises and boost performance at every step of production.

Let's find out about these and many other benefits of IoT in this domain.

Challenges in the Oil and Gas Industry

Historically, this sector is associated with high turnover, enormous financial returns and great weight in the economy of a given country and the world in general. This description is valid today, however, things aren't always easy in this industry.

Executives in the oil and gas value chain constantly confront major challenges. For example**:**

Aging equipment and legacy systems

To make it clear, when we are talking about the equipment in this field, it implies powerful super-machines, huge drills, tankers and complex monitoring systems that perform crucial calculations to both maintain the performance and keep workers safe. They are rugged, work hard and require continuous monitoring and fast response to wear and other maintenance needs. Today, many wells rely on aging equipment and legacy monitoring systems. Upgrading them requires big money and manpower, though the downtime costs even more.

Hazardous environments

Another reason why maintaining wells and other parts of the oil and gas supply chain is challenging is the environment and accessibility. Often, deposits are found in remote offshore areas. Many oil-producing wells are built in the perilous northern seas, gas pipelines go through harsh zones like deserts and tundras. Difficult environment, accessibility and hazardous working conditions make any breakage or leak harder to contain and fix.

This makes the capabilities of Internet of Things in oil and gas industry particularly valuable.

Fig.1.6 : IoT revolution in Oil and Gas Industry

Innovation in Oiland Gas Industry

Today's technology has great potential not only to solve the challenges but also to enhance the performance in this field. Using IoT innovation for oil and gas sector is one of the approaches that receives the most attention and investment.

Sensors

Using the network of IoT sensors for oil and gas extraction and processing helps maintain ongoing control in the supply chain and quickly respond to changes. Sensor-based technology can be leveraged to monitor the pressure in the pipes, oversee the drilling process, machinery conditions and detect leakages. In this industry, the high speed of addressing issues usually translates to billions of dollars in savings.

Fig.1.7 : Sensors in Oil and Gas Industry

Smart algorithms

Smart algorithms cross-reference and analyze data and events registered and detected by diverse sensors. They create unique insights that help management make important decisions, for example, when exactly to start and stop drilling to avoid issues.

Predictive and preventive maintenance

Smart algorithms can predict when the conditions of expensive equipment change and it requires maintenance, either regular or urgent. It goes without saying that timely on-demand maintenance is more efficient than routine checks and acts as a guarantor of workers' safety.

Robots and drones

Among IoT devices for oil and gas supply chain, drones and robots play an important role. They enable efficient site exploration, ongoing data gathering and 3D mapping of landfills and can withstand hard conditions regular for drill sites.

Wearables

Wearables already boost efficiency and even save lives in this sector. Sensor-based suits, wristbands, smart glasses and helmets allow to continuously monitor the conditions of the workers which perform dangerous operations, seamlessly connect them with the base and even augment worker's capabilities providing timely advice, notification or warnings.

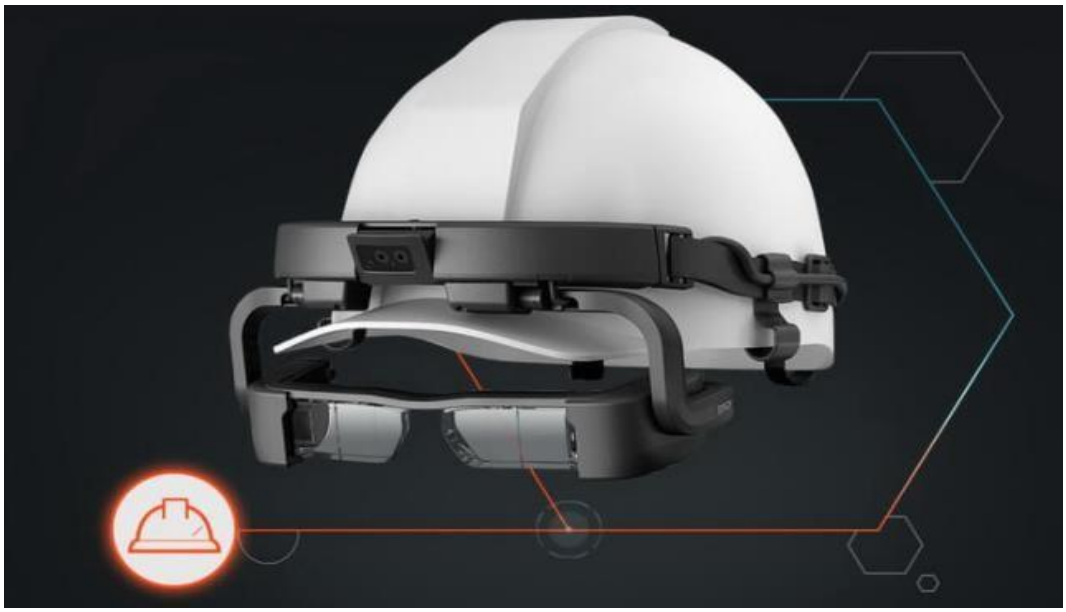


Fig. 1.8 : Smart Helmet in Oil and Gas Industry

Oil and Gas Monitoring Systems

IoT is inherently the first technology to think of when it comes to continuous monitoring and analysis of datum. In this respect, IoT applications in oil and gas industry for system monitoring, management and remote control make a big difference.

First of all, thanks to advancing sensor technology and increasing connectivity options, the management in the oil and gas industry can monitor anything in real-time from the changing sea bed topography, the chemical composition of the crude oil to the integrity of a gas pipeline and tanker fleet positioning. The amplified capabilities of IoT-based monitoring systems can be applied all along the oil and gas supply chain upstream, midstream and downstream.

Secondly, the development of IoT technology brings advanced data analytics and visualization tools to the table. Today, management uses convenient dashboards and can track operations and read measurements on PC or mobile from the comfort of one's office, so as respond to changes remotely using actuators and controls.

Advantages of IoT in Oil and Gas Industry

Remote monitoring which spares workers from going on-site and performing routine manual checks are not the only benefits of using IoT for oil and gas production and distribution.
Let's go through other advantages of IoT in oil and gas industry:

- Enable real-time equipment, fleet and environmental conditions monitoring and provide better transparency and control over processes.
- Allow for timely on-demand equipment maintenance and optimize related cost and effort.
- Ensure better worker safety and transfer risky on-site operations to robots and UAVs.
- Introduce automation, including automated leakage and breakage control.
- Reduce the negative environmental impact associated with oil and gas production and distribution.
- Optimize manpower and cut down on non-productive time and downtime.

IoT Smart Office

Smart office is a place equipped with the latest technology and devices that let people work faster, better and smarter. Such space is not just a new trend, it's a great modern way of thinking that brings a diversity of opportunities. IoT it is possible to do so many things that earlier were considered to be a futuristic tale. Here are just a few examples of what we can do in office if we have IoT solutions:

- Adjust the humidity and heating in office by using a smartphone.
- Use a smart scheduling system to never experience issues with booking a conference room.
- Benefit from smart sensors that will turn lightning and devices off when no one is in the office and make company more energy efficient.
- Improve security system by refusing from the old approach and passcodes and replacing them with smart locks and security cameras.
- Clean space during the lunch break thanks to smart vacuum and window cleaners that can be managed from a device.

And now let's take a look at the statistic that speaks louder than words and proves that smart IoT technologies are at the peak of their glory. Let's take a look at the table provided by Statistics that shows the number of IoT devices used worldwide.

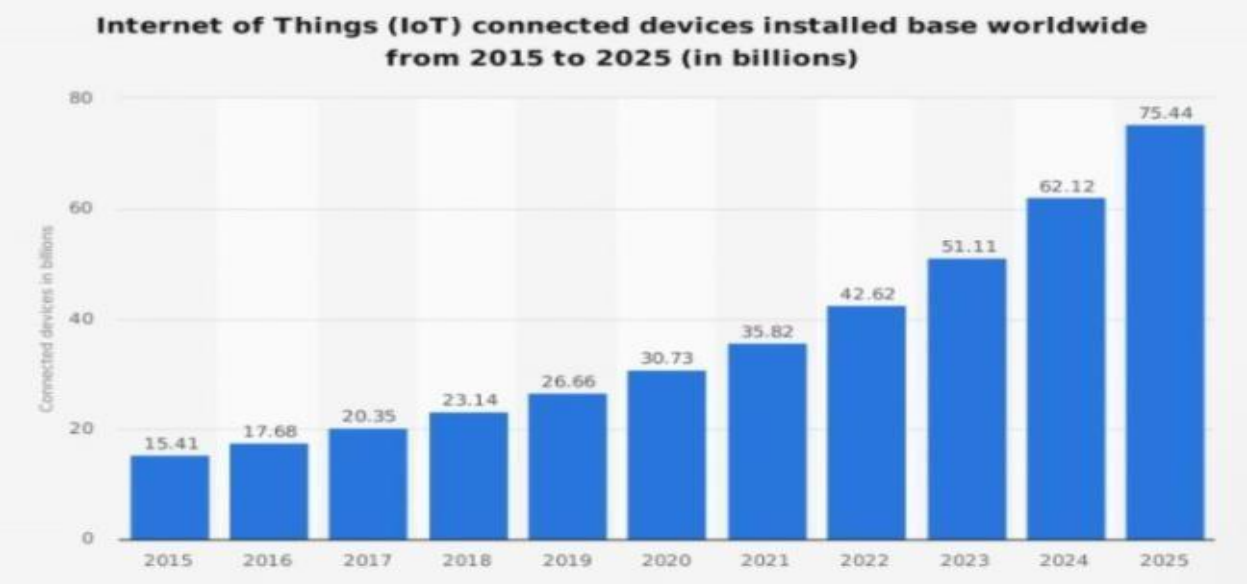


Fig.1.8 : Statistics of IoT Connected Devices

This tendency shows that more and more companies and households are going to adopt IoT technology and invest in smart projects. And this is not just because it is convenient to make everything connected and control it from a distance, the IoT adoption has way more benefits like:

- Decreased operational costs,
- Improved productivity,
- New business opportunities,
- Huge competitive advantage,
- Higher level of employee motivation and involvement.

Top 8 examples of IoT solutions for smart office

Digital transformation is crucial for offices of all sizes. It doesn't really matter whether you have 50 or 500 employees who are going to spend their working hours in office. What really matters is whether their work is efficient and whether company provide them with all necessities to boost their productivity. So to inspire change in office, we have collected several great examples of smart IoT solutions that can be actively used in working space or become the model for own custom development. And before proceed to them, just take a look at the video showing how smart office concept performs in action.

Fig. 1.9 : IoT Smart Office

Smart Business Assistant

We all use Alexa or Siri, but let's be honest, they are great for fun and domestic use but lack many things to become full time business assistant. And to have one these days is a necessity. Technologies used by Alexa for Business include numerous skill APIs (smart home, music, video, education skills), and text-to-speech.

IoT tagging for tracking devices

Although in many offices the heads encourage the usage of own smart devices for work, in others it is still preferable to use office owned devices. But this requires their constant tracking – who took what, and where that device is now. Let's take a real time tracker called Aruba as an example. If there is a smart solution of a kind in company, then we can monitor and pinpoint where this or that device is. It takes a couple of seconds to find what we need within office space.

Smart thermostats

These were the first IoT devices that found their place in our households, but they can be no less useful in office. The ability of smart thermostats to dynamically adjust the temperature in the office can make the working environment more comfortable and let cut costs spent on inefficient and expensive climate control systems. Smart thermostats can be controlled remotely and also with the help of the voice assistance, and there is a huge variety of them available on the market.

Tools for environment monitoring

Twine is a small but extremely intelligent IoT device that is responsible for the advanced diagnostics in household or office space. It can monitor internal and inside temperature, humidity, and even provide advanced reports after analyzing the collected data. We can also use the additional sensors of this device to set a web app to listen to vibrations, noises, etc. and send notifications via email, SMS or even Twitter. So the app can let know when someone knocked on door, when some device stopped working, etc. Here is how the device looks.

Intelligent lightning

The smart bulbs can be connected to the devices of your employees to learn their schedules and turn the light on whenever it is needed. Also the employees will be able to adjust the brightness, color and balance of light during the day to feel comfortable and to minimize eye strain. Among

such smart bulbs you can find the ones produced by Sengled which is the leading company in smart lightning. Here is the scheme of how the lamps work.

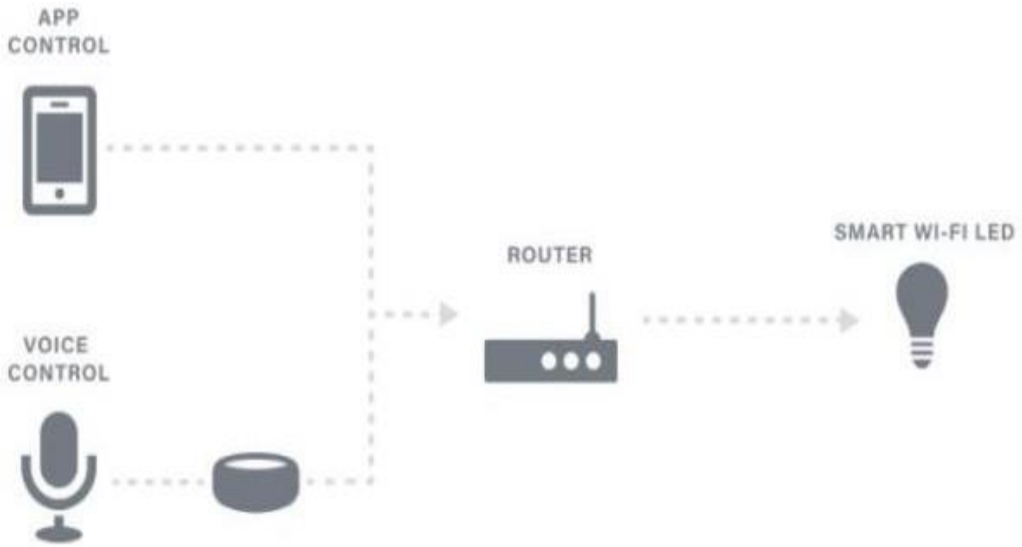


Fig.1.10 : IoT Intelligent Lighting

IoT printers

Nobody will be surprised by the fact that you can manage printers thanks to your device and Internet connection. However, smart printing is not the only advantage of these IoT powered devices. They can monitor the supplies (i.e. the paper, ink) and even perform self-diagnostics to notify the employees that a quick fix or serious repair is needed. And what's more exciting such smart IoT printers can be connected to your inventory system and let you know via notification when you need to fill in your office supplies.

Smart locks

This IoT solution will be appreciated by the owners of big companies, because employees tend to lose the keys and swipe cards that are used to let them into the office. So to keep your office safe and make entrance procedure easier for the employees, you can install smart locks connected to your attendance system. With the right set of options smart locks can even let you know what employee and when enteredthe office, and how many times he lives it during the day. On top of that, smart locks minimize the risk of break-ins.

Logistic Industry Innovation with IoT

Logistics management requires monitoring multiple activities at once — supply chain, warehousing, and so on. There are dozens of factors that can influence the process itself and cause delays. To streamline processes and increase customer satisfaction, logistics managers should embrace innovations that IoT has to offer.

Challenges within the logistics process

The following illustration shows an example of the process of a logistics network operator. Changing external factors, as well as increasing customer requirements pose the need for further development and expansion of the current process. With the pressure to become more and more efficient, this confronts the logistics industry with several challenges.

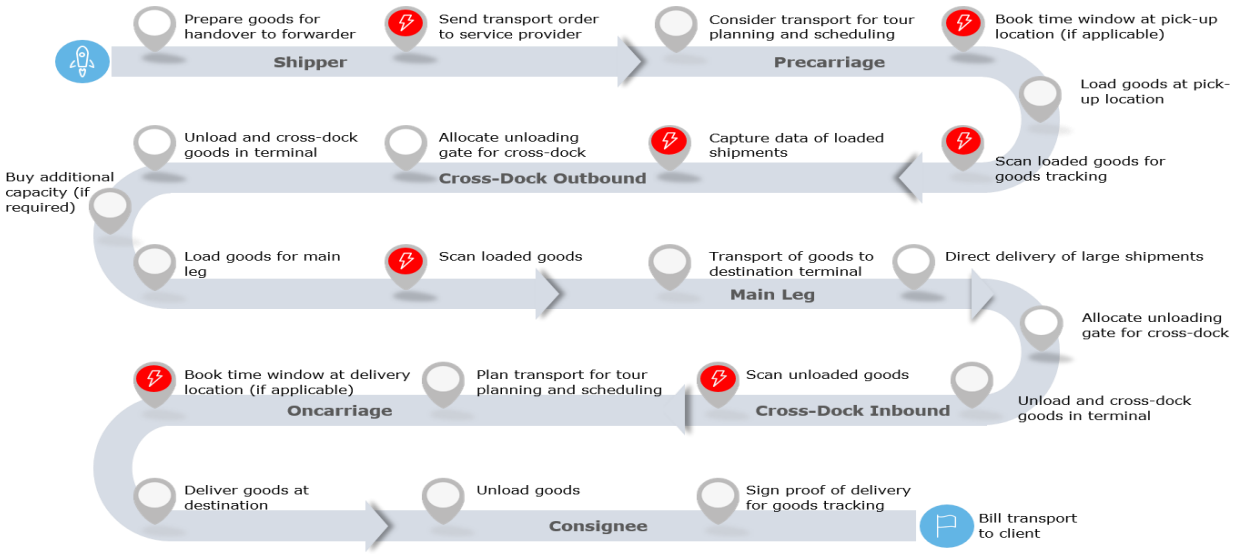


Fig.1.11 : Challenges in logistics process

Receiving transport orders

Digitization in companies is advancing more and more. Even though processes are increasingly being carried out electronically, logistics service providers often receive their orders outside the

transport systems - often still via analog channels. This increases the effort required on the part of logistics service providers to record shipment data.

Booking time windows

In order to optimize the yard planning, staff utilization and operational processes, an increasing number of clients is asking their logistics providers to book dedicated time windows to pick-up and deliver goods. The booking of the time windows usually needs to take place 24h before pick-up/delivery. The booking itself causes additional effort for the logistics providers and reduces efficiency. In addition to that, the flexibility to plan the pick-up/delivery tour is reduced which can negatively affect the utilization of the available loading capacity.

Scanning of goods

Customers are increasingly demanding reliable transparency on the status of shipments from their service providers. This requirement presents logistics companies with an enormous challenge. On the one hand, the recording of the conventional barcode by warehouse personnel is time-consuming and therefore inefficient. On the other hand, additional information, such as temperature or vibrations, must be transmitted to the customer more frequently. Current transport systems are often not suited to provide this type of information.

Capture data of loaded shipments

In the event that companies do not transmit their shipment data electronically, the data must be recorded by the service provider. This entails the risk of incorrect information being recorded, which is even increased by the existing shortage of skilled workers and outsourcing. In addition, shipments cannot be reloaded as long as the shipment data has not been recorded. In reality, this repeatedly leads to delays in operational processes.

Goals of IoT Technology for the Logistics Industry

Companies use IoT to improve logistics processes — both in warehouses and beyond. According to Statistics, by next year worldwide companies will spend over $40 billion on connected products to increase the productivity of deliveries.

1. Improved security and theft detection

Connected applications increase the control over who enters the warehouse at any given time, help track all items, and alert the business manager in case something goes missing. Examples of IoT-powered security applications are connected CCTVs, apps that allow warehouse managers to block the doors of the facility remotely, asset tracking logistics applications that help monitor deliveries, and more.

2. Higher employee safety

Unreliable machinery puts companies at risk of endangering the lives of its staff. The Internet of Things' effects on logistics include employee protection by detecting equipment issues long before a human interacts with a tool. IoT sensors significantly increase the response time, in case something happened to an employee. A wearable immediately detects a critical change in vitals and transfers the data to a dedicated platform that alerts the manager and even calls an ambulance.

3. End-to-end product tracking

Increasing the transparency of the delivery process is the primary objective business managers want to achieve with IoT implementation. Being able to track the product all the way from the warehouse to the customer's doorstep increases the manager's confidence that all stages of the supply chain are completed smoothly. It also boosts the client's trust in the brand and saves support agents a ton of time as customers no longer bother customer support with delivery status update requests.

4. Providing business managers with advanced analytics

Thanks to a broad range of applications, the Internet of Things provides business managers with the big-picture view of the way all operations are handled.By collecting data from sensors and presenting it in a concise, understandable way, IoT offers a holistic approach to logistics management.

Here's a range of things company owners will be aware of:

- The number of items in the warehouse;
- The temperature in the warehouse;
- The environmental conditions during deliveries;
- Employee efficiency.

Real-time delivery and inventory monitoring improves the quality of planning and budget allocation. The datum, provided by the Internet of Things and logistics, comes in handy during inspections as well so that nothing will catch a manager by surprise.

5. Improving delivery

There are a handful of ways the Internet of Things facilitates delivery management. RFID tags and connected GPS sensors help business managers track shipping all the way to its final stage.

Also, thanks to connected sensors, logistics managers can get real-time location data to ensure the weather or other environmental changes will not jeopardize the delivery.

Internet of Things is Improving the Transportation Industry

Logistics and transportation have always been risky fields due to the lack of control over weather conditions, high odds of scams, and a wide range of assets to manage.

With the Internet of Things, logistics can finally become a fully controlled domain, where all the factors that could negatively impact the delivery process can be either neutralized or avoided.

In a nutshell, these are the benefits of using the Internet of Things in transportation:

- Reliable vehicle tracking. The Internet of Things helps businesses track the location of each vehicle and compare the most cost-efficient route with the one a driver has taken. Being highly aware of the ins-and-outs of the delivery process helps company owners evaluate employee performance and incentivize best practices, proactively react to problems on the road,and manage them with the lowest number of losses.

- Reducing shipping costs. Automated order processing and status updating help companies cut the number of employees in charge of shipping, reducing overall operating costs. Using connected bots for last-mile delivery helps cut costs exponentially, as well as increase customer satisfaction. Amazon has been benefitting from autonomous bots and drop-shipping lockers for over 5 years, improving the convenience of delivery and generating profit.
- Improved supply chain planning. The Internet of Things provides businesses with multi-faceted data — how much time it takes to sell a given amount of products in the inventory, what the ways to optimize deliveries are, which employees have better track records, which distribution centers have higher conversions. As a result, business managers can plan operations and predict the outcome of business decisions rather accurately.
- Employee monitoring. IoT devices in logistics allow business managers to monitor the staff's physical safety using wearables and vital sensors. With the Internet of Things, you will be able to protect employees from exposure to toxic substances and alert drivers if they are not adhering to safety practices. There are even devices that can detect an employee's moving speed and the number of bathroom breaks and evaluate staff's efficiency based on the data. To some managers, such in-depth tracking may be over-the-top. However, it does provide you with a better understanding of employee motivation and time-management skills.
- Preventing product theft and monitoring transportation conditions. The range of IoT and logistics anti-theft devices is enormous — connected hardware to detect intrusion, sensors for real-time asset tracking, alarm systems, smart fences, and others. A business manager will be able to find an IoT solution for logistics that provides an increased inventory and supply chain visibility, protects from scams, product theft, or tampering.

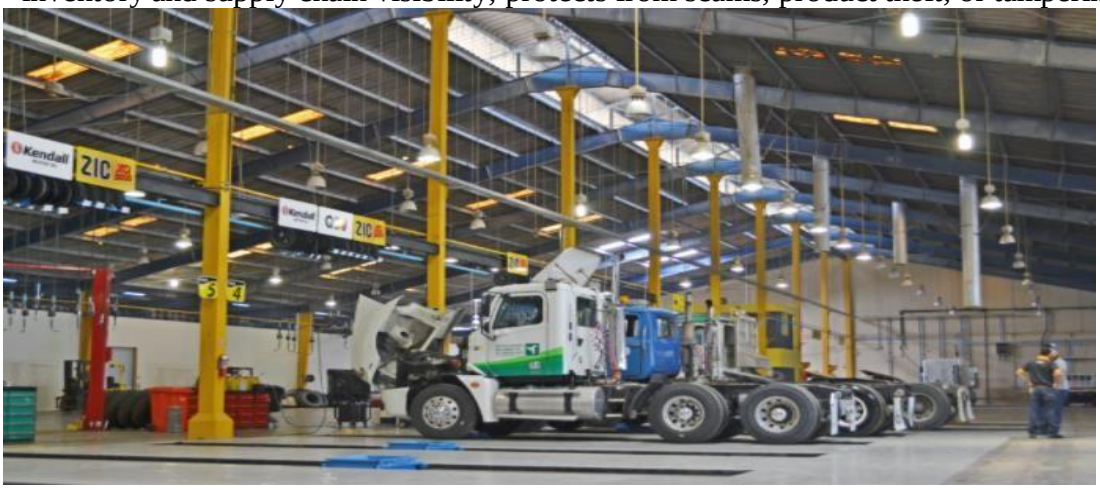


Fig.1.12 : Internet of Things in Transportation

Internet of Things Logistics Use Cases

1. Inventory tracking systems

Inventory tracking systems assist logistics managers in planning re-stocking and distributions.

2. Predictive analytics systems

Predictive analytics solutions help business managers make informed decisions regarding warehouse management and supply chain planning.

3. Location management tools

With IoT in transportation and logistics, company managers can keep track of the real-time location of each vehicle, delivery statuses, and the estimated time needed to complete the process.

4. Drone-based delivery

Drones are an efficient way to speed up and automate deliveries. In logistics, they can be used to improve navigation within the warehouse, provide customers with instant in-store deliveries, and solve last-mile delivery issues.

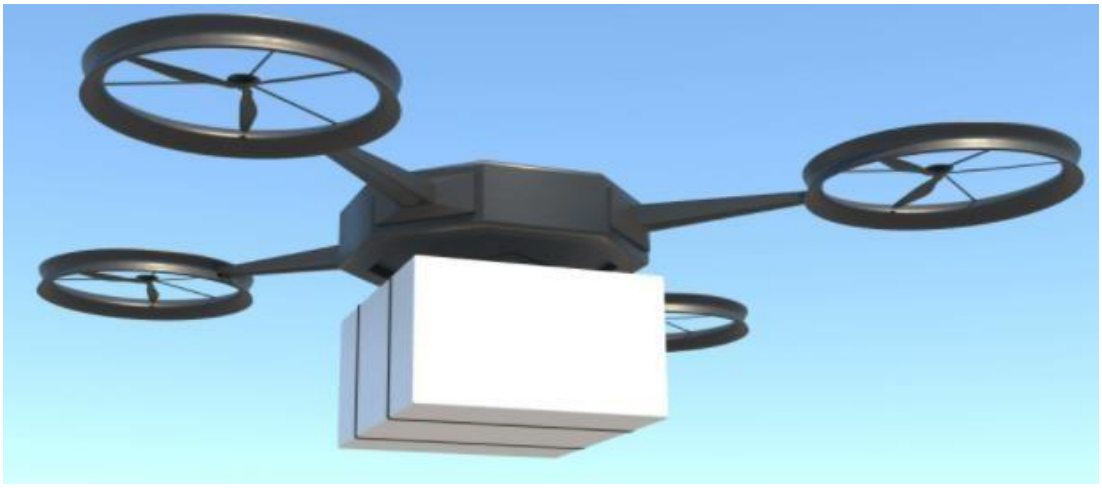


Fig.1.13 : Drone based delivery

5. Automated vehicles

Using AVs helps business managers have more control over the delivery process, reduce the impact of human error, and benefit from machine intelligence.

IoT Innovations in Retail

Amazon.com alone made an incredible leap from an average of 2 billion to 2.5 billion monthly visitors starting from February 2020.The potential of retail in our digital age is staggering i.e the industry is aware of its own issues and drawbacks. Consumer habits, high pressure on delivery services, buyers' mistrust in online purchases or lack of tech fluency are all factors that hold sellers back in an environment they could otherwise be thriving in. However, the pandemic has made many retailers rethink their strategies and speed up towards digitization.

Customer Experience Optimization with IoT

The Internet of Things and retail allows store managers to find new ways to establish a connection with a client, create a short and direct customer journey, improve the process of product maintenance, and build a long-lasting bond with first-time shoppers. Here are several examples of the application of IoT in retail for customer experience improvement.

1. Personalized communications based on the IoT-collected data

The Internet of Things makes a good case for improving communication between a customer and a brand. IoT sensors can track down a customer’s habits and share insights with the marketing team. Content teams will be able to create segment-specific personalized content that would help a shopper to find the product he or she is looking for, get tips and advice, or introduce a friend to the store.

2. Optimizing product usage

All the data collected by IoT allows brands to improve product maintenance, features, even design. A company can tweak the settings and make necessary updates on a product as a client uses it at home. Moreover, all insights during the run of the product will be collected and transferred back to the company’s server. When it’s time to design a new lineup, all the gathered data will prove useful.

3. Monitor and predict in-store wait times

This is an important benefit of IoT technology in retail in the time of the pandemic. Long lines at cash decks increase the risks associated with COVID-19 and lower customer retention rates. It’s

not just the wait that leaves the customer frustrated so much as not being able to predict the amount of time they will spend waiting. The Internet of Things is helpful as it allows brands to manage in-store wait times. The technology can provide a store's employees with data on how long a user has been waiting, suggest distractions, or offer a quieter place or useful activities to make the time in line more tolerable.

4. Using wearables for loyalty programs

Wearable technology has been a known success for fitness and healthcare. However, the wearable IoT application in retail is not limited to tracking health data. In fact, retail companies can benefit from wearables to identify loyal clients. Hotels use wristbands to identify premium guests and offer additional bonuses and discount programs for their stay. Wristbands are a non-invasive way to offer a loyalty program and say 'thank you' to those who have supported the brand since its first days.

5. Keep the customer updated on the product delivery status

The demand for delivery has surged during the pandemic and so has the pressure on the delivery services. However, insecurities regarding product delivery time and safety remain. Brands can use the Internet of Things to make sure customers are clear on expectations. The technology allows a retailer to create updates regarding the delivery status so a user can see the location of their order in real time.

Ideas for IoT use cases in retail

There is a ton of applications of IoT in retail industry for improving customer experience and just about as many in retail management. Here are the main opportunities of the Internet of Things in retail.

Customer experience personalization

Using the Internet of Things is a good way for a brand to foster a personal connection between the brand and its customers. For instance, you can attract passersby to visit your store by sending an IoT-enabled notification to their smartphones. Retailers can use the technology to find out more about a customer in order to lay the groundwork for microtargeting. This way, marketing

managers will be able to make more conscious choices and use more focused and cost-efficient advertising.

Supply chain optimization

GPS and RFID technology will allow brands to track each individual item through the entire delivery process. You will be able to have a tight grip on your vendors as you will be able to monitor the delivery conditions and the location as well as predict a precise delivery time. The range of applications of IoT in supply chain management is impressive. For instance, you can test different vendors, vehicles, and delivery routes; collect the data on the process; and find the cheapest framework that also transports the product with no damage.

Innovating in-store experiences

Implementing the Internet of Things can help retailers to redesign their stores completely. You'll be able to provide a new experience for fitting rooms, create a system of intelligent suggestions, and go as far as to replace human workers with connected technology. Amazon Go is, without a doubt, the most famous and successful example of large-scale IoT implementation for revolutionizing the in-store experience.

Increased store management efficiency

The Internet of Things and cloud-based technologies empower a range of solutions that improve the efficiency of business operations in retail. These include:

- Automated packaging services;
- SKU accounting;
- IoT drones for inventory monitoring

Implementing IoT for retail management and as a part of warehouse technology results in reducing shrinkage, managing each storage unit, and navigating the inventory easier.

Decrease the amount of workforce needed for running a store

One of the most effective ways of reducing the amount of workforce involved in store management is automating tasks using robot tech and network solutions. What seemed like something straight out of sci-fi just a few years ago is now commonplace.

IoT Applications in Retail

Location tracking

The Internet of Things solves one of the biggest issues in retail — lack of delivery reliability. The technology is capable of increasing operational efficiencies and improving logistic transparency.

Predictive equipment maintenance

Malfunctioning electric appliances (refrigerator units, for example) can lead to tremendous reputational and monetary losses, and it can send dozens of product units to waste. In order to be updated on store maintenance and take a proactive approach in equipment managers, store managers often use IoT in retail. The technology is capable of providing real-time equipment monitoring and notifying the user in case of likely malfunctioning. predictive equipment maintenance has already been implemented in transportation (Volvo and IBM), manufacturing (Chevron and Microsoft), and utilities (Florida Power Light)

Inventory management

IoT allows store managers to automate product orders, is capable of notifying when a certain product needs to be re-ordered, gathers and analyses data regarding the popularity of a certain item, and prevents theft.

There is no lack of inventory-centered IoT solutions, including:

MIT Drone Inventory System — an IoT-based drone that monitors inventory in real time and sends alerts in case there are no available units left.

Intel Retail Sensor Platform — the RFID antenna scans the number of units on the sales floor and alerts a store manager in case it's low. The platform has a plug-and-play interface and easy to adopt and use.

Lululemon uses RFID-based technology for customer-facing inventory managers. A buyer can conduct a real-time check to ensure the desired product is available at the nearest store.

Shopper mapping and analyzing mall traffic

By placing IoT sensors around the store, managers will be able to get a better understanding of the most popular zones and products. User activity heat maps help understand where it's better to put items for sale, how to optimize store space to use spaces with low activity in a more efficient way, and record and trace shopping trends over time.

Smart shelves

Smart shelf technology was widely introduced to the retail market when Kroger, the supermarket chain with the highest revenue in the US, tested over 2,000 smart shelves in 2016. As a stocker walks around the shop with a digital shopping list on their smartphone, the cell phone will vibrate in case a needed product is on the shelf nearby. Smart shelves have three common elements — an RFID tag, an RFID reader, and an antenna. All the data collected by smart shelves during the day will be later shared with a store manager to provide customer-related insights.

Personalized alerts

IoT-based hyper-personalization is widely used in retail. Geofencing and IoT beacons are both tried-and-true ways to catch a customer's attention. Here are a few IoT examples in retail that have to do with these technologies:

- Starbucks IoT beacons. Passing by a Starbucks, people would get notifications about new coffee brews or promotions and were invited to visit. According to RT Insights, the campaign has proven to be highly efficient.
- Starbucks IoT beacons. Passing by a Starbucks, people would get notifications about new coffee brews or promotions and were invited to visit. According to RT Insights, the campaign has proven to be highly efficient.

Benefits of IoT in the Retail Industry

Reducing shrinkage and fraud

As the Internet of Things adds an additional layer of traceability and visibility of the inventory and delivery process.

Optimizing product placement.

IoT allows store managers to identify premium store areas, test the placement of different items in those spots, and find the most efficient layout thanks to detailed reports based on the data gathered by sensors.

Efficient use of in-store staff.

IoT can use cameras, sensors, and facial recognition algorithms in order to identify an impatient or confused shopper. Staff will be able to make proactive decisions and successfully engineer theatmosphere within the store

Improved retail management and tracking.

IoT helps store managers be aware of the number of products on the shelves and in the inventory

Connecting online and in-store experiences

The Internet of Things in the retail industry allows users to benefit from brand-related digital solutions while using physical stores. This way, retail companies can achieve synergy between ecommerce and in-store experiences.

Text/Reference Books

1. S. Misra, A. Mukherjee, and A. Roy, Introduction to IoT. Cambridge University Press, 2020
2. S. Misra, C. Roy, and A. Mukherjee, Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.2020
3. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015

Question Bank

PART-A

1. List the major benefits of IIOT.
2. Compare IOT and IIOT in terms of functionality, connectivity and usage.
3. Illustrate the infrastructure of Industrial Internet of Things.
4. Identify the challenges in IIOT in Oil and gas industry.
5. Examine how IOT in smart office helps better productivity.
6. List down any 3 major use cases of IOT.
7. Examine how predictive maintenance is done using IIOT.

PART-B

1. Explain in detail the architecture of Industrial Internet of Things.
2. Describe in detail the working and communication of any four IOT intelligent devices.
3. Design and explain an architecture to implement IOT for health care in an hospital where Covid situation must be controlled via IOT.
4. How IOT in OIL and GAS industry works, Explain in detail.
5. Discuss in detail about the communication methods employed in IIOT.
6. Examine the challenges and working of Logistic Industry in IOT.

INDUSTRIAL INTERNET OF THINGS – SECA4005

UNIT – II TECHNICAL AND BUSINESS INNOVATORS OF INDUSTRIAL INTERNET

TECHNICAL AND BUSINESS INNOVATORS OF INDUSTRIAL INTERNET

Miniaturization – Cyber Physical Systems – Wireless technology – IP Mobility – Network Functionality Virtualization – Cloud and Fog - Big Data and Analytics – M2M Learning and Artificial Intelligence.

MINIATURIZATION

In the world of Internet of Things (IoT), miniaturization is enabling new applications in the form of wearables, vehicles and transportation, disposable tracking tech for pharmaceuticals and produce, and more uses than we can count for smart city and smart home use.

In this digital era, as we wirelessly connect more and more devices to the Internet, researchers and engineers face several challenges, like how to package a radio transmitter into their existing device real estate, how to make increasingly smaller devices, how to reduce the area coverage for mounting chips. They are also striving to meet consumer demand for Internet of Things (IoT) products that are ergonomically easy to use.

Ideally, engineers would tend to use IoT components that are smaller in size, have better RF performance, and have reasonable prices. However, these characteristics do not usually converge in IoT component offerings, and that presents a challenge for solution providers.

Fortunately, the size of a silicon die has been getting smaller and smaller over the years as the industry adopts new silicon manufacturing processes. The industry has been solving the space issue for IoT implementations by combining the MCU and RF frontend into system-on-chip (SoC) configurations.

The demand for embedded SIM (eSIM) is steadily rising among the smartphone manufacturers, laptop manufacturers, energy & utility sector companies. The OEMs across the globe are focusing on the development and integration of eSIM in numerous applications.

The increasing demand for miniaturization of IoT components across various industries is also boosting the demand for eSIM globally.

In 2018, researchers from the Green IC group at the National University of Singapore (NUS) in collaboration with associate professor Paolo Crovetti from the Polytechnic University of Turin in Italy created the timer, that trigger sensor to perform their tasks when required, is believed to be so efficient that it runs using an on-chip solar cell with a diameter close to that of a human hair. This is a major step in IoT miniaturization claimed with low-power.

The wake-up timer can continue operations even when a battery is not available and with very little ambient power, as demonstrated by a miniaturized on-chip solar cell exposed to moonlight. An on-chip capacitor used for slow and infrequent wake-up also helps reduce the device's silicon manufacturing cost thanks to its small surface area of 49 microns on both sides.

IoT sensor nodes are individual miniaturized systems containing one or more sensors, as well as circuits for data processing, wireless communication, and power management. To keep power consumption low, they are kept in sleep mode most of the time, and wake-up timers are used to trigger the sensors to carry out a task. As they are turned on most of the time, wake-up timers set the minimum power consumption of IoT sensor nodes. They also play a fundamental role in reducing the average power consumption of systems-on-chip.

When designing a hardware module, one of the pressing questions is about Antenna. Developers must work around the space reserved for antenna and the type of antenna they will use to integrate with a corresponding module. PCB trace antennas are general preference because of their low bill of material (BoM) costs. But they require a significant size which can cause devices to be large and difficult to work with.

The smaller size we try to achieve, the less efficiency we can have for the RF performance. Chip antennas are famous for various applications as they simplify design efforts and optimize size consumption.

According to statistics of Bluegiga, approximately only 10 percent of these evaluated designs deploy the external antenna, and 90 percent of the customers choose modules with a built-in chip antenna. Hence, it becomes necessary to continuously evaluate the possibility of space reduction on chipboard, something Cloud of Things has successfully achieved with our latest DeviceTone Genie product line, working with great partners including Nordic Semiconductor and AES with their minIot devices.

Importance of Miniaturization

Miniaturization produced sleeker computers and phones that take up less space and produce less waste in the manufacturing and assembly processes, but smaller technology is more stylish.

Miniaturization in form factor chipsets and modules has contributed to cost-effective, faster-running, and more powerful computer components.

In the world of Internet of Things (IoT), miniaturization is enabling new applications in the form of wearables, vehicles and transportation, disposable tracking tech for pharmaceuticals and produce, and more uses than we can count for smart city and smart home use.

Miniaturization in MEMS Sensors

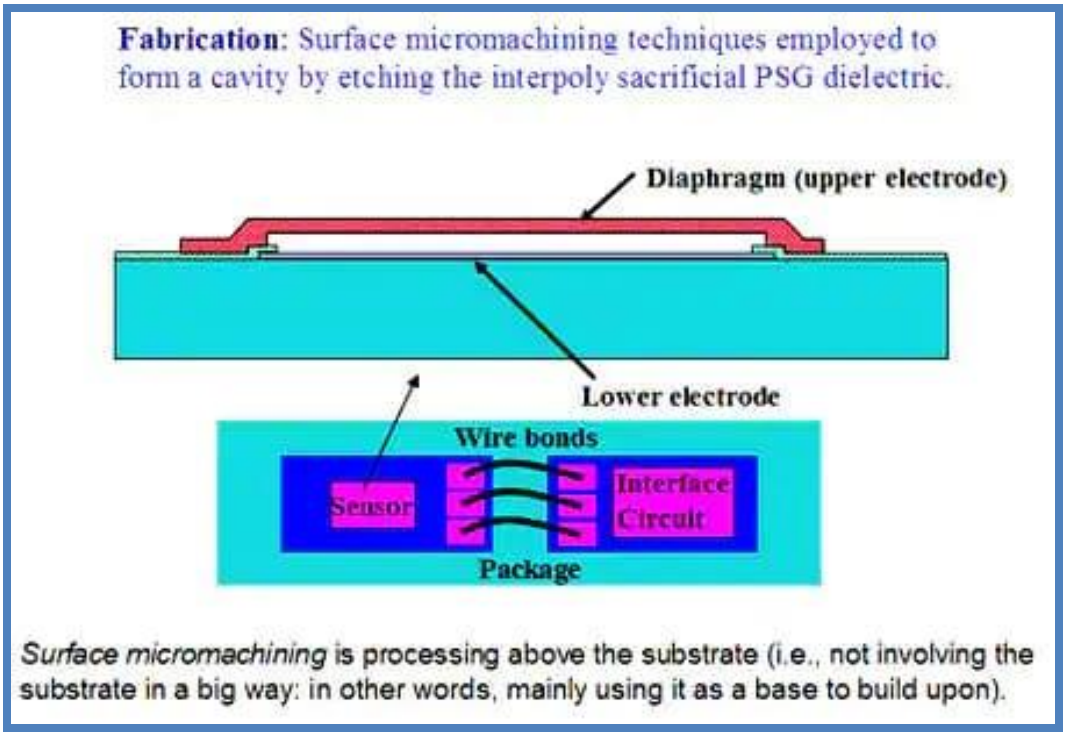


Fig.2.1 Miniaturization

Micromachining has become a key technology for the miniaturization of sensors. Being able to reduce the size of the sensing element by using standard semiconductor manufacturing technology allows a dramatic reduction in size. Integrating the signal processing alongside the sensing element further enhances the opportunities to reduce the size of the system, eliminating the need for extra pins to link to external devices.

The choice of micromachining process technology can also determine the limits of miniaturization, but this is often determined by the sensor type. Piezoelectric micro machined elements for pressure sensing have less opportunity to scale than a diaphragm built from CMOS silicon on the surface of a substrate, for example, but can deliver higher performance.

Uses of Miniaturized Technology

The increased applications for IoT extend from personal use with wrist wear, footwear, eyewear, body wear, and neckwear associated with personal use in training and fitness, as well as more practical applications, such as sports, infotainment, healthcare, defense, enterprise, and industry.

The industrial applications of wearable technology will see major benefits in the healthcare segment in which connected devices improve efficiency and reduce operational costs.

By creating more powerful devices with smaller footprints – particularly through the use of improved edge processing – providers and facilities will gain the ability to keep track of patients through real-time monitoring of vital signs and health stats. From wristbands to implants, data is

transmitted through the cloud and analyzed to produce more accurate outcomes and treatment options.

In the military industry, wearable technology "can help soldiers in the field by tracking them more accurately, giving central command more precision in coordinating operations," according to Emily Rector with MarketScale. Wireless, hands-free communications and more efficient battery life could contribute to timesaving and lifesaving operations.

Limits Of Miniaturization

In addition, miniaturized equipment is frequently not as easy to maintain and therefore typically does not receive the same routine maintenance and care that larger equipment receives.

This can lead to increased overall costs as a result of disposal and the overheads required to keep additional equipment on hand.

CYBER PHYSICAL SYSTEMS

Cyber – computation, communication, and control that are discrete, logical, and switched

Physical – natural and human-made systems governed by the laws of physics and operating in continuous time

Cyber-Physical Systems – systems in which the cyber and physical systems are tightly integrated at all scales and levels

A cyber-physical system (CPS) or intelligent system is a computer system in which a mechanism is controlled or monitored by computer-based algorithms. In cyber-physical systems, physical and software components are deeply intertwined, able to operate on different spatial and temporal scales, exhibit multiple and distinct behavioral modalities, and interact with each other in ways that change with context. CPS involves transdisciplinary approaches, merging theory of cybernetics, mechatronics, design and process science. The process control is often referred to as embedded systems. In embedded systems, the emphasis tends to be more on the computational elements, and less on an intense link between the computational and physical elements. CPS is also similar to the Internet of Things (IoT), sharing the same basic architecture; nevertheless, CPS presents a higher combination and coordination between physical and computational elements.

Examples of CPS include smart grid, autonomous automobile systems, medical monitoring, industrial control systems, robotics systems, and automatic pilot avionics. Precursors of cyber-physical systems can be found in areas as diverse as aerospace, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing, transportation, entertainment, and consumer appliances.

CPS Characteristics

• CPS are physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated.

• This intimate coupling between the cyber and physical is what differentiates CPS from other fields.

Some hallmark characteristics:

• Cyber capability in every physical component

• Networked at multiple and extreme scales

• Complex at multiple temporal and spatial scales

• Constituent elements are coupled logically and physically

• Dynamically reorganizing/reconfiguring open system.

• High degrees of automation, control loops closed at many scales

• Unconventional computational & physical substrates (such as bio, nano, chem, ...)

• Operation must be dependable, certified in some cases.

Mobile Cyber-physical systems

Mobile cyber-physical systems, in which the physical system under study has inherent mobility, are a prominent subcategory of cyber-physical systems. Examples of mobile physical systems include mobile robotics and electronics transported by humans or animals. The rise in popularity of smart phones has increased interest in the area of mobile cyber-physical systems. Smartphone platforms make ideal mobile cyber-physical systems for a number of reasons, including:

- Significant computational resources, such as processing capability, local storage
- Multiple sensory input/output devices, such as touch screens, cameras, GPS chips, speakers, microphone, light sensors, proximity sensors
- Multiple communication mechanisms, such as WiFi, 4G, EDGE, Bluetooth for interconnecting devices to either the Internet, or to other devices
- High-level programming languages that enable rapid development of mobile CPS node software, such as Java,] C#, or JavaScript
- Readily available application distribution mechanisms, such as Google Play Store and Apple App Store
- End-user maintenance and upkeep, including frequent re-charging of the battery

Examples of Cyber Physical System

Common applications of CPS typically fall under sensor-based communication-enabled autonomous systems. For example, many wireless sensor networks monitor some aspect of the environment and relay the processed information to a central node. Other types of CPS include smart grid, autonomous automotive systems, medical monitoring, process control systems, distributed robotics, and automatic pilot avionics.

A real-world example of such a system is the Distributed Robot Garden at MIT in which a team of robots tend a garden of tomato plants. This system combines distributed sensing (each plant is equipped with a sensor node monitoring its status), navigation, manipulation and wireless networking.

A focus on the control system aspects of CPS that pervade critical infrastructure can be found in the efforts of the Idaho National Laboratory and collaborators researching resilient control systems. This effort takes a holistic approach to next generation design, and considers the resilience aspects that are not well quantified, such as cyber security,[18] human interaction and complex interdependencies.

Another example is MIT's ongoing CarTel project where a fleet of taxis work by collecting real-time traffic information in the Boston area. Together with historical data, this information is then used for calculating fastest routes for a given time of the day.

CPS are also used in electric grids to perform advanced control, especially in the smart grids context to enhance the integration of distributed renewable generation. Special remedial action scheme are needed to limit the current flows in the grid when wind farm generation is too high. Distributed CPS are a key solution for this type of issues.

In industry domain, the cyber-physical systems empowered by Cloud technologies have led to novel approaches that paved the path to Industry 4.0 as the European Commission IMC-AESOP project with partners such as Schneider Electric, SAP, Honeywell, Microsoft etc. demonstrated.

The Internet of Things (IoT) starts with connectivity, but since IoT is a widely diverse and multifaceted realm, you certainly cannot find a one-size-fits-all communication solution. Continuing our discussion on mesh and star topologies, in this article we'll walk through the six most common types of IoT wireless technologies.

Each solution has its strengths and weaknesses in various network criteria and is therefore best-suited for different IoT use cases.

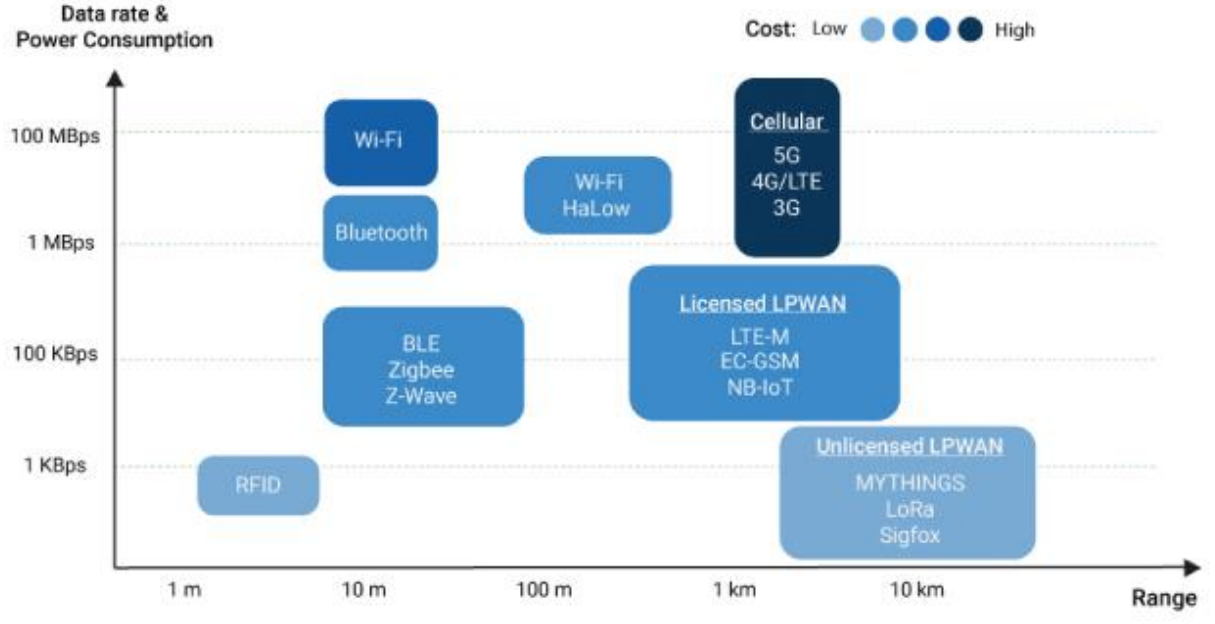


Fig. 2.2 : Wireless technologies

1. LPWANs

Low Power Wide Area Networks (LPWANs) are the new phenomenon in IoT. By providing long-range communication on small, inexpensive batteries that last for years, this family of technologies is purpose-built to support large-scale IoT networks sprawling over vast industrial and commercial campuses.

LPWANs can literally connect all types of IoT sensors – facilitating numerous applications from **asset tracking, environmental monitoring** and **facility management** to **occupancy detection** and **consumables monitoring**. Nevertheless, LPWANs can only send small blocks of data at a low rate, and therefore are better suited for use cases that don't require high bandwidth and are not time-sensitive.

Also, not all LPWANs are created equal. Today, there exist technologies operating in both the licensed (NB-IoT, LTE-M) and unlicensed (e.g. MYTHINGS, LoRa, Sigfox etc.) spectrum with varying degrees of performance in key network factors. For example, while power consumption is a major issue for cellular-based, licensed LPWANs; Quality-of-Service and scalability are

main considerations when adopting unlicensed technologies. Standardization is another important factor to think of if you want to ensure reliability, security, and interoperability in the long run.

2. Cellular (3G/4G/5G)

Well-established in the consumer mobile market, cellular networks offer reliable broadband communication supporting various voice calls and video streaming applications. On the downside, they impose very high operational costs and power requirements.

While cellular networks are not viable for the majority of IoT applications powered by battery-operated sensor networks, they fit well in specific use cases such as **connected cars** or **fleet management in transportation and logistics**. For example, in-car infotainment, traffic routing, advanced driver assistance systems (ADAS) alongside fleet telematics and tracking services can all rely on the ubiquitous and high bandwidth cellular connectivity.

Cellular next-gen 5G with high-speed mobility support and ultra-low latency is positioned to be the future of autonomous vehicles and augmented reality. 5G is also expected to enable real-time video surveillance for **public safety**, real-time mobile delivery of medical data sets for **connected health**, and several **time-sensitive industrial automation** applications in the future.

3. Zigbee and Other Mesh Protocols

Zigbee is a short-range, low-power, wireless standard (IEEE 802.15.4), commonly deployed in mesh topology to extend coverage by relaying sensor data over multiple sensor nodes. Compared to LPWAN, Zigbee provides higher data rates, but at the same time, much less power-efficiency due to mesh configuration.

Because of their physical short-range (< 100m), Zigbee and similar mesh protocols (e.g. Z-Wave, Thread etc.) are best-suited for medium-range IoT applications with an even distribution of nodes in close proximity. Typically, Zigbee is a perfect complement to Wi-Fi for various home automation use cases like smart lighting, HVAC controls, security and energy management, etc. – leveraging home sensor networks.

Until the emergence of LPWAN, mesh networks have also been implemented in industrial contexts, supporting several remote monitoring solutions. Nevertheless, they are far from ideal for many industrial facilities that are geographically dispersed, and their theoretical scalability is often inhibited by increasingly complex network setup and management.

4. Bluetooth and BLE

Defined in the category of Wireless Personal Area Networks, Bluetooth is a short-range communication technology well-positioned in the consumer marketplace. Bluetooth Classic was originally intended for point-to-point or point-to-multipoint (up to seven slave nodes) data exchange among consumer devices. Optimized for power consumption, Bluetooth Low-Energy was later introduced to address small-scale Consumer IoT applications.

BLE-enabled devices are mostly used in conjunction with electronic devices, typically smartphones that serve as a hub for transferring data to the cloud. Nowadays, BLE is widely integrated into fitness and medical wearables (e.g. smartwatches, glucose meters, pulse oximeters, etc.) as well as Smart Home devices (e.g. door locks) – whereby data is conveniently communicated to and visualized on smartphones.

The release of Bluetooth Mesh specification in 2017 aims to enable a more scalable deployment of BLE devices, particularly in retail contexts. Providing versatile indoor localization features, BLE beacon networks have been used to unlock new service innovations like in-store navigation, personalized promotions, and content delivery.

5. Wi-Fi

There is virtually no need to explain Wi-Fi, given its critical role in providing high-throughput data transfer for both enterprise and home environments. However, in the IoT space, its major limitations in coverage, scalability and power consumption make the technology much less prevalent.

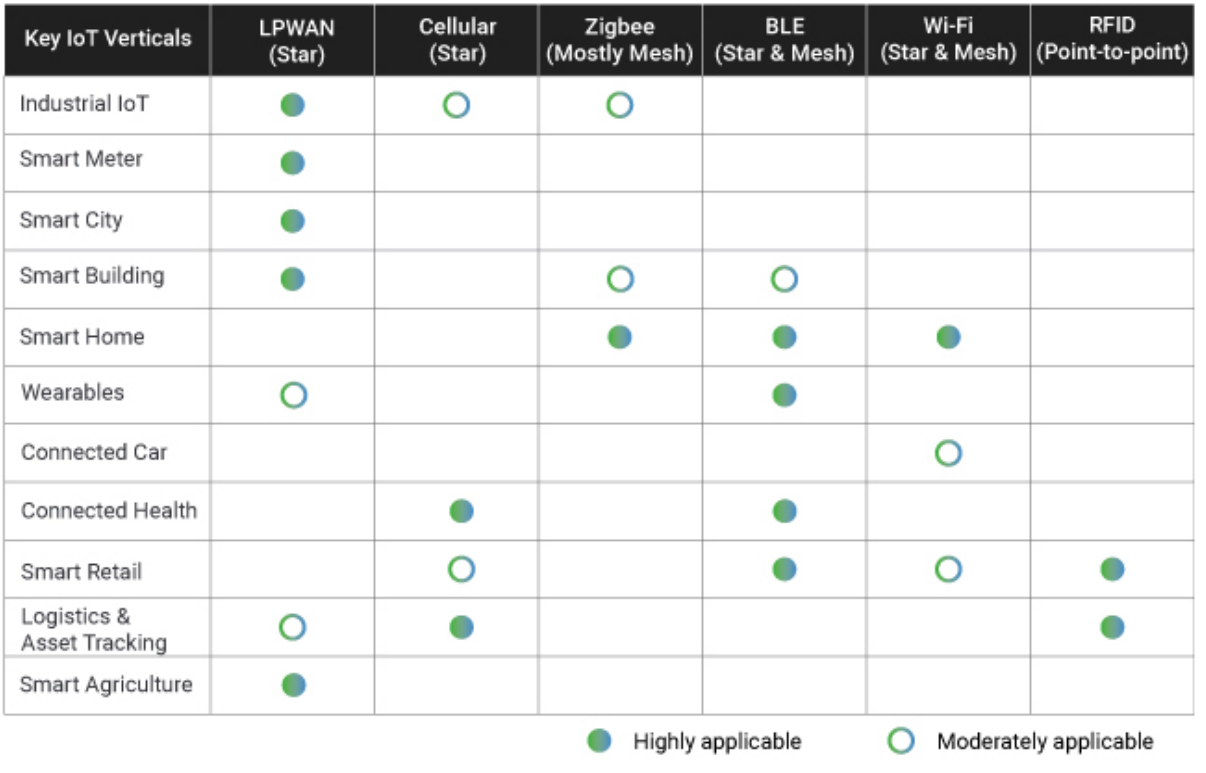
Imposing high energy requirements, Wi-Fi is often not a feasible solution for large networks of battery-operated IoT sensors, especially in industrial IoT and smart building scenarios. Instead, it more pertains to connecting devices that can be conveniently connected to a power outlet like smart home gadgets and appliances, digital signages or security cameras.

Wi-Fi 6 – the newest Wi-Fi generation – brings in greatly enhanced network bandwidth (i.e. <9.6 Gbps) to improve data throughput per user in congested environments. With this, the standard is poised to level up public Wi-Fi infrastructure and transform customer experience with new digital mobile services in retail and mass entertainment sectors. Also, in-car networks for infotainment and on-board diagnostics are expected to be the most game-changing use case for Wi-Fi 6. Yet, the development will likely take some more time.

6. RFID

Radio Frequency Identification (RFID) uses radio waves to transmit small amounts of data from an RFID tag to a reader within a very short distance. Till now, the technology has facilitated a major revolution in retail and logistics.

By attaching an RFID tag to all sorts of products and equipment, businesses can track their inventory and assets in real-time – allowing for better stock and production planning as well as optimized supply chain management**.** Alongside increasing IoT adoption, RFID continues to be entrenched in the retail sector, enabling new IoT applications like smart shelves, self-checkout, and smart mirrors.



IP MOBILITY

The increasing use of virtualization in the data center has enabled an unprecedented degree of flexibility in managing servers and workloads. One important aspect of this newfound flexibility is mobility. As workloads are hosted on virtual servers, they are decoupled from the physical infrastructure and become mobile by definition. As end-points become detached from the physical infrastructure and are mobile, the routing infrastructure is challenged to evolve from a topology centric addressing model to a more flexible architecture. This new architecture is capable of allowing IP addresses to freely and efficiently move across the infrastructure. There are several ways of adding mobility to the IP infrastructure, and each of them addresses the problem with

different degrees of effectiveness. LISP Host Mobility is poised to provide a solution for workload mobility with optimal effectiveness. This document describes the LISP Host Mobility solution, contrasts it with other IP mobility options, and provides specific guidance for deploying and configuring the LISP Host mobility solution.

IP Mobility Requirements

The requirements for an IP mobility solution can be generalized to a few key aspects. To make a fair comparison of existing solutions and clearly understand the added benefit of the LISP Host Mobility solution, The different functional aspects that must be addressed in an IP mobility solution are

• Redirection

The ultimate goal of IP mobility is to steer traffic to the valid location of the end-point. This aspect is generally addressed by providing some sort of re-direction mechanism to enhance the traffic steering already provided by basic routing. Redirection can be achieved by replacing the destination address with a surrogate address that is representative of the new location of the end-point. Different techniques will allow the redirection of traffic either by replacing the destination's address altogether or by leveraging a level of indirection in the addressing such as that achieved with tunnels and encapsulations. The different approaches impact applications to different degrees. The ultimate goal of IP mobility is to provide a solution that is totally transparent to the applications and allows for the preservation of established sessions, as end-points move around the IP infrastructure.

• Scalability

Most techniques create a significant amount of granular state to re-direct traffic effectively. The state is necessary to correlate destination IP addresses to specific locations, either by means of mapping or translation. This additional state must be handled in a very efficient manner to attain a solution that can support a deployable scale at a reasonable cost in terms of memory and processing.

• Optimized Routing

As end-points move around, it is key that traffic is routed to these end-points following the best possible path. Since mobility is based largely on re-direction of traffic, the ability to provide an optimal path is largely a function of the location of the re-directing element. Depending on the architecture, the solution may generate sub-optimal traffic patterns often referred to as traffic triangulation or hair-pinning in an attempt to describe the unnecessary detour traffic needs to take when the destination is mobile. A good mobility solution is one that can provide optimized paths regardless of the location of the end-point.

• Client Independent Solution

It is important that the mobility solution does not depend on agents installed on the mobile end-points or on the clients communicating with these end-points. A network based solution is highly

desirable and is key to the effective deployment of a mobility solution given the precedent of the large installed base of end-points that cannot be changed or managed at will to install client software.

• Address Family Agnostic Solution

The solution provided must work independently of IPv4 or IPv6 end-points and networks. Since mobility relies on the manipulation of the mapping of identity to location, address families with lengthier addresses tend to provide alternatives not available with smaller address spaces. These address dependent solutions have limited application as they usually call for an end to end deployment of IPv6. To cover the broad installed base of IPv4 networking and end-points, the ideal solution should work for IPv4 or IPv6 independently.

Existing IP Mobility Solutions

The following IP Mobility technology solutions are available and described below:

• Route Health Injection (RHI) and Host Routing

• Mobile IPv4

• Mobile IPv6

• DNS Based Redirection: Global Site Selector (GSS)

Route Health Injection (RHI) and Host Routing

One simple way to redirect traffic to a new location when a server (or group of servers) moves is to inject a more specific route to the moved end-point(s) into the routing protocol when the moves are detected. In the extreme case, this means injecting a host route from the "landing" location every time a host moves. Load balancers with the Route Health Injection (RHI) functionality implemented can provide an automated mechanism to detect server moves and inject the necessary host routes when the servers move.

This approach, although simple, pollutes the routing tables considerably and causes large amount of churn in the routing protocol. Forcing churning of the routing protocol is a risky proposition as it could lead to instabilities and overall loss of connectivity, together with adding delays to roaming handoffs.

Mobile IPv4

Mobile IP is defined for IPv4 in IETF RFC 3344. Basically mobile IPv4 provides a mechanism to redirect traffic to a mobile node whenever this node moves from its "Home Network" to a "Foreign Network." Every host will have a "Home Address" within a "Home Network" which is front-ended by a router that acts as a "Home Agent" and that advertises the "Home Network" into the routing protocol. Traffic destined to the "Home Address" will always be routed to the "Home Agent." If the mobile node is in its "Home Network" traffic will be forwarded directly in the data plane to the host

as per regular routing. If the host has moved to a "Foreign Network", traffic will be IP tunneled by the "Home Agent" to a "Care-of- Address" which is the address of the gateway router for the "Foreign Network." With Mobile IPv4 there is always a triangular traffic pattern. Also, Mobile IPv4 does not offer a solution for multicast. Since the mobile node is usually sourcing traffic, if the Foreign Agent is not directly connected, there is the need for host route injection at the foreign site to get RPF to work. In addition, multicast traffic from the mobile node has to always hairpin through the home agent since the distribution tree is built and rooted at the "Home Agent."

Mobile IPv6

IETF RFC 3775 defines mobility support in IPv6. IPv6 takes a step beyond IPv4 mobility and provides optimal data paths between server and client. The process in IPv6 is similar to that of IPv4 with a few additions. Rather than having the Home Agent always redirect the traffic to the Care-of-Address (CoA) for the server that has moved, the Home Agent is taken out of the data path by distributing the CoA to Home Address Binding information to the client itself. Once the client has the CoA information for a particular server, it can send traffic directly to the CoA rather than triangulating it through the Home Address. This provides a direct path from client to server. Although Mobile IPv6 provides direct path routing for mobile nodes, it is limited to IPv6 enabled end-points, it requires that the entire data path be IPv6 enabled, and it also requires that the end-points have IPv6 mobility agents installed on them.

DNS Based Redirection: Global Site Selector (GSS)

It may be possible to direct traffic to a moving server by updating the DNS entries for the moving server as the server moves locations. This scheme assumes that every time a server moves it is assigned a new IP address within the server's "landing" subnet. When the server moves, its DNS entry is updated to reflect the new IP address. Any new connections to the server will use the new IP address that is learnt via DNS resolution. Thus traffic is redirected by updating the mapping of the DNS name (identity) to the new IP address (location). The new IP address assigned after the move may be assigned directly to the server or may be a new Virtual IP (VIP) on a load balancer front-ending the server at the new location. When using load balancers at each location, the load balancers can be leveraged to determine the location of a host by checking the servers' health with probes. When a change of location is detected, the integration of workflow in vCenter (VMware) updates the Global Site Selector (GSS) of the new VIP for the server and the GSS will in turn proceed to update the DNS system with the new VIP to server-name mapping. Established connections will continue to try to reach the original VIP, it is up to the load balancers to be able to re-direct those connections to the new host location and create a hair-pinned traffic pattern for the previously established connections. New connections will be directed to the new VIP (provided the DNS cache has been updated on the client) and will follow a direct path to this new VIP. Eventually all old connections are completed and there are no hair-pinned flows.

The main caveats with this approach include:

• Rate of refresh for the DNS cache may impact either the convergence time for the move or the scalability of the DNS system if the rate is too high.

• Works only for name-based connections while many applications are moving to an IP based connection model.

• Previously established connections are hair-pinned. This implies that there is a period of time where there are active connections to the old address and some new connections to the new address in the second data center. During this state the network administrator may not be able to ascertain that these two addresses are the same system (from the point of view of the application).

NETWORK FUNCTIONALITY VIRTUALIZATION

Network Function Virtualization, or NFV, is a way to reduce cost and accelerate service deployment for network operators by decoupling functions like a firewall or encryption from dedicated hardware and moving them to virtual servers.

Instead of installing expensive proprietary hardware, service providers can purchase inexpensive switches, storage and servers to run virtual machines that perform network functions. This collapses multiple functions into a single physical server, reducing costs and minimizing truck rolls.

If a customer wants to add a new network function, the service provider can simply spin up a new virtual machine to perform that function.

For example, instead of deploying a new hardware appliance across the network to enable network encryption, encryption software can be deployed on a standardized server or switch already in the network.

This virtualization of network functions reduces dependency on dedicated hardware appliances for network operators, and allows for improved scalability and customization across the entire network. Different from a virtualized network, NFV seeks to offload network functions only, rather than the entire network.

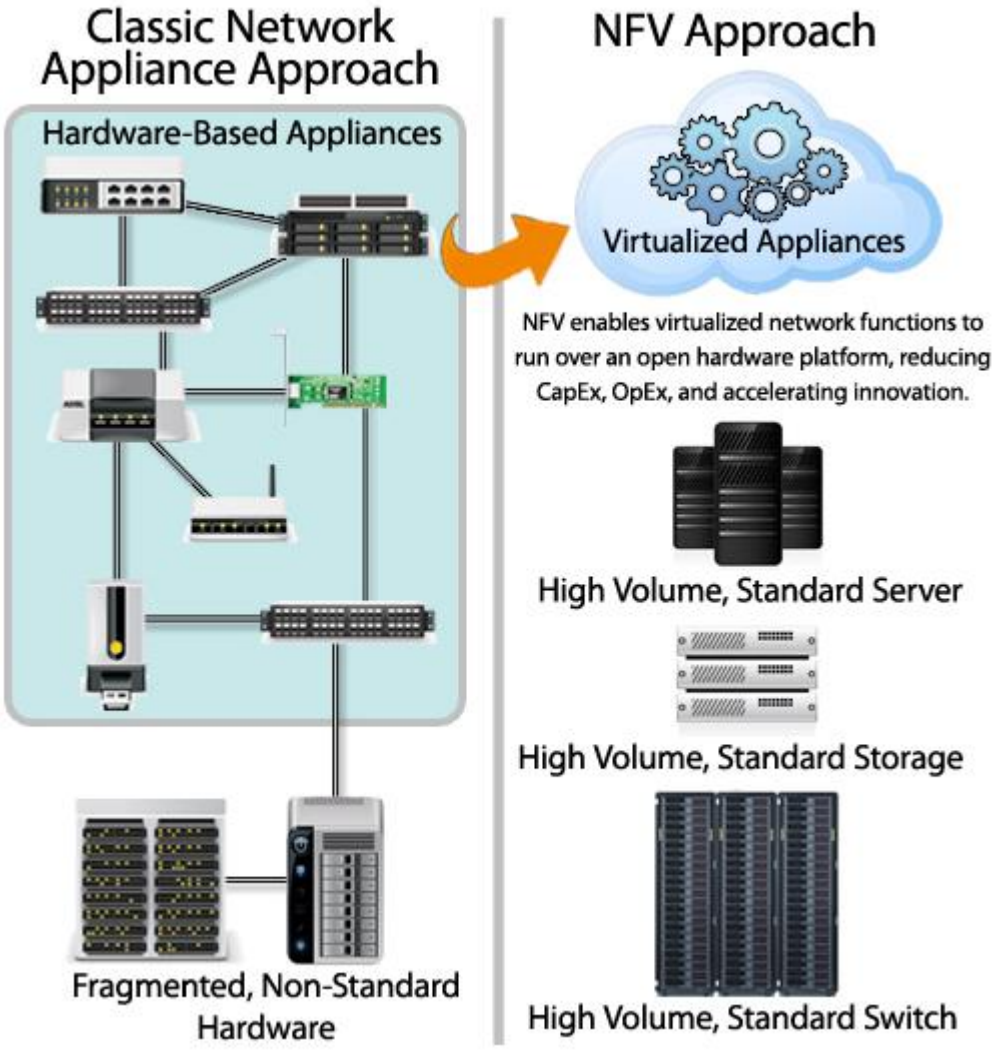


Fig.2.3 : Network Function Virtualization architecture

NFV architecture

The NFV architecture proposed by the European Telecommunications Standards Institute (ETSI) is helping to define standards for NFV implementation. Each component of the architecture is based on these standards to promote better stability and interoperability.

NFV architecture consists of:

- Virtualized network functions (VNFs)are software applications that deliver network functions such as file sharing, directory services, and IP configuration.
- Network functions virtualization infrastructure (NFVi) consists of the infrastructure components—compute, storage, networking—on a platform to support software, such as a hypervisor like KVM or a container management platform, needed to run network apps.
- Management, automation and network orchestration (MANO) provides the framework for managing NFV infrastructure and provisioning new VNFs.

Software-defined networking (SDN) and NFV

NFV and SDN are not dependent on each other, but they do have similarities. Both rely on virtualization and use network abstraction, but how they separate functions and abstract resources is different.

SDN separates network forwarding functions from network control functions with the goal of creating a network that is centrally manageable and programmable. NFV abstracts network functions from hardware. NFV supports SDN by providing the infrastructure on which SDN software can run.

NFV and SDN can be used together, depending on what you want to accomplish, and both use commodity hardware. With NFV and SDN, you can create a network architecture that is more flexible, programmable, and uses resources efficiently.

The benefits of using NFV

There are plenty of reasons for organizations to use NFV, including the following benefits:

- Better communication
- Reduced costs
- Improved flexibility and accelerated time to market for new products and updates
- Improved scalability and resource management
- Reduced vendor lock-in

Better communication and information accessibility

In addition to managing networks, NFV improves network function by transforming how the network architects generate network services. This process is performed by using an architectural and creatively designed method to link together different network nodes to produce a communication channel that can provide freely accessible information to users.

Reduced costs

Often used to great effect for decoupling network services, NFV can also be used as an alternative for routers, firewalls and load balancers. One of the appeals of NFV over routers, firewalls and load balancers is that it doesn't require network proprietors to purchase dedicated hardware devices to perform their work or generate service chains or groups. This benefit helps to reduce the cost of operating expenses and allows work to be performed with fewer potential operating issues.

Improved scalability

Because VMs have virtualized services, they can receive portions of the virtual resources on x86 servers, allowing multiple VMs to run from a single server and better scale, based on the remaining

resources. This advantage helps direct unused resources to where they're needed and boosts efficiency for data centers with virtualized infrastructures.

NFV allows networks the ability to quickly and easily scale their resources based off of incoming traffic and resource requirements. And software-defined networking (SDN) software lets VMs automatically scale up or down.

Better resource management

Once a data center or similar infrastructure is virtualized, it can do more with fewer resources because a single server can run different VNFs simultaneously to produce the same amount of work. It allows for an increased workload capacity while reducing the data center footprint, power consumption and cooling needs.

Flexibility and accelerated time to market

NFV helps organizations update their infrastructure software when network demands change, starkly reducing the need for physical updates. As business requirements change and new market opportunities open, NFV helps organizations quickly adapt. Because a network's infrastructure can be altered to better support a new product, the time-to-market period can be shortened.

Reduced vendor lock-in

The largest benefit of running VNFs on COTS hardware is that organizations aren't chained to proprietary, fixed-function boxes that take truck rolls and lots of time and labor for deployment and configuration.

Fog Computing

Fog computing, also called fog networking or fogging, describes a decentralized computing structure located between the cloud and devices that produce data. This flexible structure enables users to place resources including applications and the data they produce, in logical locations to enhance performance.

Working of Fog computing

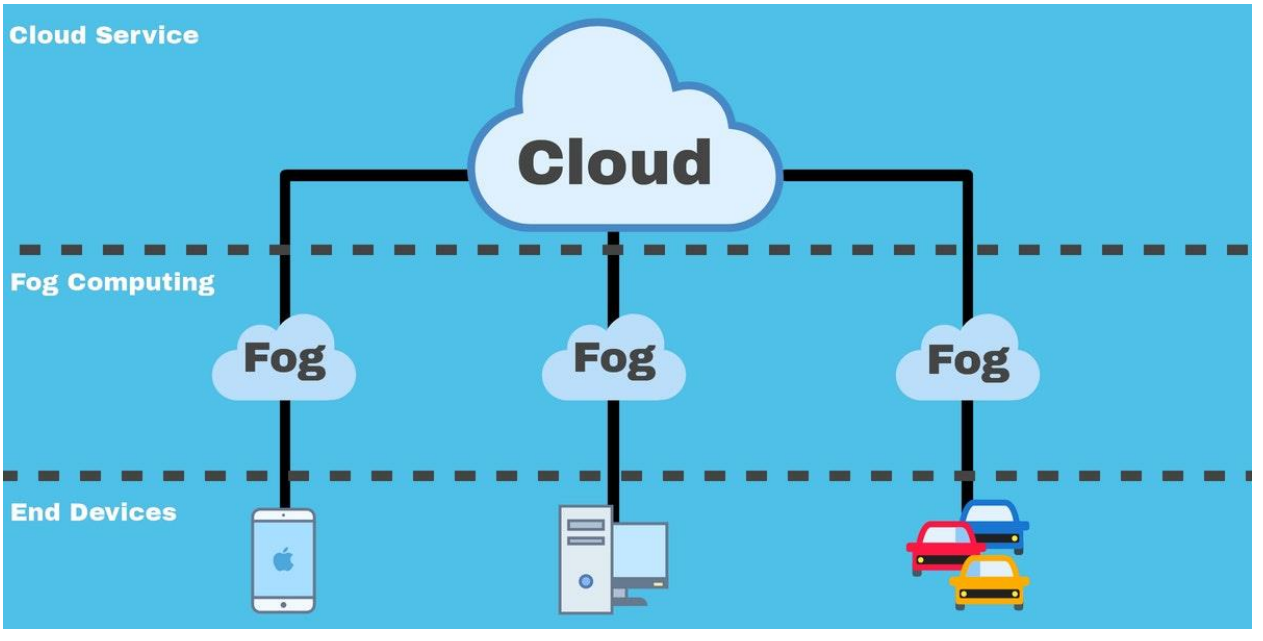


Fig.2.4 : Fog Computing

Fog computing works by deploying fog nodes throughout your network. Devices from controllers, switches, routers, and video cameras can act as fog nodes. These fog nodes can then be deployed in target areas such as your office floor or within a vehicle. When an IoT device generates data this can then be analyzed via one of these nodes without having to be sent all the way back to the cloud. The main difference between cloud computing and fog computing is that the former provides centralized access to resources whereas the latter provides a decentralized local access.

Transporting data through fog computing has the following steps:

- Signals from IoT devices are wired to an automation controller which then executes a control system program to automate the devices.
- The control system program sends data through to an OPC server or protocol gateway.
- The data is then converted into a protocol that can be more easily understood by internet-based services (Typically this is a protocol like HTTP or MQTT).

- Finally, the data is sent to a fog node or IoTgateway which collects the data for further analysis. This will filter the data and in some cases save it to hand over to the cloud later.

Features between fog, edge & cloud computing

Features	FOG computing	Edge computing	Cloud computing
Availability of server nodes	Availability high range of servers	Less scalable than fog computing	Availability of few servers
Type of services	Distributed and localized limited and special for specific domain	Mostly uses in cellular mobile networks	Worldwide and global services
Location identification	Yes	Yes	No
Mobility features	Provided and fully supported	Provided and partially supported	Limited
Real-time interaction	Supported	Supported	Supported
Real-time response	Highest	Higher	Lower
Big data storage & duration	Short duration and targeted to specific area	Depends on the scenario of services and applications	Life time duration as its managing for big data
Big data analytic capacity and computation quality	Short time capacity with high level computation functionality	Short time capacity for prioritized computing facilities	Long time capacity only with categorization computing facilities
Working environment & positions	Streets, roadside, home, malls, field tracks (e.g., every Internet existing areas)	Deployed by the specific services provider in specific indoor areas	Indoors with massive components at cloud service provider owned place
Architectural design	Distributed	Distributed	Centralized
Number of users facilitated	Locally related fields (e.g., IIoT, STL devices)	Specific related fields (e.g., mobile users)	General Internet connected users
Major service provided	Cisco IOx, Intel	Cellular network companies	Google, Amazon, IBM, and Microsoft Azure

Applications of fog computing

- Linked vehicles: Self-driven or self-driven vehicles are now available on the market, producing a significant volume of data. The information has to be easily interpreted and processed based on the information presented such as traffic, driving conditions, environment, etc. All this information is processed quickly with the aid of fog computing.
- Smart Grids and Smart Cities: Energy networks use real-time data for the efficient management of systems. It is necessary to process the remote data near to the location where it is produced. It is also likely that data from multiple sensors will be produced. Fog computing is constructed in such a manner that all problems can be sorted.

- Real-time analytics: Data can be transferred using fog computing deployments from the location where it is produced to different locations. Fog computing is used for real-time analytics that passes data to financial institutions that use real-time data from production networks.

Characteristics of Fog

Cognition:

Cognition is responsiveness to client centric objectives. Fog based data access and analytics give a better alert about customer requirements, best position handling for where to transmit, store, and control functions throughout cloud to the IoT continuum. Applications, due to close proximity, at end devices provide a better conscious and responsive reproduced customer requirement relation.

Heterogeneity:

Fog computing is a virtualized structure so it offers computational, storage, and networking services between the main cloud and devices at the end. Its heterogeneity featured servers consist of hierarchical building blocks at distributed positions.

Geographical Environment Distribution:

Fog computing environment has a widely circulated deployment in context to provide QoS for both mobiles and motionless end devices. Fog network distributes geographically its nodes and sensors in the scenario of different phase environment, for example, temperature monitoring at chemical vat, weather monitoring sensors, STLS sensors, and health care monitoring system.

Edge Location with Low Latency:

The coming out smart applications services are inadequate due to the lack of support at the proximity of devices with featured QoS at the edge of the core network. Video streaming in small TV support devices, monitoring sensors, live gaming applications.

Real-Time Interaction:

Real-time interaction is a variety and requirement of fog applications, like monitoring a critical process at oil rig with the fog edge devices or sensors, real-time transmission for traffic monitoring systems, electricity distribution monitoring system applications, and so on. Fog applications are having real-time processing capabilities for QoS rather than batch processing.

Large Scale Sensor Network:Fog has a feature applicable when environment monitoring system, in near smart grid applications, inherently extends its monitoring systems caused by hierarchical computing and storage resource requirements.

Widespread Wireless Access:

In this scenario wireless access protocols (WAP) and cellular mobile gateways can be classical examples as fog node proximity to the end users.

Interoperable Technology:

Fog components must be able to work in interoperating environment to guarantee support for wide range of services like data streaming and real-time processing for best data analyses and predictive decisions.

Benefits or Advantages of Fog computing

➥It offers better security. Fog nodes can be protected using same procedures followed in IT environment.

➥It processes selected data locally instead of sending them to the cloud for processing. Hence it can save network bandwidth. This leads to lower operational costs.

➥It reduces latency requirements and hence quick decisions can be made. This helps in avoiding accidents.

➥It offers better privacy to the users data as they are analyzed locally instead of sending them to the cloud. Moreover IT team can manage and control the devices.

➥It is easy to develop fog applications using right tools which can drive machines as per customers need.

➥Fog nodes are mobile in nature. Hence they can join and leave the network at any time.

➥Fog nodes can withstand harsh environmental conditions in places such as tracks, vehicles, under sea, factory floors etc. Moreover it can be installed in remote locations.

➥Fog computing offers reduction in latency as data are analyzed locally. This is due to less round trip time and less amount of data bandwidth.

Disadvantages of Fog computing

➥Encryption algorithms and security policies make it more difficult for arbitrary devices to exchange data. Any mistakes in security algorithms lead to exposure of data to the hackers. Other security issues are IP address spoofing, man in the middle attacks, wireless network security etc.

➥To achieve high data consistency in the the fog computing is challenging and requires more efforts.

➥Fog computing will realize global storage concept with infinite size and speed of local storage but data management is a challenge.

➥Trust and authentication are major concerns.

➡Scheduling is complex as tasks can be moved between client devices, fog nodes and back end cloud servers.

➡Power consumption is high in fog nodes compare to centralized cloud architecture.

Cloud Computing

cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.

Benefits of Cloud Computing

Flexibility

Users can scale services to fit their needs, customize applications and access cloud services from anywhere with an internet connection.

Efficiency

Enterprise users can get applications to market quickly, without worrying about underlying infrastructure costs or maintenance.

Strategic value

Cloud services give enterprises a competitive advantage by providing the most innovative technology available.

Flexibility

- Scalability: Cloud infrastructure scales on demand to support fluctuating workloads.
- Storage options: Users can choose public, private, or hybrid storage offerings, depending on security needs and other considerations.
- Control choices: Organizations can determine their level of control with as-a-service options. These include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).
- Tool selection: Users can select from a menu of prebuilt tools and features to build a solution that fits their specific needs.
- Security features: Virtual private cloud, encryption, and API keys help keep data secure.

Efficiency

- Accessibility: Cloud-based applications and data are accessible from virtually any internet-connected device.
- Speed to market: Developing in the cloud enables users to get their applications to market quickly.
- Data security: Hardware failures do not result in data loss because of networked backups.
- Savings on equipment: Cloud computing uses remote resources, saving organizations the cost of servers and other equipment.
- Pay structure: A "utility" pay structure means users only pay for the resources they use.

Strategic value

- Streamlined work: Cloud service providers (CSPs) manage underlying infrastructure, enabling organizations to focus on application development and other priorities.
- Regular updates: Service providers regularly update offerings to give users the most up-to-date technology.
- Collaboration: Worldwide access means teams can collaborate from widespread locations.
- Competitive edge: Organizations can move more nimbly than competitors who must devote IT resources to managing infrastructure.

Types of cloud computing

- Not all clouds are the same and not one type of cloud computing is right for everyone. Several different models, types and services have evolved to help offer the right solution for your needs.
- First, you need to determine the type of cloud deployment or cloud computing architecture, that your cloud services will be implemented on. There are three different ways to deploy cloud services: on a public cloud, private cloud or hybrid cloud.

Hybrid cloud

- A hybrid cloud is a type of cloud computing that combines on-premises infrastructure—or a private cloud—with a public cloud. Hybrid clouds allow data and apps to move between the two environments.
- Many organisations choose a hybrid cloud approach due to business imperatives such as meeting regulatory and data sovereignty requirements, taking full advantage of on-premises technology investment or addressing low latency issues.

- The hybrid cloud is evolving to include edge workloads as well. Edge computing brings the computing power of the cloud to IoT devices—closer to where the data resides. By moving workloads to the edge, devices spend less time communicating with the cloud, reducing latency and they are even able to operate reliably in extended offline periods.

Advantages of the hybrid cloud

- Control—your organisation can maintain a private infrastructure for sensitive assets or workloads that require low latency.
- 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.

Public cloud

- Public clouds are the most common type of cloud computing deployment. The cloud resources (like servers and storage) are owned and operated by a third-party cloud service provider and delivered over the internet. With a public cloud, all hardware, software and other supporting infrastructure are owned and managed by the cloud provider. Microsoft Azure is an example of a public cloud.
- In a public cloud, you share the same hardware, storage and network devices with other organisations or cloud "tenants," and you access services and manage your account using a web browser. Public cloud deployments are frequently used to provide web-based email, online office applications, storage and testing and development environments.

Advantages of public cloud

- Lower costs—no need to purchase hardware or software and you pay only for the service you use.
- No maintenance—your service provider provides the maintenance.
- Near-unlimited scalability—on-demand resources are available to meet your business needs.
- High reliability—a vast network of servers ensures against failure.

Private cloud

- A private cloud consists of cloud computing resources used exclusively by one business or organisation. The private cloud can be physically located at your organisation's on-site datacenter or it can be hosted by a third-party service provider. But in a private cloud, the

services and infrastructure are always maintained on a private network and the hardware and software are dedicated solely to your organisation.

- In this way, a private cloud can make it easier for an organisation to customise its resources to meet specific IT requirements. Private clouds are often used by government agencies, financial institutions, any other mid- to large-size organisations with business-critical operations seeking enhanced control over their environment.

Advantages of a private cloud

- More flexibility—your organisation can customise its cloud environment to meet specific business needs.
- More control—resources are not shared with others, so higher levels of control and privacy are possible.
- More scalability—private clouds often offer more scalability compared to on-premises infrastructure.

BIG DATA AND ANALYTICS

DATA

The quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.

BIG DATA

Big data is a combination of structured, semistructured and unstructured data collected by organizations that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics applications.

Importance of Big data

Companies use big data in their systems to improve operations, provide better customer service, create personalized marketing campaigns and take other actions that, ultimately, can increase revenue and profits. Businesses that use it effectively hold a potential competitive advantage over those that don't because they're able to make faster and more informed business decisions.

For example, big data provides valuable insights into customers that companies can use to refine their marketing, advertising and promotions in order to increase customer engagement and conversion rates. Both historical and real-time data can be analyzed to assess the evolving preferences of consumers or corporate buyers, enabling businesses to become more responsive to customer wants and needs.

Types of Big Data

Following are the types of Big Data:

1. Structured
2. Unstructured
3. Semi-structured

Structured

Any data that can be stored, accessed and processed in the form of fixed format is termed as a 'structured' data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the rage of multiple zetta bytes.

Examples of Structured Data

An 'Employee' table in a database is an example of Structured Data

Employee_ID	Employee_Name	Gender	Department	Salary_In_lacs
2365	Rajesh Kulkarni	Male	Finance	650000
3398	Pratibha Joshi	Female	Admin	650000
7465	Shushil Roy	Male	Admin	500000
7500	Shubhojit Das	Male	Finance	500000
7699	Priya Sane	Female	Finance	550000

Unstructured

Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, un-structured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format.

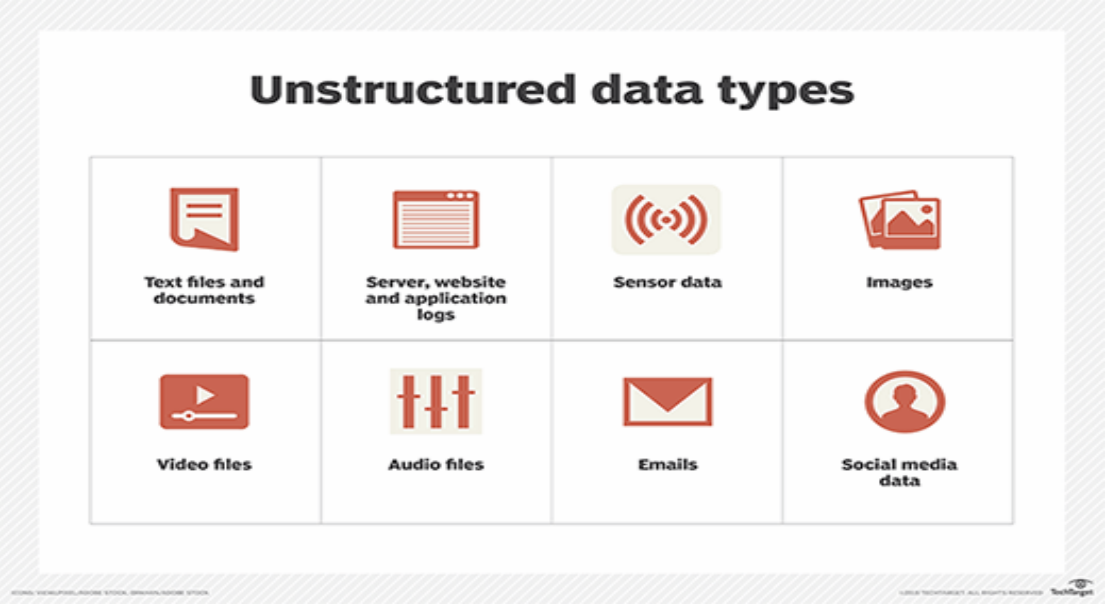


Fig.2.5 : Unstructured data types

Semi-structured

Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in an XML file.

Characteristics of Big Data

Big data can be described by the following characteristics:

- Volume
- Variety
- Velocity
- Variability

(i) Volume **–** The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, 'Volume' is one characteristic which needs to be considered while dealing with Big Data solutions.

(ii) Variety **–** The next aspect of Big Data is its variety. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

(iii) Velocity – The term 'velocity' refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.

Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors, Mobile devices, etc. The flow of data is massive and continuous.

(iv) Variability – This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Types of data that comes under big data.

- Black box data: The black box of aeroplane , jets,Helicopter are used to store microphone voices, Performance information etc.
- Social media data: Different social media websites Hold information about various users.
- Stock exchange data: It holds information about Buy and sell shares etc.
- Transport data: The transport data holds information About model, capacity, distance and many other things of vehicles.
- Search engine data: Different search engines retrieve data from different database.

Advantages of Big Data Processing

Ability to process Big Data in DBMS brings in multiple benefits, such as-

- Businesses can utilize outside intelligence while taking decisions

Access to social data from search engines and sites like facebook, twitter are enabling organizations to fine tune their business strategies.

- Improved customer service

Traditional customer feedback systems are getting replaced by new systems designed with Big Data technologies. In these new systems, Big Data and natural language processing technologies are being used to read and evaluate consumer responses.

- Early identification of risk to the product/services, if any
- Better operational efficiency

Big Data technologies can be used for creating a staging area or landing zone for new data before identifying what data should be moved to the data warehouse. In addition, such integration of Big Data technologies and data warehouse helps an organization to offload infrequently accessed data.

Different Types of Big Data Analytics

Here are the four types of Big Data analytics:

1. Descriptive Analytics

This summarizes past data into a form that people can easily read. This helps in creating reports, like a company's revenue, profit, sales, and so on. Also, it helps in the tabulation of social media metrics.
Use Case: The Dow Chemical Company analyzed its past data to increase facility utilization across its office and lab space. Using descriptive analytics, Dow was able to identify underutilized space. This space consolidation helped the company save nearly US $4 million annually.

2. Diagnostic Analytics

This is done to understand what caused a problem in the first place. Techniques like drill-down, data mining, and data recovery are all examples. Organizations use diagnostic analytics because they provide an in-depth insight into a particular problem.
Use Case: An e-commerce company's report shows that their sales have gone down, although customers are adding products to their carts. This can be due to various reasons like the form didn't load correctly, the shipping fee is too high, or there are not enough payment options available. This is where you can use diagnostic analytics to find the reason.

3. Predictive Analytics

This type of analytics looks into the historical and present data to make predictions of the future. Predictive analytics uses data mining, AI, and machine learning to analyze current data and make predictions about the future. It works on predicting customer trends, market trends, and so on.
Use Case: PayPal determines what kind of precautions they have to take to protect their clients against fraudulent transactions. Using predictive analytics, the company uses all the historical payment data and user behavior data and builds an algorithm that predicts fraudulent activities.

4. Prescriptive Analytics

This type of analytics prescribes the solution to a particular problem. Perspective analytics works with both descriptive and predictive analytics. Most of the time, it relies on AI and machine learning.
Use Case: Prescriptive analytics can be used to maximize an airline's profit. This type of analytics is used to build an algorithm that will automatically adjust the flight fares based on numerous factors, including customer demand, weather, destination, holiday seasons, and oil prices.

Big Data Analytics Tools

Here are some of the key big data analytics tools :

- Hadoop - helps in storing and analyzing data

- MongoDB - used on datasets that change frequently
- Talend - used for data integration and management
- Cassandra - a distributed database used to handle chunks of data
- Spark - used for real-time processing and analyzing large amounts of data
- STORM - an open-source real-time computational system
- Kafka - a distributed streaming platform that is used for fault-tolerant storage

Big Data Industry Applications

Here are some of the sectors where Big Data is actively used:

- Ecommerce - Predicting customer trends and optimizing prices are a few of the ways e-commerce uses Big Data analytics
- Marketing - Big Data analytics helps to drive high ROI marketing campaigns, which result in improved sales
- Education - Used to develop new and improve existing courses based on market requirements
- Healthcare - With the help of a patient's medical history, Big Data analytics is used to predict how likely they are to have health issues
- Media and entertainment - Used to understand the demand of shows, movies, songs, and more to deliver a personalized recommendation list to its users
- Banking - Customer income and spending patterns help to predict the likelihood of choosing various banking offers, like loans and credit cards
- Telecommunications - Used to forecast network capacity and improve customer experience
- Government - Big Data analytics helps governments in law enforcement, among other things

Challenges In Big Data Analytics

- Uncertainty of Data Management Landscape: Because big data is continuously expanding, there are new companies and technologies that are being developed every day. A big challenge for companies is to find out which technology works bests for them without the introduction of new risks and problems.
- The Big Data Talent Gap: While Big Data is a growing field, there are very few experts available in this field. This is because Big data is a complex field and people who understand the complexity and intricate nature of this field are far few and between. Another major challenge in the field is the talent gap that exists in the industry

- Getting data into the big data platform: Data is increasing every single day. This means that companies have to tackle a limitless amount of data on a regular basis. The scale and variety of data that is available today can overwhelm any data practitioner and that is why it is important to make data accessibility simple and convenient for brand managers and owners.
- Need for synchronization across data sources: As data sets become more diverse, there is a need to incorporate them into an analytical platform. If this is ignored, it can create gaps and lead to wrong insights and messages.
- Getting important insights through the use of Big data analytics: It is important that companies gain proper insights from big data analytics and it is important that the correct department has access to this information. A major challenge in big data analytics is bridging this gap in an effective fashion.

M2M LEARNING AND ARTIFICIAL INTELLIGENCE

M2M Learning

Machine-to-machine, or M2M, is a broad label that can be used to describe any technology that enables networked devices to exchange information and perform actions without the manual assistance of humans. Artificial intelligence (AI) and machine learning (ML) facilitate the communication between systems, allowing them to make their own autonomous choices. M2M technology was first adopted in manufacturing and industrial settings, where other technologies, such as SCADA and remote monitoring, helped remotely manage and control data from equipment. M2M has since found applications in other sectors, such as healthcare, business and insurance. M2M is also the foundation for the internet of things (IoT).

Working of M2M

The main purpose of machine-to-machine technology is to tap into sensor data and transmit it to a network. Unlike SCADA or other remote monitoring tools, M2M systems often use public networks and access methods -- for example, cellular or Ethernet -- to make it more cost-effective.

The main components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link, and autonomic computing software programmed to help a network device interpret data and make decisions. These M2M applications translate the data, which can trigger preprogrammed, automated actions.

One of the most well-known types of machine-to-machine communication is telemetry, which has been used since the early part of the last century to transmit operational data. Pioneers in telemetric first used telephone lines, and later, radio waves, to transmit performance measurements gathered from monitoring instruments in remote locations.

The Internet and improved standards for wireless technology have expanded the role of telemetry from pure science, engineering and manufacturing to everyday use in products such as heating units, electric meters and internet-connected devices, such as appliances.

Beyond being able to remotely monitor equipment and systems, the top benefits of M2M include:

- reduced costs by minimizing equipment maintenance and downtime;
- boosted revenue by revealing new business opportunities for servicing products in the field; and
- improved customer service by proactively monitoring and servicing equipment before it fails or only when it is needed.

M2M applications and examples

Machine-to-machine communication is often used for remote monitoring. In product restocking, for example, a vending machine can message the distributor's network, or *machine*, when a particular item is running low to send a refill. An enabler of asset tracking and monitoring, M2M is vital in warehouse management systems (WMS) and supply chain management (SCM).

Utilities companies often rely on M2M devices and applications to not only harvest energy, such as oil and gas, but also to bill customers -- through the use of smart meters -- and to detect worksite factors, such as pressure, temperature and equipment status.

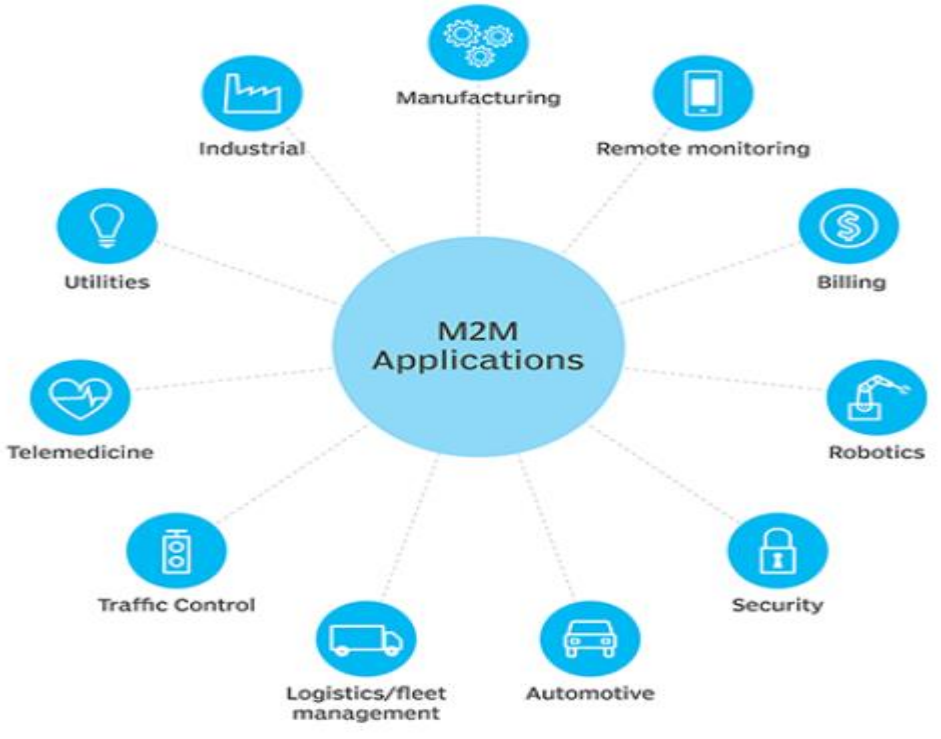


Fig.2.6 : M2M applications

In telemedicine, M2M devices can enable the real time monitoring of patients' vital statistics, dispensing medicine when required or tracking healthcare assets.

The combination of the IoT, AI and ML is transforming and improving mobile payment processes and creating new opportunities for different purchasing behaviors. Digital wallets, such as Google Wallet and Apple Pay, will most likely contribute to the widespread adoption of M2M financial activities.

Smart home systems have also incorporated M2M technology. The use of M2M in this embedded system enables home appliances and other technologies to have real time control of operations as well as the ability to remotely communicate.

M2M is also an important aspect of remote-control software, robotics, traffic control, security, logistics and fleet management and automotive.

Key features of M2M

Key features of M2M technology include:

- Low power consumption, in an effort to improve the system's ability to effectively service M2M applications.
- A Network operator that provides packet-switched service
- Monitoring abilities that provide functionality to detect events.
- Time tolerance, meaning data transfers can be delayed.
- Time control, meaning data can only be sent or received at specific predetermined periods.
- Location specific triggers that alert or wake up devices when they enter particular areas.
- The ability to continually send and receive small amounts of data.

M2M requirements

According to the European Telecommunications Standards Institute (ETSI), requirements of an M2M system include:

- Scalability - The M2M system should be able to continue to function efficiently as more connected objects are added.
- Anonymity - The M2M system must be able to hide the identity of an M2M device when requested, subject to regulatory requirements.
- Logging - M2M systems must support the recording of important events, such as failed installation attempts, service not operating or the occurrence of faulty information. The logs should be available by request.

- M2M application communication principles - M2M systems should enable communication between M2M applications in the network and the M2M device or gateway using communication techniques, such as short message service (SMS) and IP Connected devices should also be able to communicate with each other in a peer-to-peer (P2P) manner.
- Delivery methods - The M2M system should support Unicast, anycast, multicast and broadcast communication modes, with broadcast being replaced by multicast or anycast whenever possible to minimize the load on the communication network.
- Message transmission scheduling - M2M systems must be able to control network access and messaging schedules and should be conscious of M2M applications' scheduling delay tolerance.
- Message communication path selection - Optimization of the message communication paths within an M2M system must be possible and based on policies like transmission failures, delays when other paths exist and network costs.

Artificial Intelligence

The intelligence demonstrated by machines is known as Artificial Intelligence. Artificial Intelligence has grown to be very popular in today's world. It is the simulation of natural intelligence in machines that are programmed to learn and mimic the actions of humans. These machines are able to learn with experience and perform human-like tasks. As technologies such as AI continue to grow, they will have a great impact on our quality of life. It's but natural that everyone today wants to connect with AI technology somehow, may it be as an end-user or pursuing a career in Artificial Intelligence.

Working of Artificial Intelligence (AI)

Building an AI system is a careful process of reverse-engineering human traits and capabilities in a machine, and using it's computational prowess to surpass what we are capable of. To understand How Aritificial Intelligence actually works, one needs to deep dive into the various sub domains of Artificial Intelligence and understand how those domains could be applied into the various fields of the industry.

- Machine Learning : ML teaches a machine how to make inferences and decisions based on past experience. It identifies patterns, analyses past data to infer the meaning of these data points to reach a possible conclusion without having to involve human experience. This automation to reach conclusions by evaluating data, saves a human time for businesses and helps them make a better decision.
- Deep Learning : Deep Learning ia an ML technique. It teaches a machine to process inputs through layers in order to classify, infer and predict the outcome.

- Neural Networks : Neural Networks work on the similar principles as of Human Neural cells. They are a series of algorithms that captures the relationship between various underying variabes and processes the data as a human brain does.
- Natural Language Processingc: NLP is a science of reading, understanding, interpreting a language by a machine. Once a machine understands what the user intends to communicate, it responds accordingly.
- Computer Vision : Computer vision algorithms tries to understand an image by breaking down an image and studying different parts of the objects. This helps the machine classify and learn from a set of images, to make a better output decision based on previous observations.
- Cognitive Computing : Cognitive computing algorithms try to mimic a human brain by anaysing text/speech/images/objects in a manner that a human does and tries to give the desired output.

Advantages of Artificial Intelligence

There's no doubt in the fact that technology has made our life better. From music recommendations, map directions, mobile banking to fraud prevention, AI and other technologies have taken over. There's a fine line between advancement and destruction. There's always two sides to a coin, and that is the case with AI as well. Let us take a look at some advantages of Artificial Intelligence-

Advantages of Artificial Intelligence (AI)

- Reduction in human error
- Available 24×7
- Helps in repetitive work
- Digital assistance
- Faster decisions
- Rational Decision Maker
- Medical applications
- Improves Security
- Efficient Communication

Top Used Applications in Artificial Intelligence

1. Google's AI-powered predictions (E.g.: Google Maps)
2. Ride-sharing applications (E.g.: Uber, Lyft)
3. AI Autopilot in Commercial Flights
4. Spam filters on E-mails
5. Plagiarism checkers and tools
6. Facial Recognition
7. Search recommendations
8. Voice-to-text features
9. Smart personal assistants (E.g.: Siri, Alexa)
10. Fraud protection and prevention.

Text/Reference Books

1. S. Misra, A. Mukherjee, and A. Roy, Introduction to IoT. Cambridge University Press, 2020
2. S. Misra, C. Roy, and A. Mukherjee, Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.2020
3. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015

Question Bank

PART-A

1. Define Big Data.
2. Distinguish between Structured and unstructured data.
3. Latency is low in fog computing, analyze the reasons.
4. Identify the domains where fog computing is used.
5. Produce an example of Moore's Law.
6. Summarize how MEMS sensors manufactured.
7. Distinguish between artificial Intelligence and Machine learning.

PART-B

1. Explain in detail on how a wireless router works with neat architectural sketch.
2. Design a Virtual Machine to manage the control room of COVID DISASTER MANAGEMENT with your own specifications
3. Describe in details any 3 Applications of AI in INDUSTRY 4.0 with its advantages and disadvantages
4. Design an IOT system to save energy and visualize data using a machine and implement algorithms to tackle problems in the industry.
5. Demonstrate and explain how Chennai can be converted into a smart city with the applications of IOT in smart cities.
6. Discuss in detail the working of Mobile IP.

INDUSTRIAL INTERNET OF THINGS – SECA4005

UNIT – III IIOT REFERENCE ARCHITECTURE

IIOT REFERENCE ARCHITECTURE

Industrial Internet Architecture Framework – Functional Viewpoint – Operational Domain, Information Domain, Application Domain, Business Domain – Implementation View point – Architectural Topology – Three Tier Topology – Data Management.

REFERENCE ARCHITECTURE

A reference architecture provides guidance for the development of system, solution and application architectures. It provides common and consistent definitions for the system of interest, its decompositions and design patterns, and a common specification of implementations.

A reference architecture provides a common framework around more detailed discussions. By staying at a higher level of abstraction, it enables the identification and comprehension of the most important issues and patterns across its applications in many different use cases.

INDUSTRIAL INTERNET REFERENCE ARCHITECTURE

The architecture description and representation are generic and at a high level of abstraction to support the requisite broad industry applicability. The IIRA distills and abstracts common characteristics, features and patterns from use cases.

It will be refined and revised continually as feedback is gathered from its application in the test beds developed in IIC as well as real-world deployment of IIoT systems.

INDUSTRIAL INTERNET ARCHITECTURE FRAMEWORK

Architecture Description

Industrial Internet Consortium defines Industrial Internet Architecture Framework (IIAF). The IIAF identifies conventions, principles and practices for consistent description of IIoT architectures. This standard-based architecture framework facilitates easier evaluation, and systematic and effective resolution of stakeholder concerns. It serves as a valuable resource to guide the development, the documentation and the communication about, the IIRA.

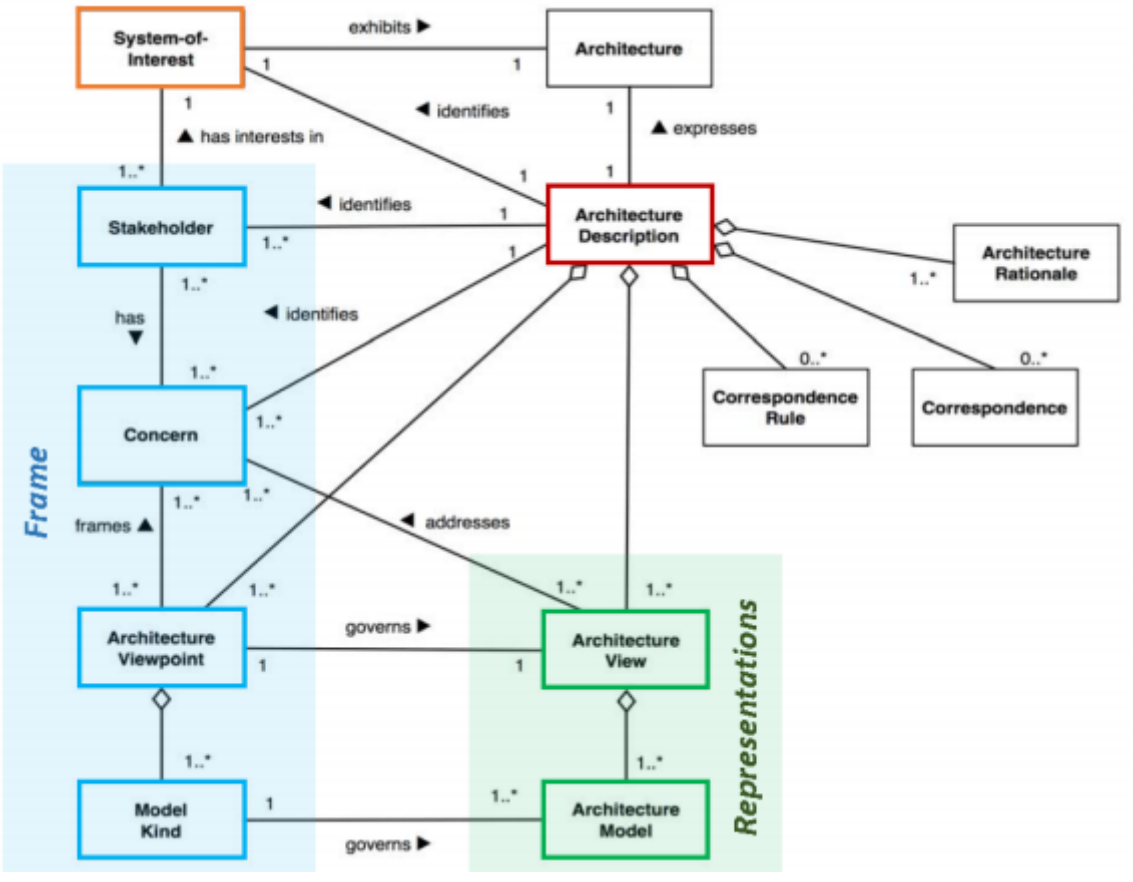


Figure 3.1: Architecture Description

ARCHITECTURE FRAMEWORK

An architecture framework contains information identifying the fundamental architecture constructs and specifies concerns, stakeholders, viewpoints, model kinds, correspondence rules and conditions of applicability.

System architects can use an architecture framework to discover, describe and organize topics of interest (concerns) about the system , they can further use architecture representation to clarify, analyze and resolve these concerns.

The key ideas of architecture framework, its frame and representations, are outlined in Figure.

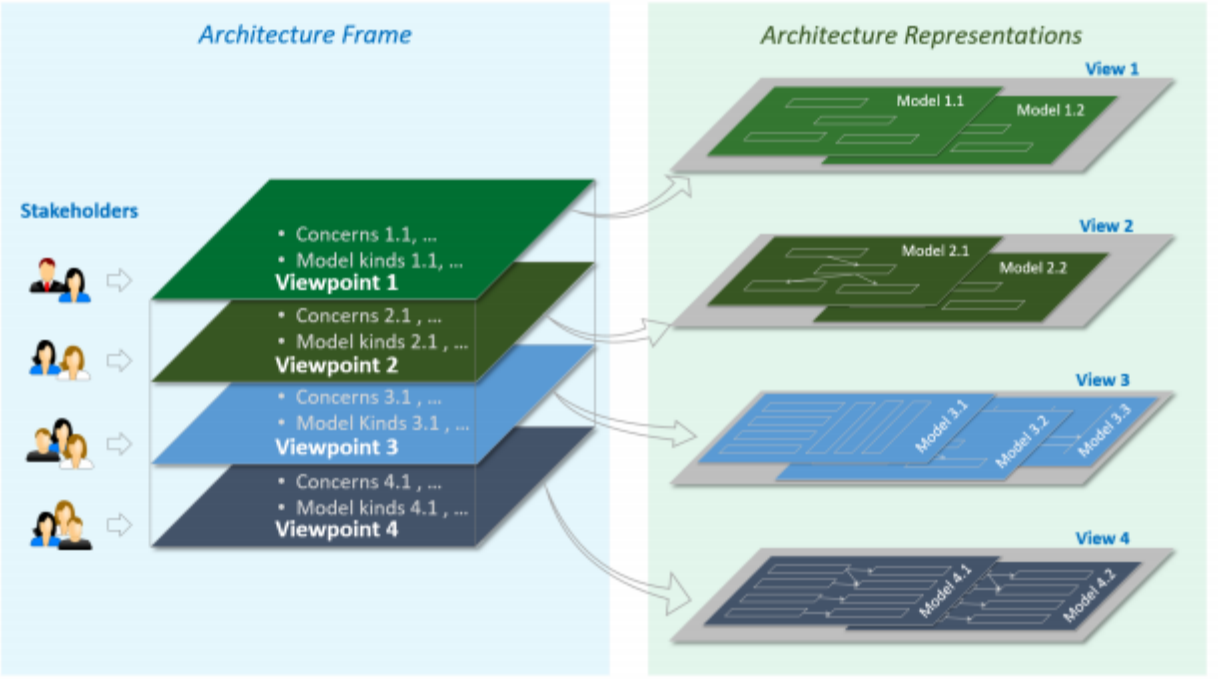


Figure 3.2: Architecture Framework

INDUSTRIAL INTERNET ARCHITECTURE FRAMEWORK

The IIAF adopts the general concepts and constructs in the architecture specification, specifically, concern, stakeholder and viewpoint as its architecture frame, and views and models as its architecture representation in describing and analyzing on important common architecture concerns for IIoT systems. The IIAF is at the foundation of the IIRA.

INDUSTRIAL INTERNET REFERENCE ARCHITECTURE

The IIRA documents the outcome of applying the IIAF to its intended class of systems of interest: Industrial Internet of Things systems. It first identifies and highlights the most important architectural concerns commonly found in IIoT systems across industrial sectors and classifies them into viewpoints along with their respective stakeholders. It then describes, analyzes and, where appropriate, provides guidance to resolve these concerns in these viewpoints, resulting in a certain abstract architecture representations.

Figure 3.3 illustrates the key ideas about the constructs of the Industrial Internet Reference Architecture and its application.

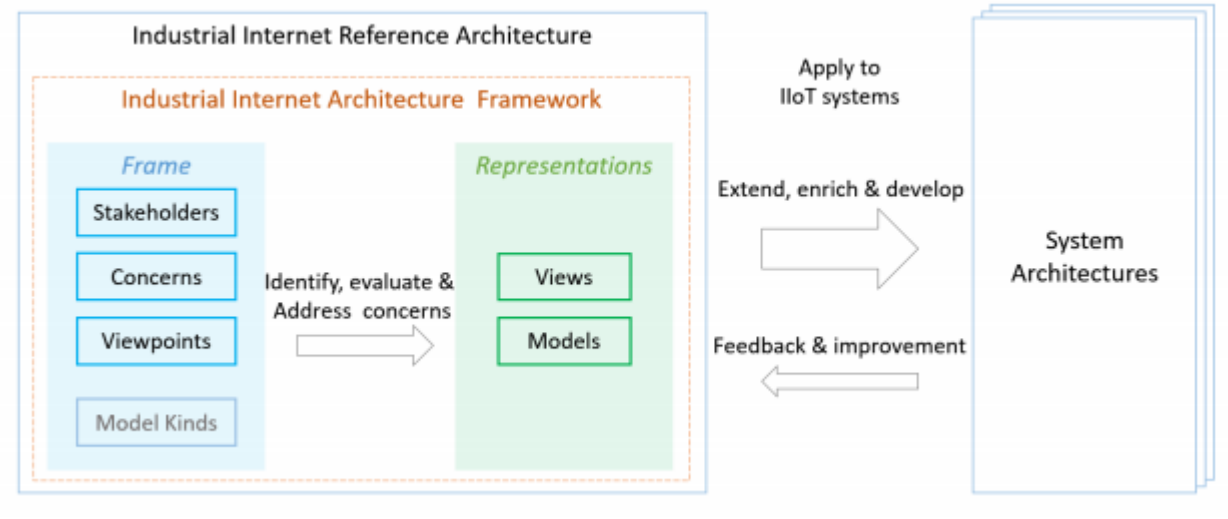


Figure 3.3: IIRA constructs and application

The IIRA is at a level of abstraction that excludes architectural elements whose evaluation requires specificities only available in concrete systems. It does not describe all the architecture constructs as outlined in Figure 3-3: IIRA constructs and application. Instead, it adapts the ISO architecture specification with minor adjustments in two aspects:

- It does not explicitly identify the model kind as a key construct of its framework but uses the concept loosely during the analysis of concerns in developing its representation; and
- It does not explicitly discuss certain architecture constructs, Correspondence Rules, Correspondence and Architecture Rationale, but leaves them to the development of concrete architectures as needed.

Within the IIRA, these models and their representations in the views are chosen because they address the respective concerns at the appropriate level of abstraction and demonstrate the key ideas of this reference architecture. They are not, however, the sole models and views for addressing concerns in the viewpoints, nor at a depth sufficient to implement a real system. The views can be used as a starting point for concrete architecting, then extended, enriched, supplemented or replaced with better ones in accordance with the needs of the specific IIoT system at hand.

INDUSTRIAL INTERNET VIEWPOINTS

The IIRA viewpoints are defined by analyzing the various IIoT use cases developed by the IIC and elsewhere, identifying the relevant stakeholders of IIoT systems and determining the proper framing of concerns.

These four viewpoints are:

- Business
- usage

- functional
- implementation

As shown in Figure 3-4, these four viewpoints form the basis for a detailed viewpoint-by viewpoint analysis of individual sets of IIoT system concerns. Architects who adapt the industrial internet viewpoints as the basis of their architecture may extend them by defining additional viewpoints to organize system concerns based on their specific system requirements.

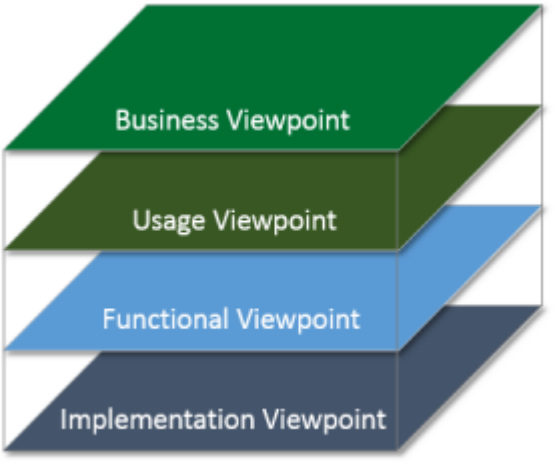


Figure 3.4: Industrial Internet Architecture Viewpoints

BUSINESS VIEWPOINT

The business viewpoint attends to the concerns of the identification of stakeholders and their business vision, values and objectives in establishing an IIoT system in its business and regulatory context. It further identifies how the IIoT system achieves the stated objectives through its mapping to fundamental system capabilities. These concerns are business-oriented and are of particular interest to business decision-makers, product managers and system engineers.

USAGE VIEWPOINT

The usage viewpoint addresses the concerns of expected system usage. It is typically represented as sequences of activities involving human or logical (e.g. system or system components) users that deliver its intended functionality in ultimately achieving its fundamental system capabilities. The stakeholders of these concerns typically consist of system engineers, product managers and the other stakeholders including the individuals who are involved in the specification of the IIoT system under consideration and who represent the users in its ultimate usage.

IMPLEMENTATION VIEWPOINT

The implementation viewpoint deals with the technologies needed to implement functional components (functional viewpoint), their communication schemes and their lifecycle procedures. These elements are coordinated by activities (usage viewpoint) and supportive of the system capabilities (business viewpoint). These concerns are of particular interest to system and component architects, developers and integrators, and system operators.

BUSINESS VIEWPOINT

Business-oriented concerns such as business value, expected return on investment, cost of maintenance and product liability must be evaluated when considering an IIoT system as a solution to business problems. To identify, evaluate and address these business concerns, we introduce a number of concepts and define the relationships between them, as shown in Figure 3.5.

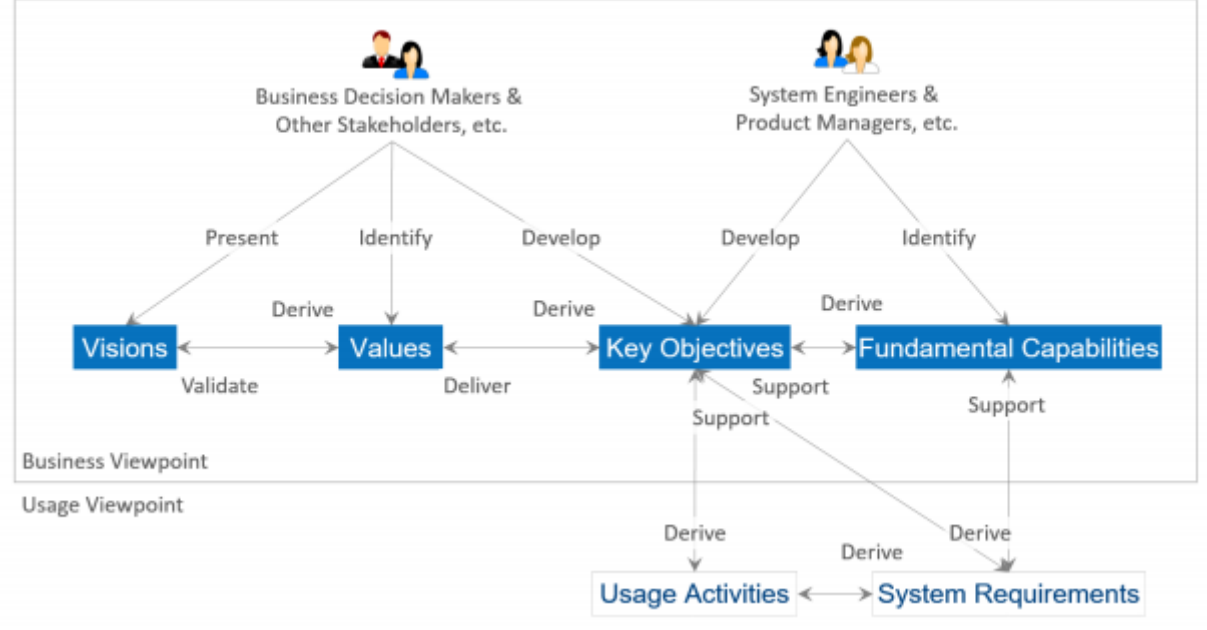


Figure 3.5: A Vision and Value-Driven Model

STAKEHOLDERS

Stakeholders have a major stake in the business and strong influence in its direction. They include those who drive the conception and development of IIoT systems in an organization. They are often recognized as important strategic thinkers and visionaries within a company or an industry. It is important to identify these major stakeholders and engage them early in the process of evaluating these business-oriented concerns.

VISION

Vision describes a future state of an organization or an industry. It provides the business direction toward which an organization executes. Senior business stakeholders usually develop and present an organization's vision.

VALUES

Values reflect how the vision may be perceived by the stakeholders who will be involved in funding the implementation of the new system as well as by the users of the resulting system. These values are typically identified by senior business and technical leaders in an organization. They provide the rationale as to why the vision has merit.

KEY OBJECTIVES

Key objectives are quantifiable high-level technical and ultimately business outcomes expected of the resultant system in the context of delivering the values. Key objectives should be measurable and time-bound. Senior business and technical leaders develop the key objectives.

FUNDAMENTAL CAPABILITIES

Fundamental capabilities refer to high-level specifications of the essential ability of the system to complete specific major business tasks. Key objectives are the basis for identifying the fundamental capabilities. Capabilities should be specified independently of how they are to be implemented (neutral to both the architecture and technology choices) so that system designers and implementers are not unduly constrained at this stage.

The process for following this approach is for the stakeholders to first identify the vision of the organization and then how it could improve its operations through the adoption of an IIoT system. From the vision, the stakeholders establish the values and experiences of the IIoT system under consideration and develop a set of key objectives that will drive the implementation of the vision. From the objectives, the stakeholders derive the fundamental capabilities that are required for the system.

To verify that the resultant system indeed provides the desired capabilities meeting the objectives, they should be characterized by detailed quantifiable attributes such as the degree of safety, security and resilience, benchmarks to measure the success of the system, and the criteria by which the claimed system characteristics can be supported by appropriate evidence.

USAGE VIEWPOINT

The usage viewpoint is concerned with how an IIoT system realizes the key capabilities identified in the business viewpoint. The usage viewpoint describes the activities that coordinate various units of work over various system components. These activities—describing how the system is used—serve as an input for system requirements including those on key system characteristics and guide the design, implementation, deployment, operations and evolution of the IIoT system.

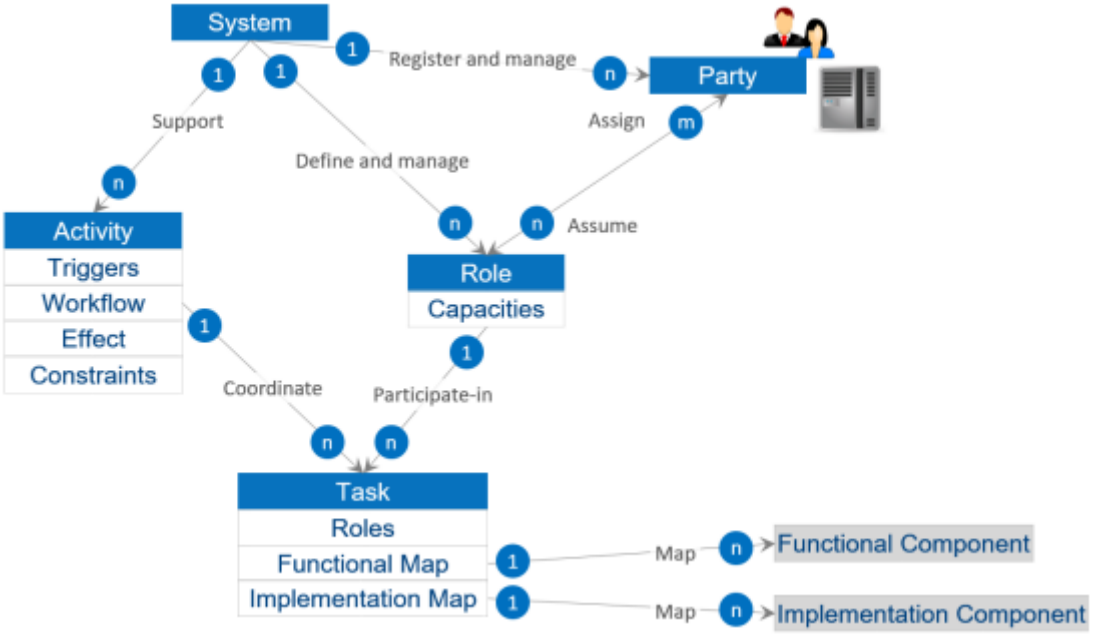


Figure 3.6: Role, Party, Activity and Task

Figure 3.6 depicts the usage viewpoint's main concepts and how they relate to each other.

The basic unit of work is a task, such as the invocation of an operation, a transfer of data or an action of a party. A task is carried out by a party assuming a role.

A role is a set of capacities assumed by an entity to initiate and participate in the execution of, or consume the outcome of, some tasks or functions in an IIoT system as required by an activity. Roles are assumed by parties. A party is an agent, human or automated, that has autonomy, interest and responsibility in the execution of tasks. A party executes a task by assuming a role that has the right capacities for the execution of the task. A party may assume more than one role, and a role may be fulfilled by more than one party. A party also has security properties for assuming a role.

FUNCTIONAL VIEWPOINT

Industrial Control Systems (ICSs) have been widely deployed to enable industrial automation across industrial sectors. As we bring these automated control systems online with broader systems in the Industrial Internet effort, control remains a central and essential concept of industrial systems. Control, in this context, is the process of automatically exercising effects on physical systems and

the environment, based on sensory inputs to achieve human and business objectives. Many control systems today apply low-latency, fine-grained controls to physical systems in close proximity, without a connection to other systems. Because of this, it is difficult to create local collaborative control, let alone globally orchestrated operations.

Some might argue that the industrial internet is the conjoining of what has been traditionally two different domains with different purposes, standards and supporting disciplines: IT and OT. In IT (information technology), everything is reducible to bits that represent ideas in the programmer's head and transformed in a way to produce useful inference—anything from the sum of numbers in a column to email systems to schedule optimization problems (e.g. using Simplex). The essential problem with such an approach, noted as one of the fundamental problems in the artificial intelligence community, is the so-called 'symbol-grounding problem'— that symbols in the machine (the numbers passed around by the processor) only correspond to world objects because of the intentions of the programmer—they have no meaning to the machine.

In OT, (operational technology) 'controls' (traditionally analogue) have been applied directly to physical processes without any attempt to create symbols or models to be processed by the machine. For example, PID (proportional-integrative-derivative) controllers may control the voltage on a line using a particular feedback equation that is defined by the control engineer and demonstrated to work for a particular application—there is no attempt at generality and no need to divide the problem among multiple processing units. The incidence of IT into the OT world has primarily come about due to a need to network larger systems and establish control over hierarchies of machines while also wanting to inject common IT ideas into the OT world (such as scheduling and optimization of resource consumption). There has also been a move toward controls that digitally simulate the physical world and base their control decisions on the simulation model rather than a control engineer's equation. This makes other kinds of approaches that have been examined in IT, such as machine learning, possible to apply to OT. This has also led to OT systems to be susceptible to IT problems as well, such as network denial of service attack and spoofing as well as the aforementioned symbol-grounding problem.

The combination of IT and OT holds forth a great possibility of advancement—embodied cognition— to a system that can avoid the symbol grounding problem by basing its representation on the world (and not on programmer supplied models) and only its own episodic experience (and thus not be limited to human conceptions of epistemology). However even nearer-term breakthroughs that will support advanced analytics based on actual world data rather than engineering models may well yield substantial improvements. The main obstacle is safety and resilience. Mission-critical OT applications are important enough that the typical levels of software reliability that are acceptable in the IT market are not sufficient for OT. Moreover, actions in the physical world generally cannot be undone, which is a consideration that IT systems normally do not have to address.

Riding on continued advancement of computation and communication technologies, the industrial

internet can dramatically transform industrial control systems in two major themes:

Increasing local collaborative autonomy:

New sensing and detection technologies provide more and more accurate data. Greater embedded computational power enables more advanced analytics of these data and better models of the state of a physical system and the environment in which it operates. The result of this combination transforms control systems from merely automatic to autonomous, allowing them to react appropriately even when the system's designers did not anticipate the current system state. Ubiquitous connectivity between peer systems enables a level of fusion and collaboration that was previously impractical.

Increasing system optimization through global orchestration:

Collecting sensor data from across the control systems and applying analytics, including models developed through machine learning, to these data, we can gain insight to a business's operations. With these insights, we can improve decision-making and optimize the system operations globally through automatic and autonomous orchestration.

These two themes have far-reaching impact on the systems that we will build, though each system will have a different focus and will balance the two themes differently.

A functional domain is a (mostly) distinct functionality in the overall IIoT system. A decomposition of a typical IIoT system into functional domains highlights the important building blocks that have wide applicability in many industrial verticals. It is a starting point for conceptualizing a concrete functional architecture. Specific system requirements will strongly influence how the functional domains are decomposed, what additional functions may be added or left out and what functions may be combined and further decomposed.

We decompose a typical IIoT system into five functional domains:

- Control domain
- Operations domain
- Information domain
- Application domain
- Business domain

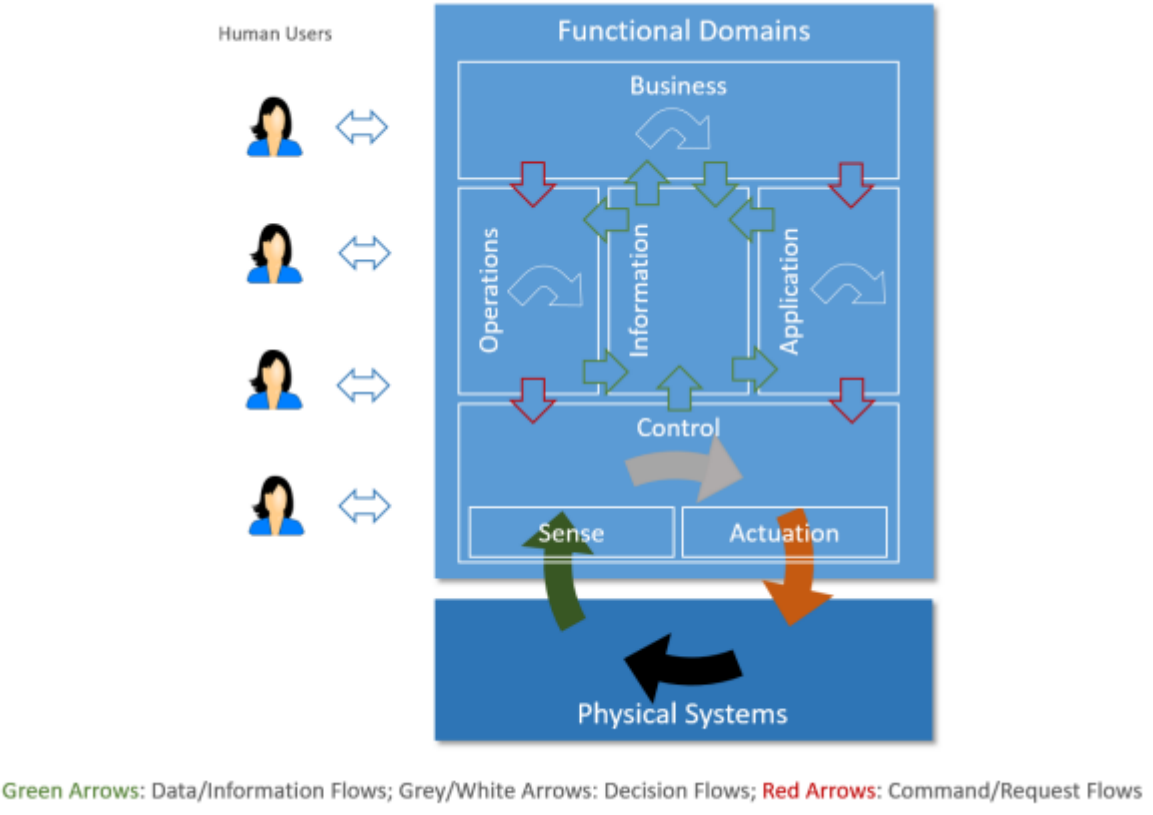


Figure 3.7: Functional Domains

Data flows and control flows take place in and between these functional domains. Figure 6-1 above illustrates how the functional domains relate to each other with regard to data and control flows. Green arrows show how data flows circulate across domains. Red arrows show how control flows circulate across domains. Other horizontal arrows illustrate some processing taking place within each domain, to process input flows and generate new forms of data or control flows.

Controls, coordination and orchestration exercised from each of the functional domains have different granularities and run on different temporal cycles. As it moves up in the functional domains, the coarseness of the interactions increases, their cycle becomes longer and the scope of impact likely becomes larger. Correspondingly, as the information moves up in the functional domains, the scope of the information becomes broader and richer, new information can be derived, and new intelligence may emerge in the larger contexts.

THE CONTROL DOMAIN

The control domain represents the collection of functions that are performed by industrial control systems. The core of these functions comprises fine-grained closed-loops, reading data from sensors ("sense" in the figure), applying rules and logic, and exercising control over the physical system through actuators ("actuation"). Both accuracy and resolution in timing is usually critical. Components or systems implementing these functions (functional components) are usually deployed in proximity to the physical systems they control, and may therefore be geographically distributed.

They may not be easily accessible physically by maintenance personnel, and physical security of these systems may require special consideration.

The control domain comprises a set of common functions, as depicted in Figure 3.8. Their implementation may be at various levels of complexity and sophistication depending on the systems, and, in a given system, some components may not exist at all. We describe each in turn. Sensing is the function that reads sensor data from sensors. Its implementation spans hardware, firmware, device drivers and software elements. Note that active sensing recursively, requires control and actuation, and may therefore have a more complex linkage to the rest of the control system, for example, an attention element to tell the sensor what is needed. Actuation is the function that writes data and control signals to an actuator to enact the actuation. Its implementation spans hardware, firmware, device drivers and software elements.

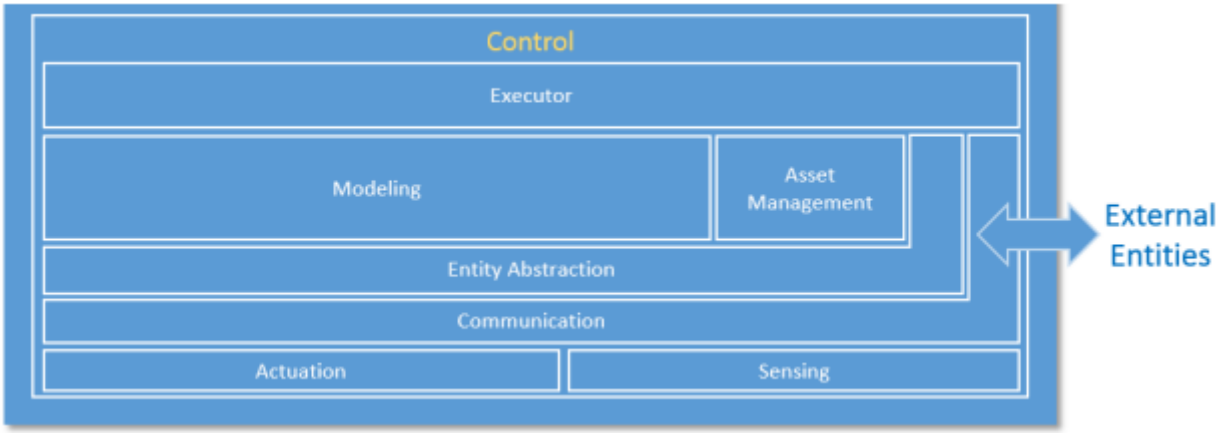


Figure 3.8: Functional Decomposition of Control Domain

Communication connects sensors, actuators, controllers, gateways and other edge systems. The communication mechanisms take different forms, such as a bus (local to an underlying system platform or remote), or networked architecture (hierarchical, hubs and spokes, meshed, pointto-point), some statically configured and others dynamically. Quality of Service (QoS) characteristics such as latency, bandwidth, jitter, reliability and resilience must be taken into account.

Within the communication function, a connectivity abstraction function may be used to encapsulate the specifics of the underlying communication technologies, using one or more common APIs to expose a set of connectivity services. These services may offer additional connectivity features that are not otherwise available directly from the underlying communication technologies, such as reliable delivery, auto-discovery and auto-reconfiguration of network topologies upon failures.

Entity abstraction, through a virtual entity representation, provides an abstraction of scores of sensors and actuators, peer controllers and systems in the next higher tiers, and expresses relationships between them. This serves as the context in which sensor data can be understood, actuation is enacted and the interaction with other entities is carried out. Generally, this includes the semantics of the terms used within the representations or messages passed between system elements.

Modeling deals with understanding the states, conditions and behaviors of the systems under control and those of peer systems by interpreting and correlating data gathered from sensors and peer systems. The complexity and sophistication of modeling of the system under control varies greatly. It may range from straightforward models (such as a simple interpretation of a time series of the temperature of a boiler), to moderately complex (a prebuilt physical model of an aircraft engine), to very complex and elastic (models built with artificial intelligence possessing learning and cognitive capabilities). These modeling capabilities, sometime referred to as edge analytics, are generally required to be evaluated locally in control systems for real-time applications. Edge analytics are also needed in use cases where it is not economical or practical to send a large amount of raw sensor data to remote systems to be analyzed even without a real time requirement.

A data abstraction sub-function of modeling may be needed for cleansing, filtering, de-duplicating, transforming, normalizing, ignoring, augmenting, mapping and possibly persisting data before the data are ready for analysis by the models or destroyed.

Asset management enables operations management of the control systems including system on boarding, configuration, policy, system, software/firmware updates and other lifecycle management operations. Note that it is subservient to the executor so as to ensure that policies (such as safety and security) are always under the responsibility and authority of the edge entity.

Executor executes control logic to the understanding of the states, conditions and behavior of the system under control and its environment in accordance with control objectives. The control objectives may be programmed or otherwise set by static configuration, be dynamic under the authority of local autonomy, or be advised dynamically by systems at higher tiers. The outcome of the control logic may be a sequence of actions to be applied to the system under control through actuation. It may also lead to interactions with peer systems or systems at higher tiers. Similar to the case of modeling, the control logic can be:

- straightforward—a set-point program employing algorithms to control the temperature of a boiler) or
- sophisticated—incorporating aspects of cognitive and learning capabilities with a high degree of autonomy, such as deciding which obstacle a vehicle should crash into—the full school bus pulling out in front of the vehicle from the grade school or the puddle of pedestrians in front of the nursing home.

The executor is responsible for assuring policies in its scope are applied so that data movement out of the scope it controls, use of actuators, etc. are within the bounds of such policies.

THE OPERATIONS DOMAIN

The operations domain represents the collection of functions responsible for the provisioning, management, monitoring and optimization of the systems in the control domain. Existing industrial control systems mostly focus on optimizing the assets in a single physical plant. The control systems of the Industrial Internet must move up a level, and optimize operations across asset types, fleets and customers. This opens up opportunities for added business and customer value as set out by higher-level, business-oriented domains.

Figure 3.9 shows how operations in an IIoT system can be supported through a suite of interdependent operations support functions.

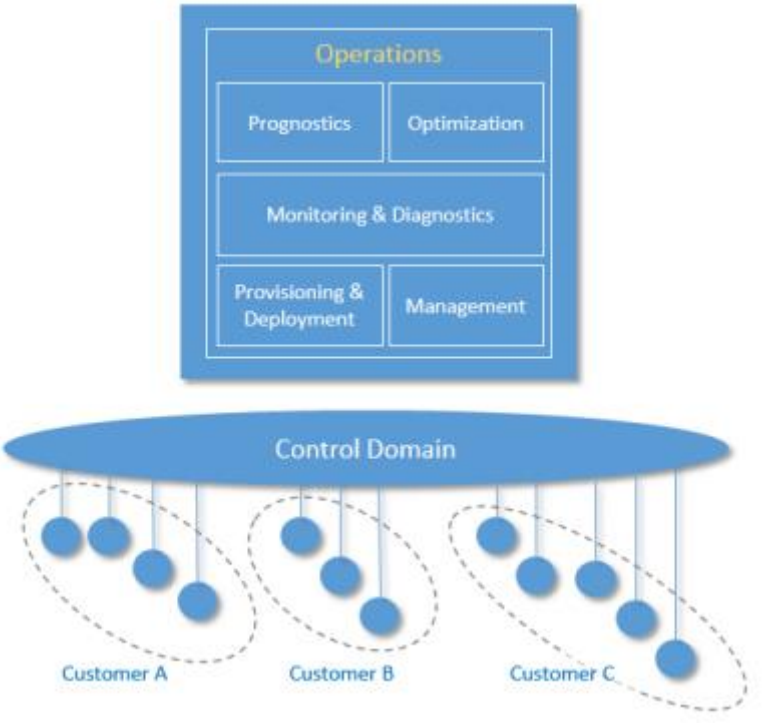


Figure 3.9: Operations Domain decomposition showing support across various customers

Provisioning and Deployment consists of a set of functions required to configure, onboard, register, and track assets, and to deploy and retire assets from operations. These functions must be able to provision and bring assets online remotely, securely and at scale. They must be able to communicate with them at the asset level as well as the fleet level, given the harsh, dynamic and remote environments common in industrial contexts.

Management consists of a set of functions that enable assets management centers to issue a suite of management commands to the control systems, and from the control systems to the assets in which the control systems are installed, and in the reverse direction enable the control systems and the assets to respond to these commands. For this, many of the legacy “dumb” assets need to be retrofitted to have compute, storage and connectivity capabilities.

Monitoring and Diagnostics consists of functions that enable the detection and prediction of

occurrences of problems. It is responsible for real-time monitoring of asset key performance indicators, collecting and processing asset health data with intelligence so that it can diagnose the real cause of a problem, and then alerting on abnormal conditions and deviations. This set of functions should assist operations and maintenance personnel to reduce the response time between detecting and addressing a problem.

Prognostics consists of the set of functions that serves as a predictive analytics engine of the IIoT systems. It relies on historical data of asset operation and performance, engineering and physics properties of assets, and modeling information. The main goal is to identify potential issues before they occur and provide recommendations on their mitigation.

Optimization consists of a set of functions that improves asset reliability and performance, reduces energy consumption, and increase availability and output in correspondence to how the assets are used. It helps to ensure assets operating at their peak efficiency by identifying production losses and inefficiencies. This process should be automated, as much as it is feasible, in order to avoid potential inaccuracies and inconsistencies.

At this level, this set of functions should support major automation and analytics features including:

- automated data collection, processing and validation,
- the ability to capture and identify major events, such as downtime, delay and
- the ability to analyze and assign causes for known problems.

Furthermore, many of the core functions in the operations domain, such as diagnostics, prognostics and optimization, may require performing advanced analytics on potentially large volume of historical asset operational and performance data. Therefore, an optimal approach is to use or share these functionalities that are available by the information domain.

THE INFORMATION DOMAIN

The information domain represents the collection of functions for gathering data from various domains, most significantly from the control domain, and transforming, persisting, and modeling or analyzing those data to acquire high-level intelligence about the overall system. The data collection and analysis functions in this domain are complementary to those implemented in the control domain. In the control domain, these functions participate directly in the immediate control of the physical systems whereas in the information domain they are for aiding decision making, optimization of system-wide operations and improving the system models over the long term. Components implementing these functions may or may not be co-located with their counterparts in the control domain. They may be deployed in building closets, in factory control rooms, in corporate datacenters, or in the cloud as a service.

Figure 3.10 illustrates the functional decomposition of the information, application and business domains.

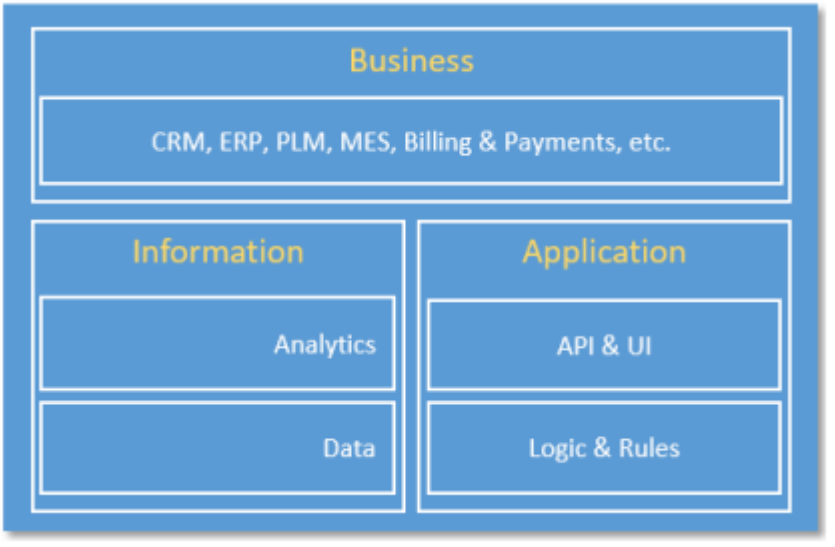


Figure 3.10: Functional Decomposition of Information, Application & Business Domains

Data consists of functions for:

- ingesting sensor and operation state data from all domains,
- quality-of-data processing (data cleansing, filtering, de-duplication, etc.),
- syntactical transformation (e.g., format and value normalization),
- semantic transformation (semantic assignment, context injection and other data
- augmentation processing based on metadata (e.g. provisioning data from the Operations Domain) and other collaborating data set, data persistence and storage (e.g. for batch analysis) and
- data distribution (e.g. for streaming analytic processing).

These functions can be used in online streaming mode in which the data are processed as they are received to enable quasi-real-time analytics in support of orchestration of the activities of the assets in the control domain. They may be used in offline batch mode (e.g. seismic sensor data collected and accumulated in an offshore oil platform that does not have high-bandwidth connectivity to the onshore datacenter).

Data governance functions may be included for data security, data access control and data rights management, as well as conventional data management functions related to data resilience (replication in storage, snapshotting and restore, backup & recovery, and so on).

Analytics encapsulates a set of functions for data modeling, analytics and other advanced data processing, such as rule engines. The analytic functions may be done in online/streaming or offline/batch modes. In the streaming mode, events and alerts may be generated and fed into functions in the application domains. In the batch mode, the outcome of analysis may be provided to the business domain for planning or persisted as information for other applications.

The data volume at the system level in most IIoT systems will eventually exceed a threshold at

which the traditional analytic toolsets and approaches may no longer scale in meeting the requirement in performance. “Big Data” storage and analytic platforms may be considered for implementing these functions.

THE APPLICATION DOMAIN

The application domain represents the collection of functions implementing application logic that realizes specific business functionalities. Functions in this domain apply application logic, rules and models at a coarse-grained, high level for optimization in a global scope. They do not maintain low-level continuing operations, as these are delegated to functions in the control domain that must maintain local rules and models in the event of connectivity loss. Requests to the control domain from the application domain are advisory so as not to violate safety, security, or other operational constraints.

The decomposition of the application domain is illustrated in Figure 3.10.

Logics and Rules comprises logics (rules, models, engines, activity flows, etc.) implementing specific functionality that is required for the use case under consideration. It is expected that there are great variations in these functions in both its contents and its constructs among the use cases.

APIs and UI represent a set of functions that an application exposes its functionalities as APIs for other applications to consume, or human user interface enabling human interactions with the application.

THE BUSINESS DOMAIN

The business domain functions enable end-to-end operations of the industrial internet of things systems by integrating them with traditional or new types of industrial internet systems specific business functions including those supporting business processes and procedural activities. Examples of these business functions include Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Product Lifecycle Management (PLM), Manufacturing Execution System (MES), Human Resource Management (HRM), asset management, service lifecycle management, billing and payment, work planning and scheduling systems.

IMPLEMENTATION VIEWPOINT

The implementation viewpoint is concerned with the technical representation of an IIoT system and the technologies and system components required implementing the activities and functions prescribed by the usage and functional viewpoints.

An IIoT system architecture and the choice of the technologies used for its implementation are also guided by the business viewpoint, including cost and go-to-market time constraints, business strategy in respect to the targeted markets, relevant regulation and compliance requirements and planned evolution of technologies. The implementation must also meet the system requirements including those identified as key system characteristics that are common across activities and must

be enforced globally as end-to-end properties of the IIoT system.

The implementation viewpoint therefore describes:

- the general architecture of an IIoT system: its structure and the distribution of components, and the topology by which they are interconnected;
- a technical description of its components, including interfaces, protocols, behaviors and other properties;
- an implementation map of the activities identified in the usage viewpoint to the functional components, and from functional components to the implementation components; and
- an implementation map for the key system characteristics.

EXAMPLE ARCHITECTURE PATTERNS

Coherent IIoT system implementations follow certain well-established architectural patterns, such as:

- Three-tier architecture pattern
- Gateway-Mediated Edge Connectivity and Management architecture pattern
- Layered Data bus pattern.

An architecture pattern is a simplified and abstracted view of a subset of an IIoT system implementation that is recurrent across many IIoT systems, yet allowing for variants. For example, an implementation of the three-tier pattern in a real IIoT system does not exclude multiple implementations of every tier—e.g. many instances of the edge tier—as well as many to-many connections between instances of a tier and instances of the next tier. Each tier and its connections will still be represented only once in the pattern definition.

We describe the three architecture patterns in more detail. The Gateway-Mediated Edge Connectivity and Layered Data bus patterns are arguably specific instances or variations of the very general Three-Tier architecture pattern.

THREE-TIER ARCHITECTURE PATTERN

The three-tier architecture pattern comprises edge, platform and enterprise tiers. These tiers play specific roles in processing the data flows and control flows (see chapter 6) involved in usage activities. They are connected by three networks, as shown in Figure 3.11.

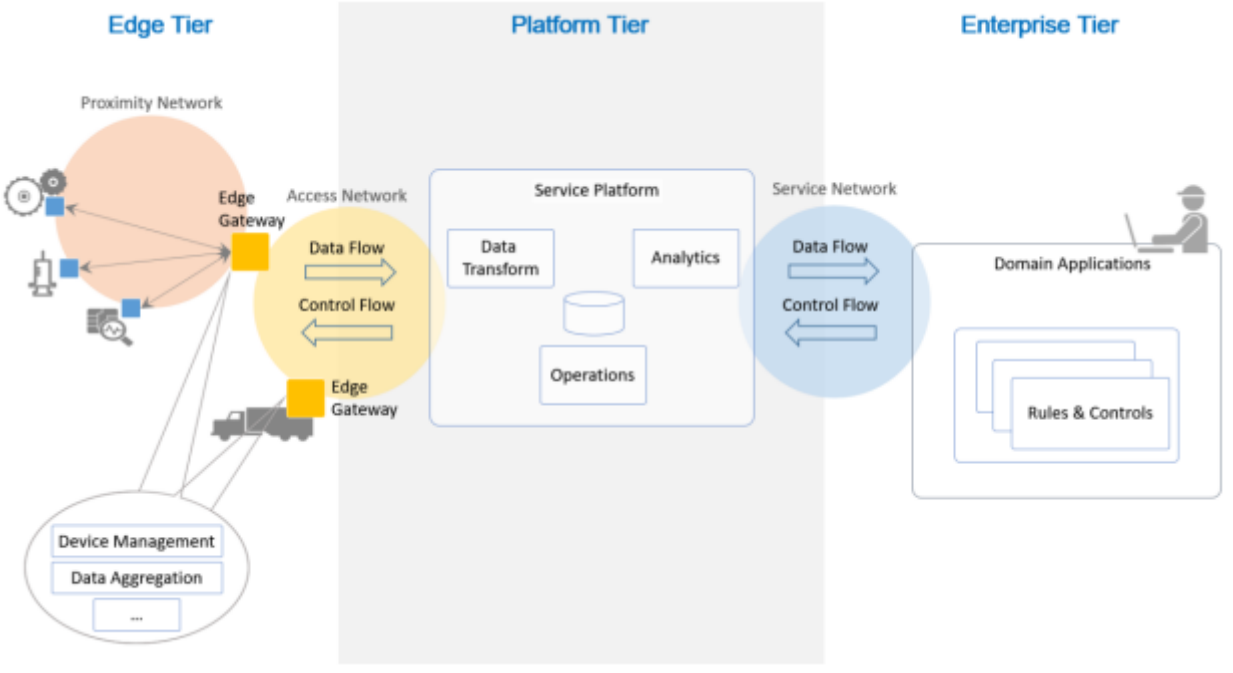


Figure 3.11: Three-Tier IIoT System Architecture

The edge tier collects data from the edge nodes, using the proximity network. The architectural characteristics of this tier, including the breadth of distribution, location, governance scope and the nature of the proximity network, vary depending on the specific use cases.

The platform tier receives, processes and forwards control commands from the enterprise tier to the edge tier. It consolidates processes and analyzes data flows from the edge tier and other tiers. It provides management functions for devices and assets. It also offers non-domain specific services such as data query and analytics.

The enterprise tier implements domain-specific applications, decision support systems and provides interfaces to end-users including operation specialists. The enterprise tier receives data flows from the edge and platform tier. It also issues control commands to the platform tier and edge tier.

The proximity network connects the sensors, actuators, devices, control systems and assets, collectively called edge nodes. It typically connects these edge nodes, as one or more clusters related to a gateway that bridges to other networks.

The access network enables connectivity for data and control flows between the edge and the platform tiers. It may be a corporate network, or an overlay private network over the public Internet or a 4G/5G network.

Service network enables connectivity between the services in the platform tier and the enterprise tier, and the services within each tier. It may be an overlay private network over the public Internet or the Internet itself, allowing the enterprise grade of security between end-users and various services.

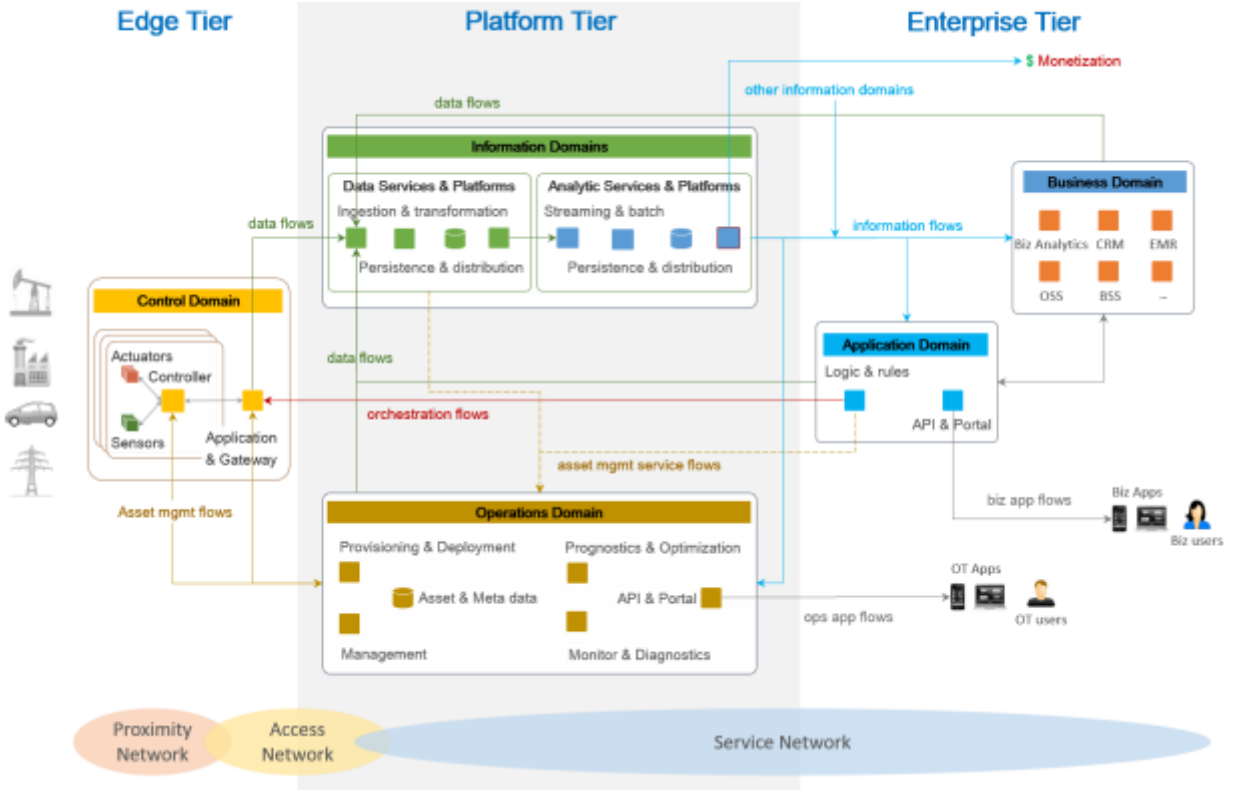


Figure 3.12: Mapping between a three-tier architecture to the functional domains

The three-tier architecture pattern combines major components (e.g. platforms, management services, applications) that generally map to the functional domains (functional viewpoint) as shown in Figure 3.12. From the tier and domain perspective, the edge tier implements most of the control domain; the platform tier most of the information and operations domains; the enterprise tier most of the application and business domains. This mapping demonstrates a simple functional partitioning across tiers. In a real system, the functional mapping of IIoT system tiers depends greatly on the specifics of the system use cases and requirements. For example, some functions of the information domain may be implemented in or close to the edge tier, along with some application logic and rules to enable intelligent edge computing.

Another reason why implementation tiers do not generally have an exclusive mapping to a particular functional domain is that these tiers often provide services to each other to complete the end-to-end activities of the system. These services, for example, data analytics from the information functional domain, then become supportive of other functional domains in other tiers.

The operations domain component of the platform tier itself provides services (asset management service flows) to other components, either in the same tier or in another.

Similar operations domain services can be provided to the application domain components in the enterprise tier as well. Conversely, the operations domain components may use data services from the information domain component in order to get better intelligence from asset data, e.g. for diagnostics, prognostics and optimization on the assets. As a result, components from all functional

domains may leverage the same data and use analytic platforms and services to transform data into information for their specific purposes.

GATEWAY-MEDIATED EDGE CONNECTIVITY AND MANAGEMENT ARCHITECTURE PATTERN

The gateway-mediated edge connectivity and management architecture pattern comprises a local connectivity solution for the edge of an IIoT system, with a gateway that bridges to a wide area network as shown in Figure 7-3. The gateway acts as an endpoint for the wide area network while isolating the local network of edge nodes. This architecture pattern allows for localizing operations and controls (edge analytics and computing). Its main benefit is in breaking down the complexity of IIoT systems, so that they may scale up both in numbers of managed assets as well as in networking. However, it may not be suited to systems where assets are mobile in a way that does not allow for stable clusters within the local network boundaries.

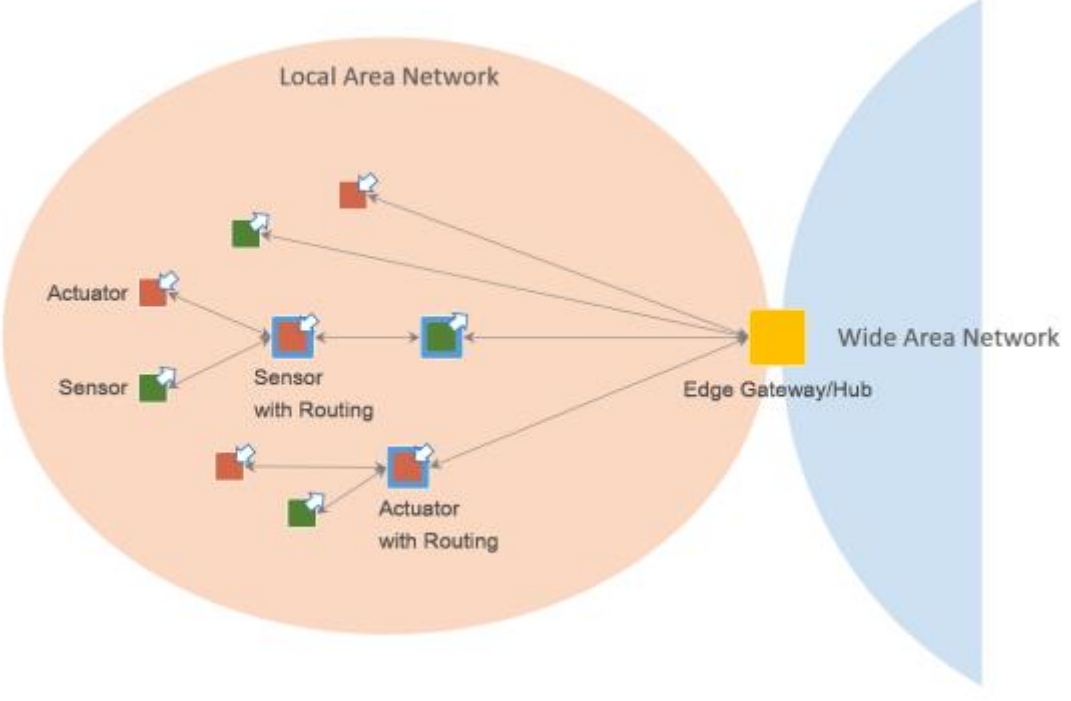


Figure 3.13: Gateway-Mediated Edge Connectivity and Management Pattern

The edge gateway may also be used as a management point for devices and assets and data aggregation point where some data processing and analytics, and control logic are locally deployed.

The local network may use different topologies. In a hub-and-spoke topology, an edge gateway acts as a hub for connecting a cluster of edge nodes to each other and to a wide area network. It has a direct connection to each edge entity in the cluster allowing in-flow data from the edge nodes, and out-flow control commands to the edge nodes.

In a mesh network (or peer-to-peer) topology, an edge gateway also acts as a hub for connecting a

cluster of edge nodes to a wide area network. In this topology, however, some of the edge nodes have routing capability. As result, the routing paths from an edge node to another and to the edge gateway vary and may change dynamically. This topology is best suited to provide broad area coverage for low-power and low-data rate applications on resource-constrained devices that are geographically distributed.

In both topologies, the edge nodes are not directly accessible from the wide area network. The edge gateway acts as the single entry point to the edge nodes and as management point providing routing and address translation.

The edge gateway supports the following capabilities:

- Local connectivity through wired serial buses and short-range wireless networks. New communication technologies and protocols are emerging in new deployments.
- Network and protocol bridging supporting various data transfer modes between the edge nodes and the wide area network: asynchronous, streaming, event-based and store-and-forward.
- Local data processing including aggregation, transformation, filtering, consolidation and analytics.
- Device and asset control and management point that manages the edge nodes locally and acts an agent enabling remote management of the edge nodes via the wide area network.
- Site-specific decision and application logic that are perform within the local scope.

LAYERED DATABUS ARCHITECTURE PATTERN

The layered data bus is a common architecture across IIoT systems in multiple industries (Figure 3.14). This architecture provides low-latency, secure, peer-to-peer data communications across logical layers of the system. It is most useful for systems that must manage direct interactions between applications in the field, such as control, local monitoring and edge analytics.

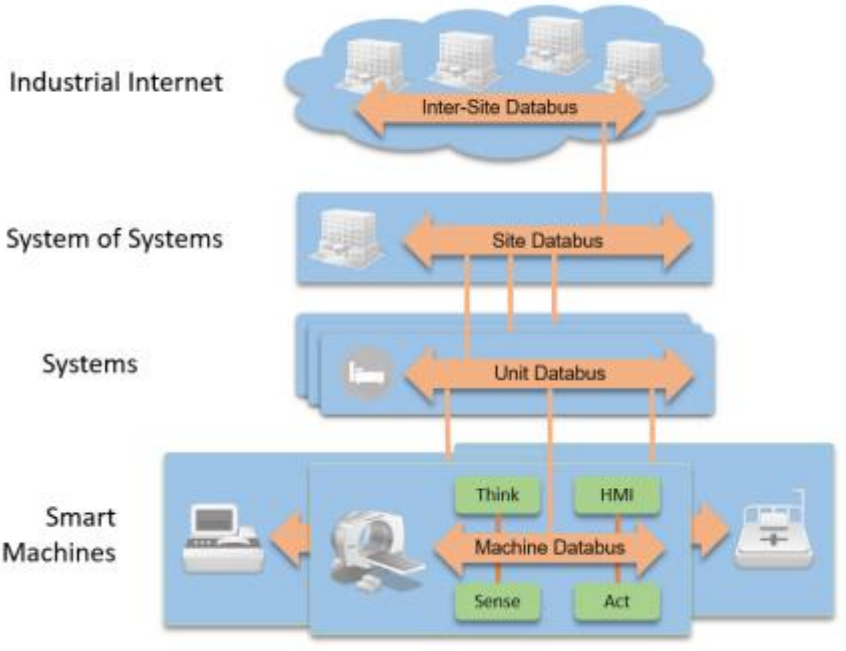


Figure 3.14: Layered Data bus Architecture

In Figure 3.14, at the lowest level, smart machines use data buses for local control, automation and real-time analytics. Higher-level systems use another data bus for supervisory control and monitoring. Federating these systems into a "system of systems" enables complex, Internet scale, potentially-cloud-based, control, monitoring and analytic applications.

A data bus is a logical connected space that implements a set of common schema and communicates using those set of schema between endpoints. Each layer of the data bus therefore implements a common data model, allowing interoperable communications between endpoints at that layer.

The data bus supports communication between applications and devices. For instance, a data bus can be deployed within a smart machine to connect its internal sensors, actuators, controls and analytics. At a higher smart system level, another data bus can be used for communications between machines. At a system of systems level, a different data bus can connect together a series of systems for coordinated control, monitoring and analysis. Each data bus may have a different set of schema or data model. Data models change between layers, as lower-level data buses export only a controlled set of internal data. Adapters may be used between layers to match data models.

The adapters may also separate and bridge security domains, or act as interface points for integrating legacy systems or different protocols.

Generally, transitions, occurring between layers, filter and reduce the data. This is important because the scope of control and analysis increases at each layer and the amount of data is generally reduced to match the broader scope, higher latencies and higher level of abstraction. An example use of this architecture for oil well monitoring and operational control, typical for large SCADA systems is represented in Figure 3.15.

In addition to its use in the control, information, application and enterprise domains, this layered

data bus architecture is useful in the operations domain for monitoring, provisioning and managing devices, applications and subsystems within the system.

Central to the data bus is a data-centric publish-subscribe communications model. Applications on the data bus simply "subscribe" to data they need and "publish" information they produce. Messages logically pass directly between the communicating nodes. The fundamental communications model implies both discovery—what data should be sent—and delivery—when and where to send it. This design mirrors time-critical information delivery systems in everyday life including television, radio, magazines and newspapers. Publish-subscribe systems are effective at distributing large quantities of time-critical information quickly, especially in the presence of unreliable delivery mechanisms.

The layered data bus architecture offers these benefits:

- fast device-to-device integration, with delivery times in milliseconds or microseconds,
- automatic data and application discovery within and between busses,
- scalable integration, comprising hundreds of thousands of sensors and actuators,
- natural redundancy, allowing extreme availability and
- hierarchical subsystem isolation, enabling development of complex system designs.

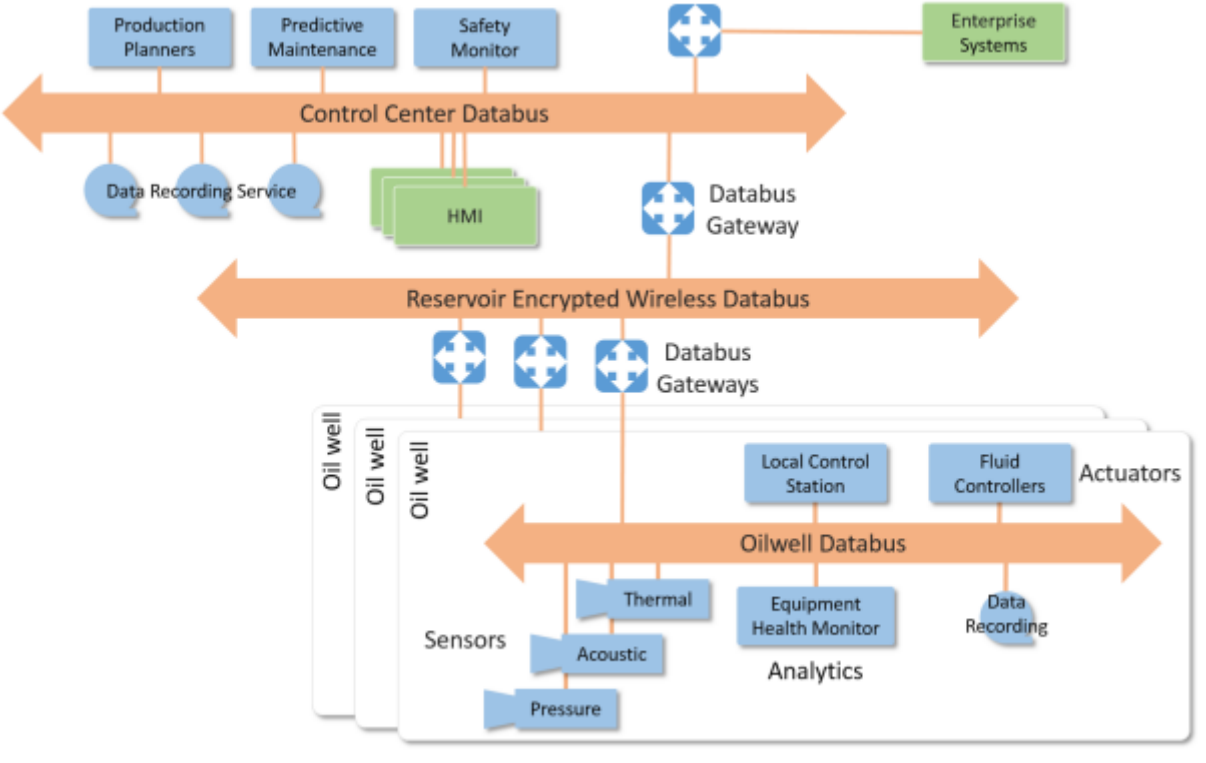


Figure 3.15: A three-layer data bus architecture.

Text/Reference Books

1. S. Misra, A. Mukherjee, and A. Roy, Introduction to IoT. Cambridge University Press, 2020
2. S. Misra, C. Roy, and A. Mukherjee, Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.2020
3. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015

Question Bank

PART-A

1. Illustrate an ISO/IEC/IEEE 42010:2011 architecture.
2. List the viewpoints in Industrial internet.
3. Describe how implementation viewpoint deals with technologies.
4. Illustrate the three-tier topology.
5. Define stakeholders.
6. Identify the three patterns where coherent IIOT system implementations follow.
7. Explain System lifecycle process.

PART-B

1. Explain and illustrate usage viewpoint in detail.
2. Describe how functional viewpoint is used in business domain, Explain with examples
3. Discuss in detail about operational and information domain.
4. Examine all the architectural patterns in detail.
5. Explain in detail about the Industrial internet architecture framework.
6. Discuss about Business and implementation viewpoints in detail.

INDUSTRIAL INTERNET OF THINGS – SECA4005

UNIT – IV INDUSTRIAL INTERNET SYSTEMS

INDUSTRIAL INTERNET SYSTEMS

Introduction-Proximity Network Protocols – WSN Edge Node – Legacy Industrial Protocols – RS232 Serial Communications, 40-20ma Current Loop, Field Bus Technologies – Modern Communication Protocols – Industrial Ethernet – Industrial Gateways.

The phrase Internet of Things is attributed to Proctor and Gamble Marketing Director Kevin Ashtong , who in 1999 was working with P&G to link their enterprise computing systems to an automated form of product data collection based around RFID (Radio Frequency Identification). Ashton noted that a major limitation of computing, at the time, was that all data needed to be collected manually. At the time, products such as RFID were used to enable remote data sensing. Auto ID Center developed a new radical approach to very low cost RFID. Rather than house product data (expensively) in the memory on the RFID tag itself, the tag memory was simply provided with identification number which when read would provide a link to further product data being (cheaply) stored on networked servers. As a result a commercial product following such a process would thus be linked to its data via an internet connection, further reinforcing the notion of an internet of things.

This notion of products being equipped with an ability to interact was formalized in 2002 with the introduction of a specification for an intelligent production.

An intelligent product is one which:

1. Possesses a unique identity
2. Can communicate effectively with its environment
3. Can retain or store data about itself
4. Deploys a language to display its features, production requirements etc.
5. Can participate in or making decisions relevant to its own destiny

Definition of Industrial IoT

Following the comments above, the definitions to be used here simply relate to the industrial use of IoT. The term industrial Internet of things (IIoT) is often encountered in the manufacturing industries, referring to the industrial subset of the IoT a Although in the context of this report we will use the rather more descriptive working definition: The application of Internet of Things developments to (create value for) industrial processes, supply chains, products and services. This is because it explicitly includes the role of products and services within its scope, as well as industrial processes and operations.

Uses of Industrial IoT

In this section, two specific benefit areas of IoT in an industrial context are raised. These relate to areas of sensing that are not traditionally part of the factory information environment and are hence not typically integrated into production or asset management considerations. In a later section, the differences between existing industrial IT systems and the options that IoT can offer will be discussed.

There is at this stage only a very limited, superficial literature on the deployment of IoT in an industrial context and even that coverage is cursory. It is extremely difficult to determine where reported applications have in fact benefited from specific IoT developments and where the reporting is simply that of a sensor application within an industrial context. Ignoring this distinction for the moment, applications that report the deployment of Industrial IoT solutions typically cover one of the following themes:

• equipment monitoring – gas turbines & construction equipment , trains , trucks

• maintenance – aircraft , wind turbines , elevators

• quality control – beverages

• energy management – manufacturing

• productivity – logistics , machine tools , oilfield production

• safety – rail transport

To summarise, the areas where Industrial IoT might provide the best immediate impact are applications in:

• Integrating data from suppliers, logistics providers, customers

• Introducing data from new technology, peripherals, tools, equipment

• Distributed production requiring addition of new data sources, locations, owners

• Sensors on board raw materials, parts, products, orders passing through organizations

MODBUS Theory

The MODBUS TCP is a byte-oriented, industrial communication protocol, open de facto standard, used for data exchange between embedded systems, devices, and industrial applications. Devices, reacting as clients, may benefit from the inexpensive implementation of such a lightweight protocol for polling industrial devices that react as servers. Polling communications follow the request–response mechanism, where a client queries the server for specific data or executes commands in the server using a frame of bytes arranged in a specific way, called a frame format. The server replies to the client queries via a frame of bytes either holding measurement data from sensors or confirming the execution of commands. Sixteen-bit data registers store measurement values, and coils hold the status of ON and OFF switches. Therefore, MODBUS TCP uses the polling mechanism, as opposed

to the event-based mechanism, explained in the next section.

As listed in Table 4.1, the protocol specifications define three categories of function codes for the access of data in remote devices. These data are stored in coils or registers as status values for measurements or transferred as set points for control. Coils perform one-bit read and write operations for switching the attached devices ON and OFF or reading and writing one-bit internal configuration values. Discrete inputs perform one-bit read operations for reading the status of the attached devices, whether they are switched ON or OFF. The 16-bit input registers are responsible for measurements from physical devices, and the 16-bit holding registers perform read and write operations related to internal reconfigurable values.

Table 4.1 : Function codes for data access in MODBUS

Data Access	Type	Function Code	Meaning
1 bit	physical discrete input	0x02	read discrete inputs
1 bit	internal bits, physical coils	0x01	read coils
1 bit	internal bits, physical coils	0x05	write single coil
1 bit	internal bits, physical coils	0x0F	write multiple coils
16 bit	physical input registers	0x04	read input registers
16 bit	internal and physical output registers	0x03	read holding registers
16 bit	internal and physical output registers	0x06	write single register
16 bit	internal and physical output registers	0x10	write multiple registers
16 bit	internal and physical output registers	0x17	read/write registers
16 bit	internal and physical output registers	0x16	mask write register
16 bit	internal and physical output registers	0x18	read first in first out (FIFO) queue

A message structure of a MODBUS TCP client query for reading input registers is shown in Figure 4.1. The slave replies to the master query in the same format with the read registers using the function code (FC) “read input registers” (FC = 0x04), or as a confirmation to executing commands in case of other function codes such as “write single coil” (FC = 0x05). The header of the MODBUS frame consists of four fields: a two-byte transaction identifier (ID); a two-byte protocol type (MODBUS over TCP); a two-byte length, which counts the number of bytes for the rest fields; and a one-byte unit identifier (Unit). However, the protocol data unit (PDU) consists of a one-byte function code (FC), which is, here, a code to read the registers and a data field that may contain other fields depending on the FC itself. Both the header and the PDU form an application data unit (ADU), which is the complete frame of the query.

The following illustrative example explains the principle of the MODBUS frame format that uses a function code (0x04) to read three continuous input registers in a remote device. The function is able to read from 1 to 125 contiguous input registers. Here, a client query asks a server to read the values of three continuous input registers—register address “14” (0x000E), register address “15” (0x000F), and register address “16” (0x0010). Therefore, the client sends a single message “000100000060104000E0003” and the server replies by sending one frame “000100000009010406FE206666A63F” that contains three values of continuous registers. The first register contains the hexadecimal value “0xFE20,” which corresponds to the sixteen-bit signed short integer value “11111110 00100000” or the decimal value “–480”. The last two registers hold the

IEEE 754 short floating-point representation “0x3FA66666

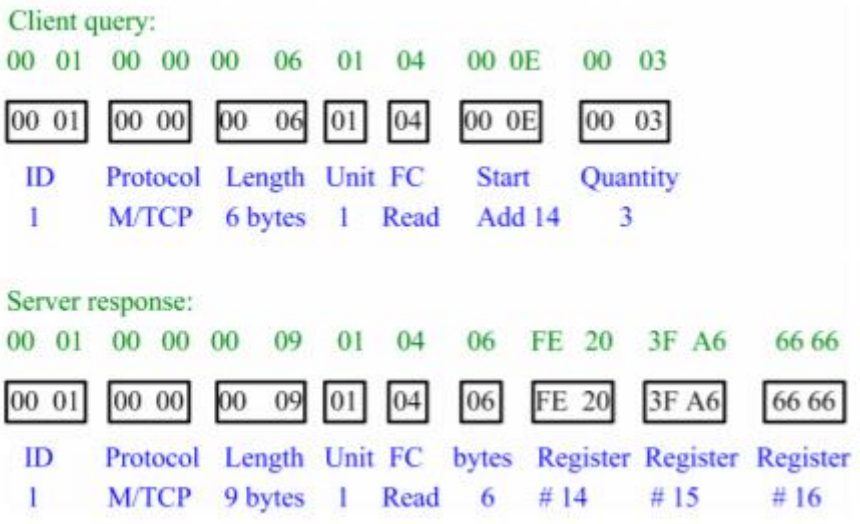


Fig 4.1 : MODBUS query and response, an illustrative example

Proximity network

A proximity network is one where the physical proximity of a user to a network makes immediately available a local service environment. Proximity networks are adjacent to the primary market sites, but less expensive than being in the market sites.

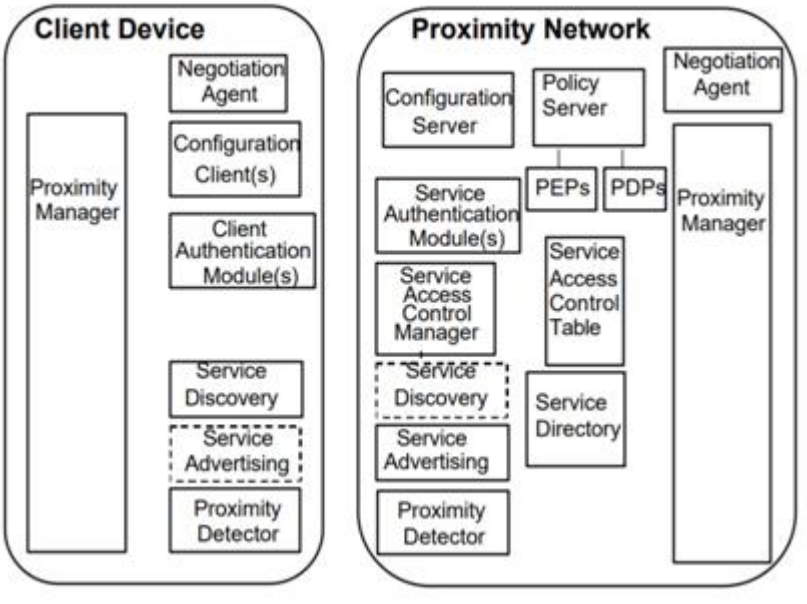


Fig 4.2 : Components of Proximity Network Architecture

The industrial Internet of Things (IoT) encompasses the broad transformation underway that will make pervasive sensing across connected machines not just a competitive advantage, but an essential fundamental service. The industrial IoT starts with the edge node, which is the sensing and measurement entry point of interest. This is where the physical world interacts with computational

data analytics. Connected industrial machines can sense a wide array of information that will be used to make key decisions. This edge sensor is likely far removed from the cloud server that stores historical analysis. It must connect through a gateway that aggregates edge data into the internet. Ideally, the edge sensor node is unobtrusive within a small nominal form factor to easily deploy in space constrained environments. Sense, Measure, Interpret, Connect, In this first of a multipart industrial IoT series, we will break down and explore the fundamental aspects of the edge node sense and measurement capabilities within the larger IoT framework: sensing, measuring, interpreting, and connecting data, with additional consideration for power management and security. Each portion presents a unique set of challenges. Smart partitioning of the edge node can be key to a successful implementation.

In some cases, ultralow power (ULP) is the most important performance metric. The vast majority of potential data may be filtered when the sensor wakes from sleep mode during key events. Sensors form the front-end edge of the industrial IoT electronics ecosystem. Measurements transform the sensed information into something meaningful such as a quantifiable value of pressure, displacement, or rotation. The interpretation stage is where edge analytics and processing transforms the measured data into an actionable event.1 Only the most valuable information should be connected beyond the node into the cloud for predictive or historical processing. All along the signal chain, the data can be rejected or filtered based on initial limits of acceptability.

Ideally, the sensor node should only send information that is absolutely necessary and should make critical decisions as soon as key data is available. The edge node must be connected to the outside network, either through a wired or wireless sensor node (WSN). Data integrity remains key in this block of the signal chain. Optimum sensed and measured data is of little value if the communication is inconsistent, lost, or corrupted. Missing data via communication cannot be an option.

Electrically noisy industrial environments can be harsh and unforgiving, especially for radio frequency communication in the presence of high metal content. Therefore, a robust communication protocol must be designed as a forethought during system architecture design. Power management for ULP systems starts with regulator component selection for maximum efficiency. But, as edge nodes may also wake and sleep with a rapid duty cycle, the power-up and power-down time should also not be ignored. An external trigger or wake-up command aids in the ability to quickly alert the edge node to begin sensing and measuring data.

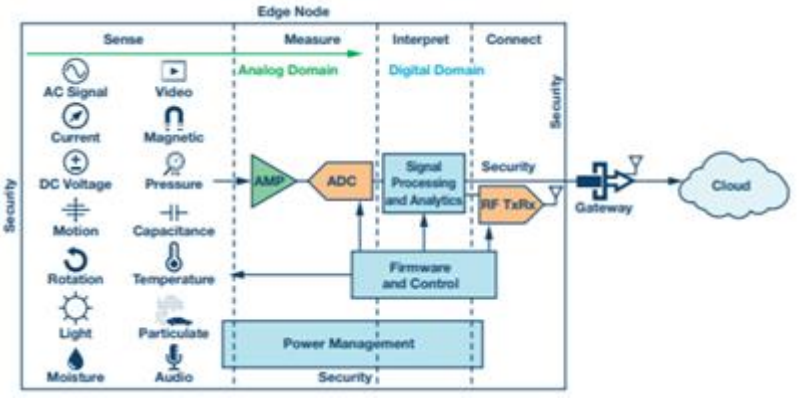


Fig 4.3 : An edge node device provides the intelligence to sense, measure, interpret, and connect to an internet gateway to the cloud

Intelligence Starts at the Edge There are a legion of sensing solutions at the edge, which may not just be a single discrete device. The edge may be a plurality of various concurrent unrelated data acquisitions. Temperature, sound, vibration, pressure, humidity, motion, pollutants, audio, and video are just some of the variables that can be sensed, processed, and sent to the cloud through a gateway for further historic and predictive analysis. It is not a hyperbole to say that sensors are the backbone of industrial IoT.2 But it might be more accurate to say they're the central nervous system for extracting insights. The edge node sense and measurement technology is the birthplace for the data of interest. If bad or incorrect data is faithfully recorded at this stage in the solution chain, no amount of post processing in the cloud can reclaim the lost value. Mission critical systems, such as healthcare and factory line-down monitoring with high stakes outcomes, require robust integrity of quality data measurements. Data quality is paramount. False positives or omissions can be costly, time consuming, and potentially life threatening. Costly errors eventually result in unplanned maintenance, inefficient labor use, or having to disable the IoT system entirely. Intelligence starts at the edge node where avoidance of the old adage still applies— garbage in, garbage out.

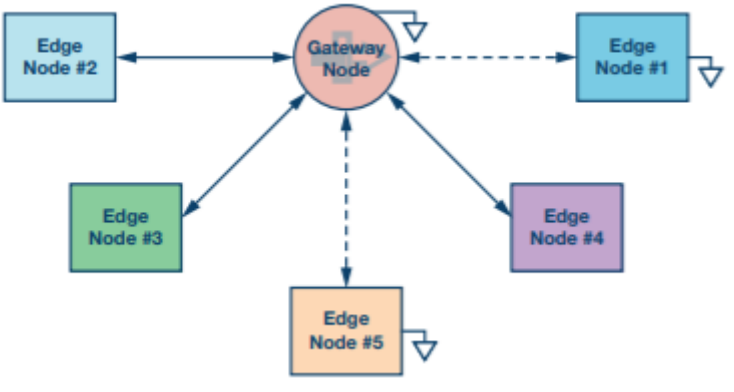


Fig 4.4: Many edge node outputs, both wired and wireless can autonomously connect to gateway node to be aggregated before transmission to a cloud server

Challenges in sensor networks

- Energy constraint : Nodes are battery powered
- Unreliable communication : Radio broadcast, limited bandwidth, bursty traffic
- Unreliable sensors : False positives
- Ad hoc deployment : Pre-configuration inapplicable
- Large scale networks : Algorithms should scale well
- Limited computation power : Centralized algorithms inapplicable
- Distributed execution : Difficult to debug & get it right

Serial communication – RS232 A popular way to transfer commands and data between a personal computer and a microcontroller is the use of standard interface, like the one described by protocols RS232 (older) or USB (newer). This chapter is devoted to communication conforming to RS232 protocol, the hardware for such interface is provided on board. An example will be presented showing the processing of commands received through RS232 interface, and sending of a string of numbers using the same interface. The protocol RS232 defines the signals used in communication, and the hardware to transfer signals between devices. The time diagram of the typical signal used to transfer character 'A' (ASCII: 6510 or 0x41) from device A to device B is given in Figure, and would appear on the upper line TX -> RX between devices.

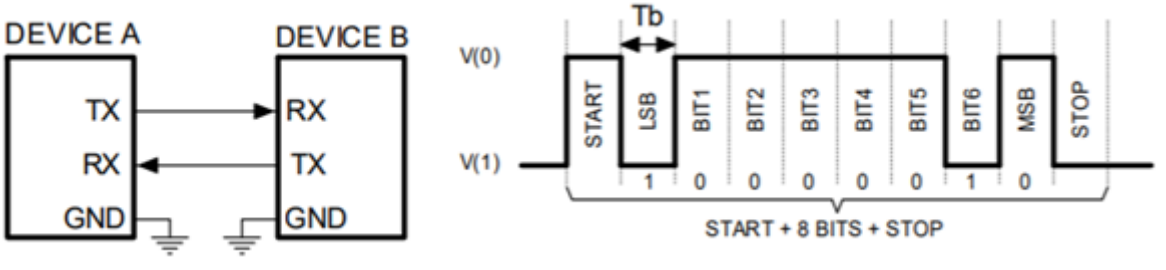


Fig 4.5 :A signal conforming to RS232 standard

The standard defines voltage levels V(0) to be at least +5V at the transmitting end of the line TX, and can be degraded along the line to become at least +3V at the receiving and of the line. Similarly voltage level V(1) must be at least -5V at TX, and at least -3V at RX. The standard also defined the upper limit for these voltages to be up to ±15V. Logic high is transferred as V(0). The microcontroller cannot handle such voltage levels, so typically a voltage level translator is inserted

between the microcontroller and the connector where the RS232 signals are available. The connectors are typically so-called D9 connectors, and the electric wiring in between two connectors at devices A and B is shown in Figure, for two female type connectors at both devices. The standard defines the number of bits to be transferred within one pack, Figure right, as eight for regular transmission, and nine for special purposes. The duration Tb of each bit defines the speed of transmission and is called the baud-rate. The typical baud-rate is 9600 bits per second (Baud, Bd), and the time Tb equals 104.16μs. Other baud rates are also common: 19200 Bd, 38400 Bd, 57600 Bd and 115200 Bd. Other baud-rates are standardized, but used less frequently. The beginning of the pack of bits is signaled by a so called "START bit", which has value 0 by definition. Its duration is equal to Tb. The pack of bits is terminated by so called "STOP bit" with a typical duration of Tb, bat can also last either 0.5, 1.5 or 2 Tb, depending on the application. The complete transmission of a byte at typical baud rate of 9600 Bd therefore takes 1.0416 ms.

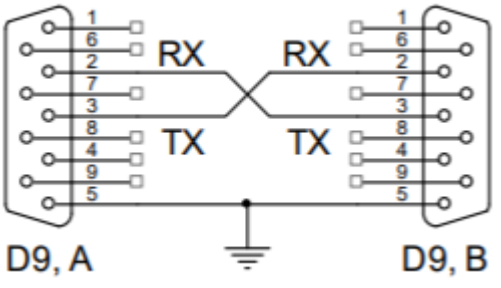


Fig 4.6: Wiring for RS232 Communication

To verify the validity of the transmission the protocol RS232 provides a so called "parity bit". A single bit is added by the transmitter before the stop bit, and its value is selected to give either odd or even number of ones in a string. The number of ones in a string can be verified at the receiving end, and if it does not match the required value, the transmission should be repeated. Other, higher level protocols to ensure the valid transmission, can be implemented in software. The protocol RS232 also accounts for testing if the receiver is capable of receiving incoming bytes and defines two additional wires called RTS (Request To Send) and CTS (Clear To Send) between devices. We will not use either of these options in our experiments. The microcontroller includes up-to six hardware modules to deal with RS232 signals. Some of the modules additionally implement other communication protocols, like I2C, CAN, SPI; module named UART4 will be used in this experiment. Its detailed description can be found in RM0090, chapter 26. The voltage level translator is added on the test board, and is industry standard chip MAX3232. The signals TX and RX are available at connector P580, pins 3 and 2 respectively. The RS232 signals RX and TX are available as alternate functions replacing the regular port bits, and corresponding port pins must be properly initialized in software. The software written demonstrates the use of serial communication. The program starts with the initialization, and proceeds into the endless loop, where it periodically

checks the status of pushbutton S370. If pressed, the LED D390 is turned on to verify the execution of the regular program. All communication tasks are processed within an interrupt function. The interrupt function is used for a good reason: the RS232 communication is rather slow, and it is a complete waste of processor to let it wait for signals from RS232. Let it rather do more important work, and jump to interrupt function dealing with RS232 only when necessary. The initialization starts with defining enabling clock for ports A and E, and defining the port pin types. Pin PA06 is used to drive an LED, and pin PE00 is used to check the pushbutton. The initialization continues with enabling the clock for UART4; the corresponding bit is located in register RCC_AHB1ENR (UART4EN, bit 19, RM0090, page 148). Since the TX and RX signals are mapped as alternate functions to regular port pins, we can find their location and corresponding alternate function in Chip description (DM00037051, page 59). The signals UART4_TX and UART4_RX are available as alternate function AF8 at port C, pins PC10 and PC11 respectively. In order to use port C the clock for this port must first be enabled, and then corresponding pins can be put into alternate function mode by setting bits 21 and 23 in the mode register for port C (GPIOC_MODER). Finally, the correct alternate mode is selected b writing 8 (01000b) into register AFR at positions reserved for pins 10 and 11. This ends the initialization for routing the signals RS232. The module UART4 needs initialization as well. Baud rate must be selected first, and this is done by writing a constant into register BRR (Baud Rate Register, UART4_BRR). The derivation of the constant is described in RM0090, and several constants for different baud rates are offered as a comment. Next the UART4 module needs to be enabled; this is done by setting bits in control register (UART4_CR1). There is one bit (bit 13, UE) to turn on the complete UART4, and then two additional bits to turn on the transmitting part (bit 3, TE) and receiving part (bit 2, RE). To set all three in one step a 0x200c is OR-ed to the content of register UART4_CR1. Finally, UART4 should be allowed to issue interrupt requests whenever a new byte is received and is available to be read by the software in the microcontroller. This gets allowed by setting a bit (bit 5, RXNEIE, Receiver Not Empty Interrupt Enable) in the same register. Since the module UART is to issue interrupt requests, the processor should be allowed to respond, and this is done by enabling the interrupt controller NVIC by a call to a function NVIC_EnableIRQ with the argument “UART4_IRQn”. After this the execution is passed to the endless loop to check the status of the pushbutton and control the LED. The initialization part of the program is given in Figure.

```
#include "stm32f4xx.h"

char *OutString;        // string must be terminated by '\0'

void main(void) {

  // GPIO clock enable, digital pin definitions
  RCC->AHB1ENR  |= 0x00000001;  // Enable clock for GPIOA
  RCC->AHB1ENR  |= 0x00000010;  // Enable clock for GPIOE
  GPIOA->MODER  |= 0x00001000;  // output pin PA06: LED D390

  //UART4 initialization
  RCC->APB1ENR  |= 0x00080000;  // Enable clock for UART4
  RCC->AHB1ENR  |= 0x00000004;  // Enable clock for GPIOC
  GPIOC->MODER  |= 0x00a00000;  // PC10, PC11 => AF mode
  GPIOC->AFR[1] |= 0x00008800;  // select AF8 (UART4,5,6) for PA10, PA11
  UART4->BRR     = 0x1120;      // 9600 baud
  //UART4->BRR     = 0x0890;      // 19200 baud
  //UART4->BRR     = 0x0450;      // 38400 baud
  //UART4->BRR     = 0x02e0;      // 57600 baud
  //UART4->BRR     = 0x016d;      // 115200 baud
  UART4->CR1    |= 0x200c;      // Enable UART for TX, RX
  UART4->CR1    |= 0x0020;      // Enable RX interrupt

  //NVIC init
  NVIC_EnableIRQ(UART4_IRQn);   // Enable IRQ for UART4 in NVIC

  // endless loop
  while (1) {
    if (GPIOE->IDR & 0x0001) GPIOA->ODR |=  0x0040;   // LED on
    else                     GPIOA->ODR &= ~0x0040;   // else LED off
  };
}
```

A response to the interrupt request requires an interrupt function. An example of such function to handle interrupts from UART4 is presented in Fig. 4, and should simply be appended to the listing in Figure before the compilation. The interrupt function is named as required in the interrupt vector table (UART4_IRQHandler), and neither receives not returns any variables. Any variables used and expected to last longer than the execution of the interrupt function must be declared as global. We are going to send a string of bytes to demonstrate the capabilities of the microcontroller, and such string, actually a character pointer to a string, was already declared in listing, Fig. 3, 3rd line. Such string is expected to be terminated by a '\0', a standard C-language termination character.

There are many possible reasons to interrupt the execution of the regular program. When a new byte is received by the UART4, it appears in the UART4_DR (Data Register), and should be removed from there and handled before it gets overwritten by a next byte received. This is so-called receive interrupt, and was already explained and enabled in listing from Figure. There are even other reasons for UART4 to request attention and issue an interrupt request; those reasons will be dealt with later. However, there is only one interrupt request available for UART4, and the reason an interrupt function was evoked is stored in status register of the UART4 (UART4_SR). Bit RXNE (bit 5, Receiver data register Not Empty) is set when the receipt of a byte causes the interrupt request, and this can be checked within the interrupt function. The body of the interrupt function therefore starts by testing the bit 5; if it is set then the reason for the interrupt is a byte waiting in the UART4 data register, and this byte must be read from there. This gets done in the 6th line of the

listing in Figure. This read from the data register simultaneously clears the bit RXNE, and the UART is ready to receive next byte. Once the received byte is safe the software compares it with some predefined values. If the value is equal to ASCII ‘a’ then an LED connected to port A, bit 6 is turned on. If its value is equal to ASCII ‘b’ then the same LED is turned off. If the value of the byte received is between ASCII ‘A’ and ASCII ‘Z’ then the byte is echoed back to the sender by a simple write back into the data register. Many times a string of bytes is to be transferred. This possibility is demonstrated next. Since many bytes are to be transferred, this is expected to take some time. Unfortunately, the data register must be written byte by byte only after the previous byte is successfully sent over the UART. The processor time would be wasted if the processor is to wait for UART to transmit a byte, so the interrupt function should be used for transmission as well. The procedure is as follows. We prepare the string of bytes to be transferred. We should terminate the string with a unique byte as in this example (the C-language automatically terminates a string by ‘\0’ character) or know the length of the string. Then another interrupt request should be enabled, this shall be issued when a byte is fully transmitted and the next one is allowed to be written into the data register; this is done by setting a bit TXEIE in the control register (Transmit data register Empty Interrupt Enable, bit 7, UART4_CR1). The last thing is to write the first byte of the string into the data register sending it through UART4, increment the pointer to the string of bytes and exit the interrupt function. These three steps are implemented as the fourth option when a byte ASCII ‘c’ is received in listing from Figure, RX IRQ part. Now when the first byte of the string is transferred over the UART4, an interrupt request calls the interrupt function again, but this time bit TXE (Transmit data register Empty) is set signalling next byte can be written into the data register, and the bit RXNE is cleared. Another IF statement is implemented in listing from Fig. 4 to check the status of bit TXE. When this bit is set, the current element of the string checked for the termination byte (‘\0’). If it differs from the termination byte then it is written into the data register, therefore sent through UART4, and the pointer to the string of bytes is incremented. The interrupt request for transmission is left enabled, so the sending of characters can continue. If the current element is equal to the termination character then the interrupts requests for transmission are disabled by clearing the bit TXEIE in control register UART4_CR1. The program can be checked by the use of HyperTerminal (Windows, pre-7 edition), or one of the freely available terminal emulators.

Some caution should be exercised when such program is debugged. Placing a breakpoint into the interrupt function stops the execution of the program and simultaneously causes a read from registers within the microcontroller by the debugging software. The read includes the data registers of the UART. Some flags may be affected by this read, for instance the flag on TXE and RXNE, causing wrong execution of the interrupt function in step-by-step mode of the debugger! Such traps are difficult to identify and the author of this text spent one afternoon in looking for an error in this program, but there was no error at all! Errors were introduced by misplaced breakpoints, and the program worked fine after the breakpoints were moved to a better place.

```
// IRQ function
void UART4_IRQHandler(void)     {

  // RX IRQ part
  if (UART4->SR & 0x0020) {     // if RXNE flag in SR is on then
    int RXch = UART4->DR;       // save received character & clear flag
    if (RXch == 'a') GPIOA->ODR |=  0x0080;   // if 'a' => LED on
    if (RXch == 'b') GPIOA->ODR &= ~0x0080;   // if 'b' => LED off
    if (RXch >= 'A' && RXch <= 'Z') UART4->DR = RXch;   // echo character
    if (RXch == 'c') {                        // if 'c' => return string
      OutString = "ABCDEFGH";   // Init string & ptr
      UART4->CR1 |= 0x0080;     // Enable TX IRQ
      UART4->DR = *OutString++; // Send first character and increment the pointer
    };
  };

  // TX IRQ part
  if (UART4->SR & 0x0080) {     // If TXE flag in SR is on then
    if (*OutString != '\0')     // if not end of string
      UART4->DR = *OutString++; // send next character and increment pointer
    else                        // else
      UART4->CR1 &= ~0x0080;    // disable TX interrupt
  };

}
```

In order to understand what a 4-20 mA direct current (DC) loop is and how it works, we will need to know a little bit of math. Don't worry; we won't be delving into any advanced electrical engineering formulas. In fact, the formula we need is relatively simple: V = I x R. This is Ohm's Law. What this is saying is that the voltage (V) is equal to the current (I) multiplied by the resistance (R) ("I" stands for Intensité de Courant, French for Current Intensity). This is the fundamental equation in electrical engineering.

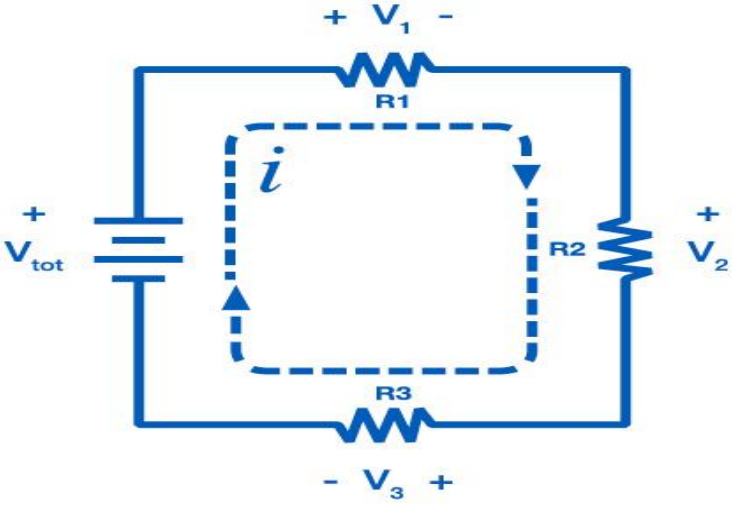


Fig 4.7 : Simple DC Circuit

Consider the simple DC circuit above, consisting of a power supply and three loads. A current loop requires voltage to drive the current. This is provided by the power supply, with the voltage of the supply labeled as Vtot. Current then flows through the loop, passing through each load. The voltage drop at each load can be calculated from Ohm's Law. The voltage drop V1 across R1 is:

Fig 4.8 : Ohm's Law

Every element in the loop either provides voltage or has a voltage drop. However, the current, "I" is the same everywhere in the loop. This is the critical principle of the 4-20 mA loop. Current is the same in all places throughout the loop. It may be difficult to understand why the current remains constant, so consider your home's water system as a comparison. There is a certain amount of pressure in the water pipes pushing the water towards your house.

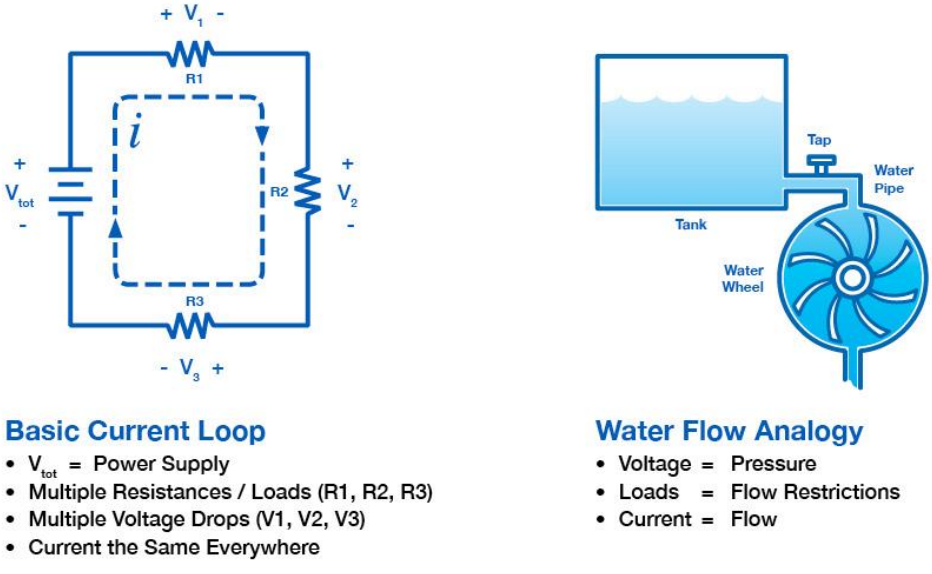
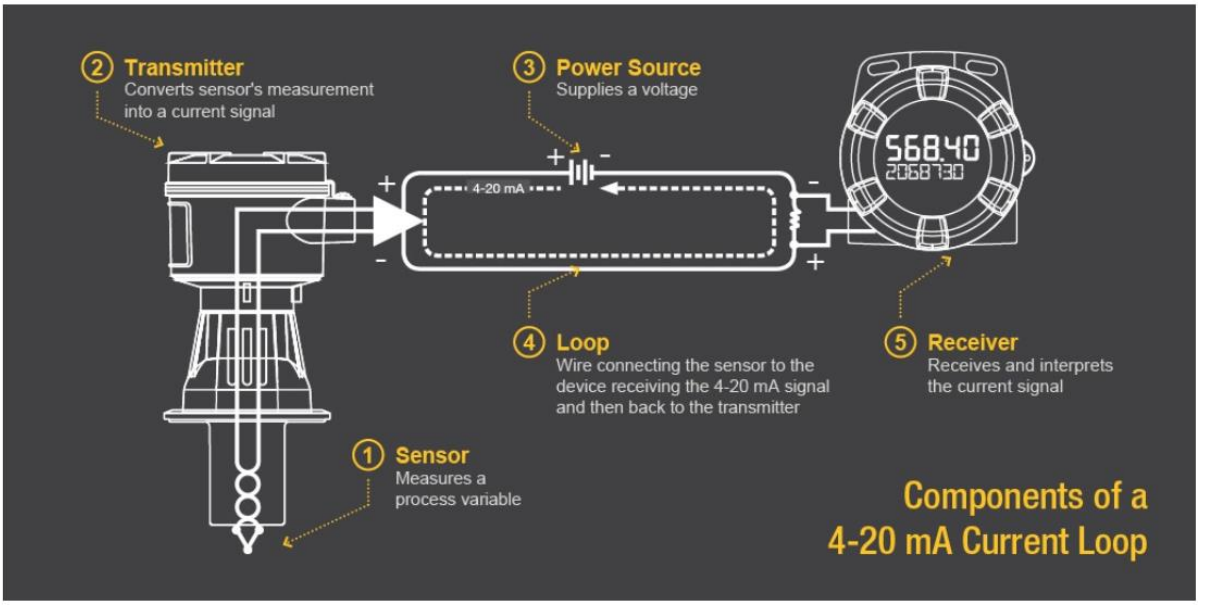


Fig 4.9 : Current / Water Flow Analogy

Voltage, in a similar fashion, acts as a pressure, pushing current through the circuit. When a tap inside your home is turned on, there is a subsequent flow of water. The flow of water is analogous to the flow of electrons, or current. The ability of the pressure to push the water through the pipes is limited by bends and restrictions in the pipe. These restrictions limit the amount of flow in the pipe, similar to how a resistor limits the current. The flow through the pipe, and likewise the current through the wire, remains constant throughout the system, even though pressure, and likewise voltage, will drop at various points. This is why using current as a means of conveying process information is so reliable.



Legacy Industrial Communication Protocols First generation industrial communications networks were built on a variety of serial-based interfaces originally created by different companies that later became de-facto standards. The result being many different standards in the marketplace. Large companies were behind these de-facto standards pushing them in the marketplace. Industrial automation equipment companies were then compelled to implement many of these protocols. Many serial-based protocols, including PROFIBUS, CAN bus, Modbus, CC-Link, and standard RS-422/RS-485 remain popular today due to the long life cycles of industrial systems. These interfaces, many of which are still in use today, were slow and limited by the number of addresses (or number of slave nodes supported) on any one network segment. Throughput rates ranged upward to 12 megabits per second (Mbps) and a network could typically support no more than several hundred addresses for end nodes. These networks made e elective use of the technology of the day and have stood the test of time. While the number of slave nodes supported is still considered sufficient, when combined with slow bus speeds, the time to exchange data between master and slave increases. This is especially critical as end nodes get more sophisticated with higher sensor integration, added intelligence, greater spatial separation, and eventually more data.

PROFIBUS

PROFIBUS is widely deployed in industrial automation systems including those for factory and process automation. PROFIBUS provides digital communication for process data and auxiliary data with speeds up to 12 Mbps supporting up to 126 addresses. CAN Control Area Network (CAN) bus, is a high-integrity serial bus system. It was originally created as an automotive vehicle bus and later came to be used as one of the field buses for industrial automation. It provides a physical and data

link layer for serial communication with speeds up to 1 Mbps. CAN open and Device Net are higher level protocols standardized on top of CAN bus to allow interoperability with devices on the same industrial network. CAN open supports 127 nodes on the network while Device Net supports 64 nodes on the same network. Modbus Modbus is a simple, robust and openly published, royalty free serial bus that connects up to 247 nodes together in the link. Modbus is easy to implement and runs on RS-232 or RS-485 with physical links speeds up to 115K baud. CC-Link CC-Link was originally developed by the Mitsubishi Electric Corporation in 1997 and is a popular open-architecture, industrial network protocol in Japan and Asia. CC-Link is based on RS-485 and can connect with up to 64 nodes on the same network with speeds up to 10 Mbps. Descriptions of Industrial Ethernet Communication Protocols Ethernet is cost effective and ubiquitous, offering common physical links with increased speed. As a result, many industrial communication systems are moving to Ethernet based solutions as illustrated in Figure 3. As the applications running on industrial networks became more sophisticated, they required greater and faster data transfer speeds. Ethernet based networking technology has provided a solution, as it is very cost effective, widely deployed, and understood. It is capable of much higher bandwidth performance than the serial fieldbus interfaces and scalable practically to an unlimited number of nodes. These capabilities translate into improved manufacturing yields, the ability to deploy new, more demanding applications and improved TCO. However, before industrial Ethernet could be widely deployed, its primary constraint had to be overcome. Since the original Ethernet protocol relied on Collision Sense Multiple Access with Collision Detection (CSMA/CD) mechanisms, it could not provide the determinism or predictability required by real-time communications applications such as industrial network-ing. Modifications could fortunately be made to Ethernet's Media Access Control (MAC) layer to facilitate real-time predictability and low latency response times. Now, several of the more popular real-time industrial Ethernet protocols will support networking speeds in the 100 megabits per second (Mbps) and the gigabit persecond (Gbps) range going forward. The most prominent real-time Ethernet protocols today include EtherCAT, EtherNet/IP, PROFINET, POWERLINK, Sercos III, Modbus TCP, and CC-Link IE. Ethernet also enables flexible network topologies that scale with the number of network nodes. Ethernet communications using standard TCP/UDP/IP are nondeterministic, with typical cycle times greater than 1000 ms, and packets subject to collision on the wire. Some Industrial Ethernet protocols use a modified Media Access Control (MAC) layer to achieve low latency and deterministic responses. Table 1 illustrates the umbrella organizations responsible for each Industrial Ethernet type discussed here. Figure 4.10 on the following page illustrates how the Industrial Ethernet protocols relate to standard TCP/UDP/IP based Ethernet, together with their MAC layer modifications. Ethernet/IP, Modbus, and standard PROFINET protocols are completely TCP/UDP/IP based. Standard Ethernet switches and controllers can be used in the network (Figure 5a). For PROFINET RT and POWERLINK, process data is carried over TCP/UDP/IP with timing controlled by a process data driver. Standard Ethernet switches, hubs, and controllers can be used in the network . For PROFINET IRT, CC-Link IE, SERCOS, and EtherCAT, process data is carried over TCP/UDP/IP with timing controlled by a process data driver. Special Ethernet link layer/MAC

hardware is required for protocol slave devices to realize determinism (Figure 5c). Nearly all Industrial Ethernet protocols, if not entirely TCP/UDP/IP based, provide some mechanism to encapsulate and carry standard Ethernet frames, (for example Ethernet-Over EtherCAT - EOE). When using an Ethernet packet sniffer such as Wireshark, most Industrial Ethernet frames are identified by checking the ETHERTYPE ID eld. Table 2 illustrates how each of the different Industrial Ethernet types are detected using the Ethernet frame ETHERTYPE ID.

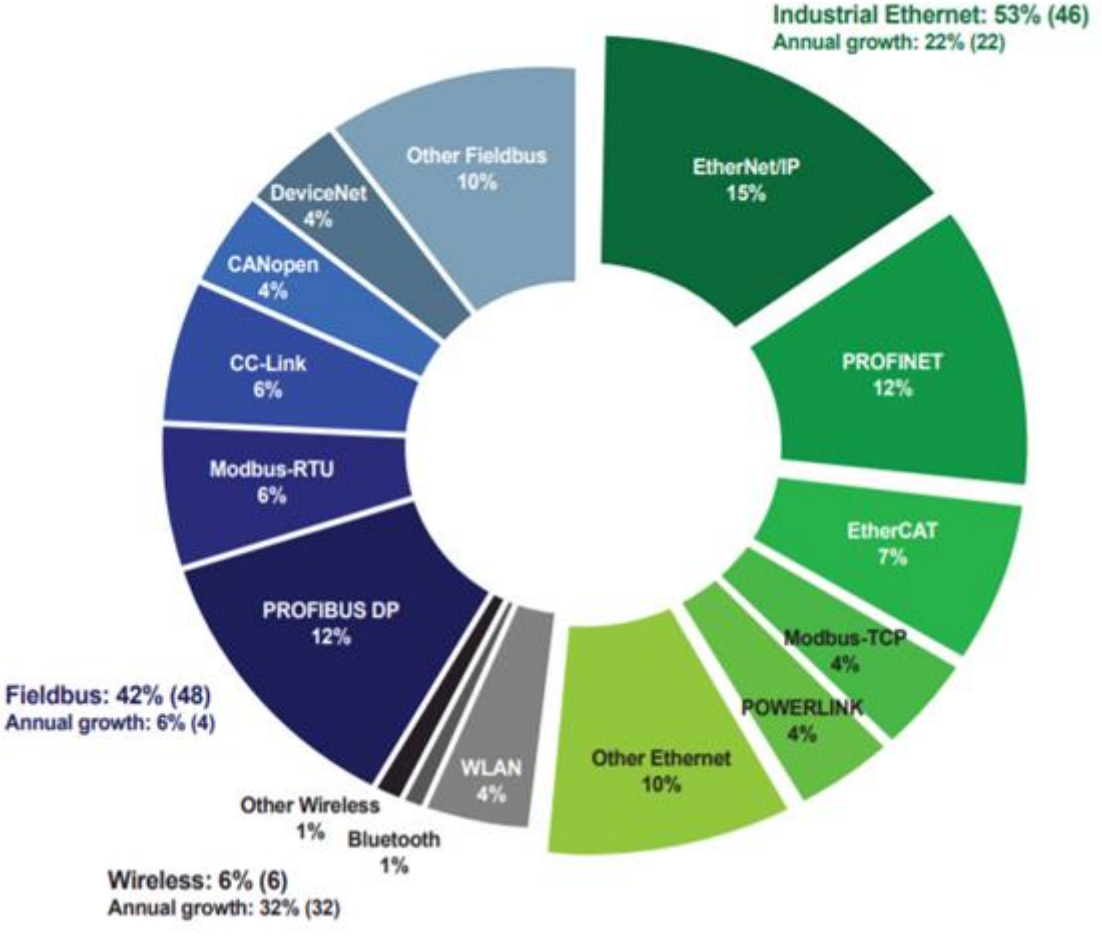


Fig 4.10 : Industrial Ethernet Market Share

Table 4.2 Industrial Ethernet Organizations

Industrial Ethernet Protocol	Organization	Contact
PROFINET	PNO	www.pro bus.com
POWERLINK	ESPG	www.ethernet-powerlink.org
Ethernet/IP	ODVA	www.odva.org
EtherCAT	ETG	www.ethercat.org
SERCOS III	SERCOS International	www.sercos.org
MODBUS TCP	Modbus Organization	www.modbus.org
CC-Link IE	CLPA	www.cc-link.org

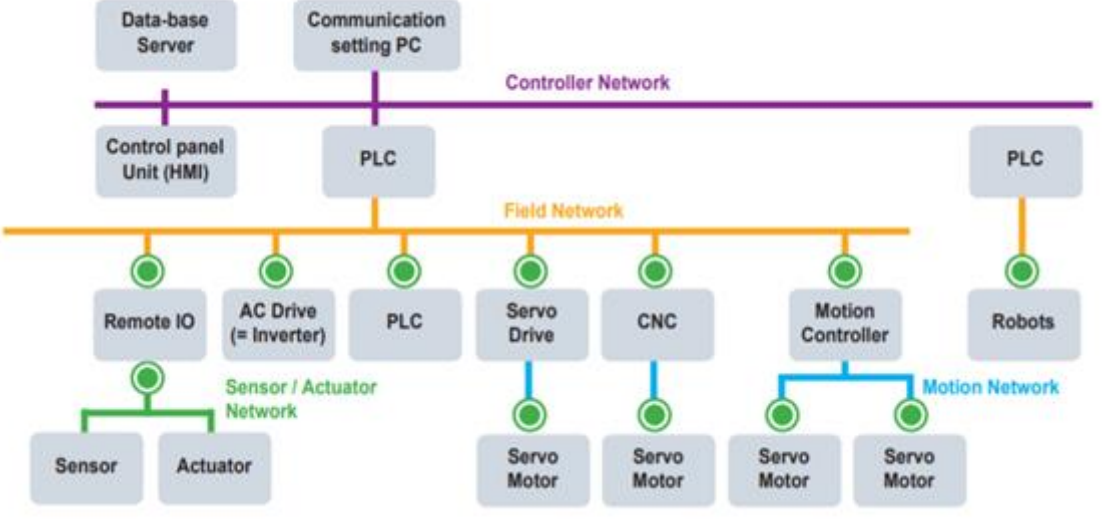


Fig 4.11 : Industrial Automation Network

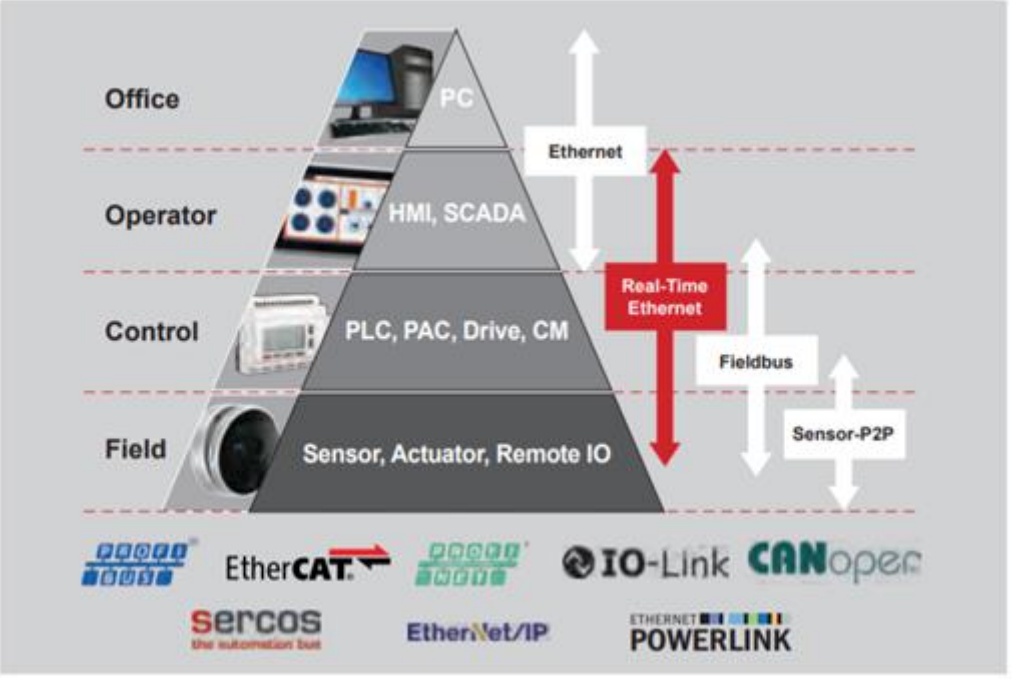


Fig 4.12 : Industrial Automation Pyramid

Text/Reference Books

1. S. Misra, A. Mukherjee, and A. Roy, Introduction to IoT. Cambridge University Press, 2020
2. S. Misra, C. Roy, and A. Mukherjee, Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.2020
3. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015

Question Bank

PART-A

1. Define and describe modbus with block diagram.
2. List the 3 major industrial automation.
3. Discuss bacnet protocol and its importance.
4. 4-20mA is considered for many sensors, convince the statement with appropriate reason.
5. Identify the modes of data transfer in serial communication
6. Enumerate the characteristics of serial communication
7. Point out the benefits of field bus technologies.
8. Distinguish between Ethernet and industrial Ethernet

PART-B

1. Explain the importance and the system architecture of Purdue Control Hierarchy.
2. Describe RS232 Serial Communication Protocol Working and Specifications.
3. Design a MODBUS system to connect 12 devices with corresponding slave address = 15-40, Parity = None, baud 9600 and form a JSON packet to send to the end point 192.168.3.145:2345
4. Explain in details Field Bus Technologies
5. Discuss the steps needed to design a BMS system to obtain Temperature and RH using any of the desired protocols.

INDUSTRIAL INTERNET OF THINGS – SECA4005

UNIT – V MIIDDLEWARE TRANSPORT PROTOCOL

MIIDDLEWARE TRANSPORT PROTOCOL

TCP/IP, UDP, RTP, CoAP –Middleware Software patterns –Software Design patterns – Application Programming Interface (API) – CAN Protocol-Web Services – Middleware IIoT – Securing the IIoT- Identity Access Management.

The TCP/IP protocol layers Like most networking software, TCP/IP is modelled in layers. This layered representation leads to the term protocol stack, which refers to the stack of layers in the protocol suite. It can be used for positioning (but not for functionally comparing) the TCP/IP protocol suite against others, such as Systems Network Architecture (SNA) and the Open System Interconnection (OSI) model. Functional comparisons cannot easily be extracted from this, because there are basic differences in the layered models used by the different protocol suites. By dividing the communication software into layers, the protocol stack allows for division of labour, ease of implementation and code testing, and the ability to develop alternative layer implementations. Layers communicate with those above and below via concise interfaces. In this regard, a layer provides a service for the layer directly above it and makes use of services provided by the layer directly below it. For example, the IP layer provides the ability to transfer data from one host to another without any guarantee to reliable delivery or duplicate suppression. Transport protocols such as TCP make use of this service to provide applications with reliable, in-order, data stream delivery.

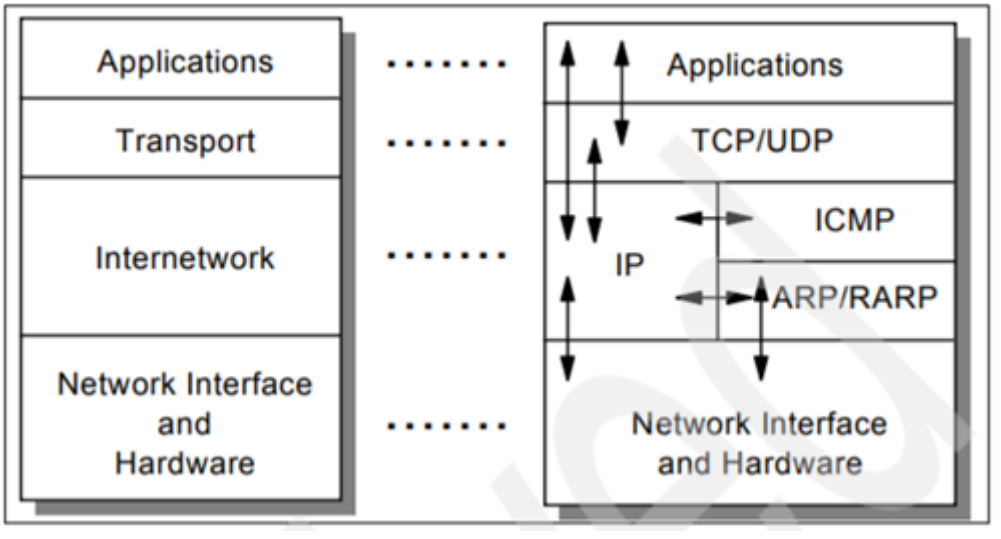


Fig 5.1 : The TCP/IP Protocol Stack : Each layer represents a package of functions

These layers include:

Application layer

The application layer is provided by the program that uses TCP/IP for communication. An application is a user process cooperating with another process usually on a different host (there is also a benefit to application communication within a single host). Examples of applications include Telnet and the File Transfer Protocol (FTP). The interface between the application and transport

layers is defined by port numbers and sockets.

Transport layer

The transport layer provides the end-to-end data transfer by delivering data from an application to its remote peer. Multiple applications can be supported simultaneously. The most-used transport layer protocol is the Transmission Control Protocol (TCP), which provides connection-oriented reliable data delivery, duplicate data suppression, congestion control, and flow control. Another transport layer protocol is the User Datagram Protocol It provides connectionless, unreliable,best-effort service. As a result, applications using UDP as the transport protocol have to provide their own end-to-end integrity, flow control, and congestion control, if desired. Usually, UDP is used by applications that need a fast transport mechanism and can tolerate the loss of some data.

Internetwork layer

The internetwork layer, also called the internet layer or the network layer, provides the "virtual network" image of an internet (this layer shields the higher levels from the physical network architecture below it). Internet Protocol (IP) is the most important protocol in this layer. It is a connectionless protocol that does not assume reliability from lower layers. IP does not provide reliability, flow control, or error recovery. These functions must be provided at a higher level. IP provides a routing function that attempts to deliver transmitted messages to their destination. A message unit in an IP network is called an IP datagram. This is the basic unit of information transmitted across TCP/IP networks. Other internetwork-layer protocols are IP, ICMP, IGMP, ARP, and RARP.

Network interface layer

The network interface layer, also called the link layer or the data-link layer, is the interface to the actual network hardware. This interface may or may not provide reliable delivery, and may be packet or stream oriented. In fact, TCP/IP does not specify any protocol here, but can use almost any network interface available, which illustrates the flexibility of the IP layer. Examples are IEEE 802.2, X.25 (which is reliable in itself), ATM, FDDI, and even SNA. TCP/IP specifications do not describe or standardize any network-layer protocols per se; they only standardize ways of accessing those protocols from the internetwork layer.

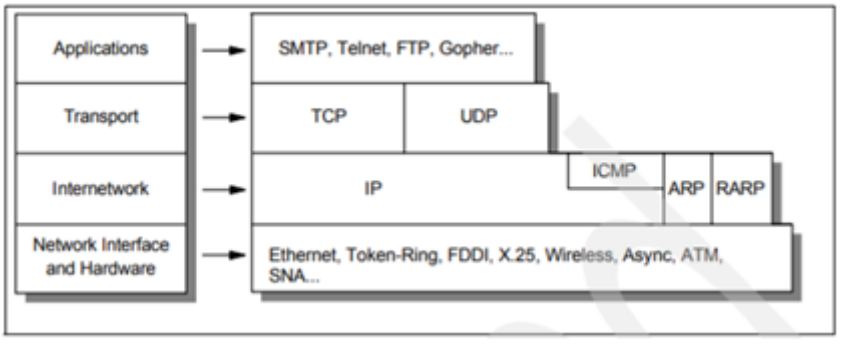
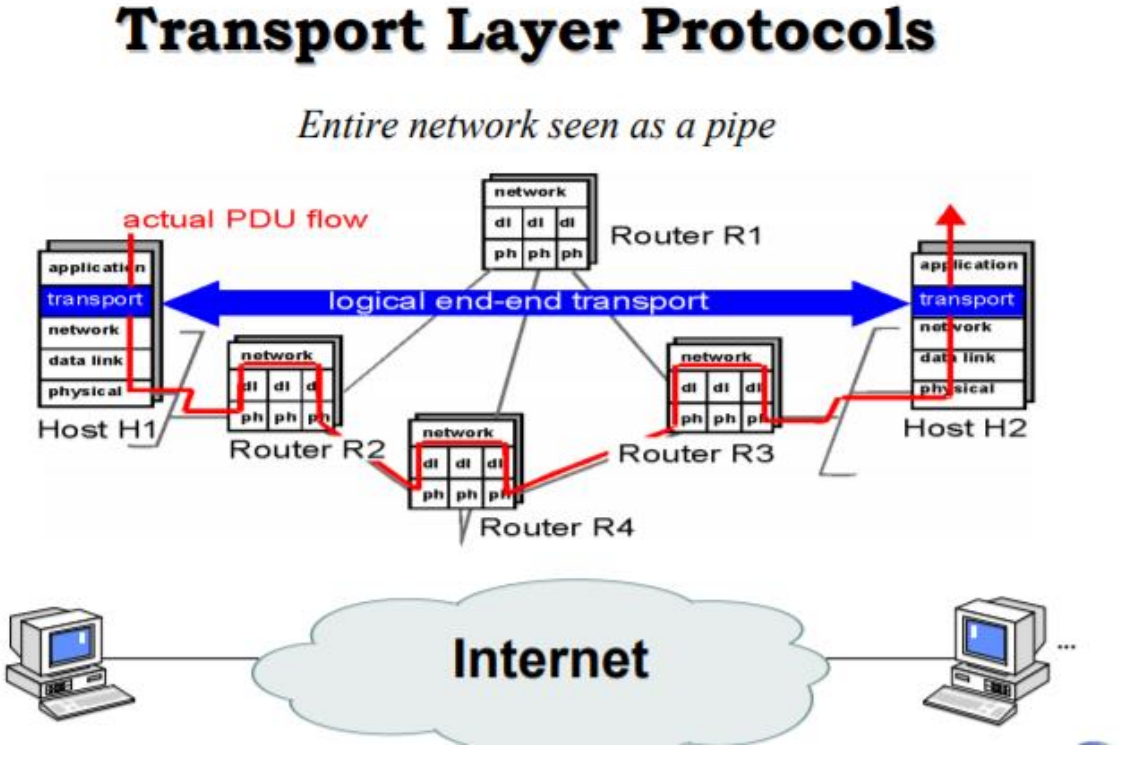


Fig. 5.2 : Detailed architectural model

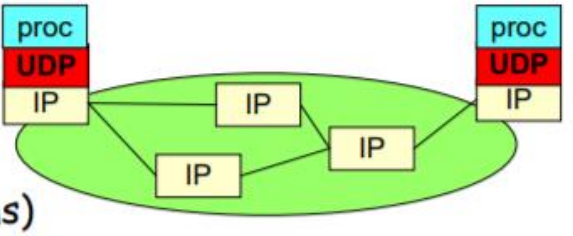
Detailed architectural model 1.1.3 TCP/IP applications

The highest-level protocols within the TCP/IP protocol stack are application protocols. They communicate with applications on other internet hosts and are the user-visible interface to the TCP/IP protocol suite. All application protocols have some characteristics in common: They can be user-written applications or applications standardized and shipped with the TCP/IP product. Indeed, the TCP/IP protocol suite includes application protocols such as: – Telnet for interactive terminal access to remote internet hosts – File Transfer Protocol (FTP) for high-speed disk-to-disk file transfers – Simple Mail Transfer Protocol (SMTP) as an internet mailing system These are some of the most widely implemented application protocols, but many others exist. Each particular TCP/IP implementation will include a lesser or greater set of application protocols. They use either UDP or TCP as a transport mechanism. Remember that UDP is unreliable and offers no flow-control, so in this case, the application has to provide its own error recovery, flow control, and congestion control functionality. It is often easier to build applications on top of TCP because it is a reliable stream, connection-oriented, congestion-friendly, flow control-enabled protocol. As a result, most application protocols will use TCP, but there are applications built on UDP to achieve better performance through increased protocol efficiencies. Most applications use the client/server model of interaction.

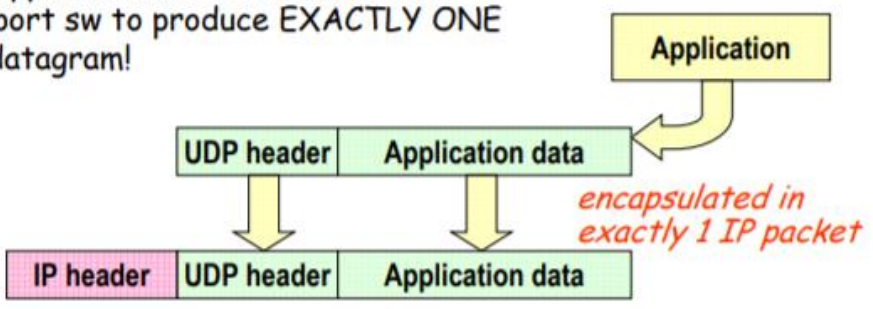


UDP Packets

➔ Connection-Less
- ⇨ (no handshaking)

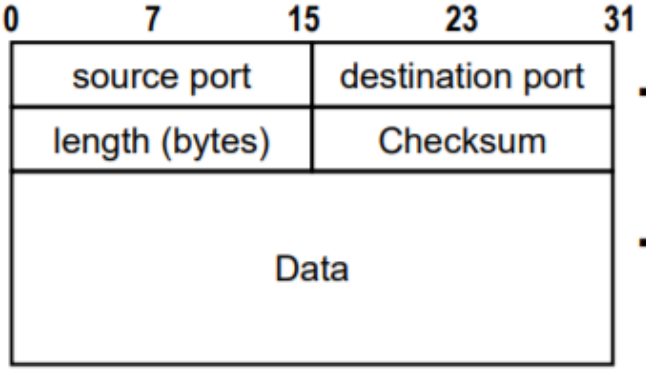


➔ UDP packets (Datagrams)
- ⇨ Each application interacts with UDP transport sw to produce EXACTLY ONE UDP datagram!



UDP datagram format

8 bytes header + variable payload



➔ UDP length field
- ⇨ all UDP datagram
- ⇨ (header + payload)

➔ payload sizes allowed:
- ⇨ Empty
- ⇨ Odd size (bytes)

➔ **UDP functions limited to:**
- ⇨ **addressing**
  - → which is the only strictly necessary role of a transport protocol
- ⇨ **Error checking**
  - → which may even be **disabled** for performance

Maximum UDP datagram size

➔ **16 bit UDP length field:**
- ⇨ Maximum up to 2^{16-1} = 65535 bytes
- ⇨ Includes 8 bytes UDP header (max data = 65527)

➔ **But max IP packet size is also 65535**
- ⇨ Minus 20 bytes IP header, minus 8 bytes UDP header
- ⇨ Max UDP_data = **65507** bytes!

➔ **Moreover, most OS impose further limitations!**
- ⇨ most systems provide 8192 bytes maximum (max size in NFS)
- ⇨ some OS had (still have?) internal implementation features (bugs?) that limit IP packet size
  - ➔ SunOS 4.1.3 had 32767 for max tolerable IP packet transmittable (but 32786 in reception…) – bug fixed only in Solaris 2.2

➔ **Finally, subnet Maximum Transfer Unit (MTU) limits may fragment datagram – annoying for reliability!**
- ⇨ E.g. ethernet = 1500 bytes; PPP on your modem = 576

UDP: a lightweight protocol

➔ **No connection establishment**
- ⇨ no initial overhead due to handshaking

➔ **No connection state**
- ⇨ greater number of supported connections by a server!

➔ **Small packet header overhead**
- ⇨ 8 bytes only vs 20 in TCP

➔ **originally intended for simple applications, oriented to short information exchange**
- ⇨ DNS
- ⇨ management (e.g. SNMP)
- ⇨ Distributed file system support (e.g. NFS)
- ⇨ etc

Unregulated send rate in UDP

- ⇨ No rate limitations
  - → No throttling due to congestion & flow control mechanisms
  - → No retransmission
- ⇨ Less overhead
- ⇨ In contrast to TCP, UDP may provide multicast support

→ extremely important features for today multimedia applications!

→ specially for real time applications which can tolerate some packet loss but require a minimum send rate.

Audio/Video Support

→ UDP is transport layer candidate

→ UDP is too elementary!

- ⇨ No sequence numbers
- ⇨ No timestamp for resynchronization at receiver
- ⇨ No multicasting

→ Old solution: let application developer build their own header

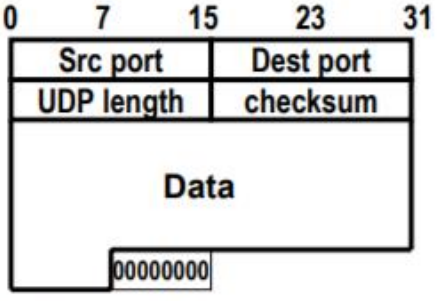
→ New solution: use an enhanced transport protocol

Real Time Protocol (RTP, RFC 3550)

Error checksum

➔ **16 bit checksum field, obtained by:**

- ⇨ summing up all 16 bit words in header data and **pseudoheader**, in 1's complement (checksum fields filled with 0s initially)
- ⇨ take 1's complement of result
- ⇨ if result is 0, set it to 111111…11 (65535==0 in 1's complement)



➔ **at destination:**

- ⇨ 1's complement sum should return 0, otherwise error detected
- ⇨ upon error, no action (just packet discard)

➔ **efficient implementation RFC 1071**

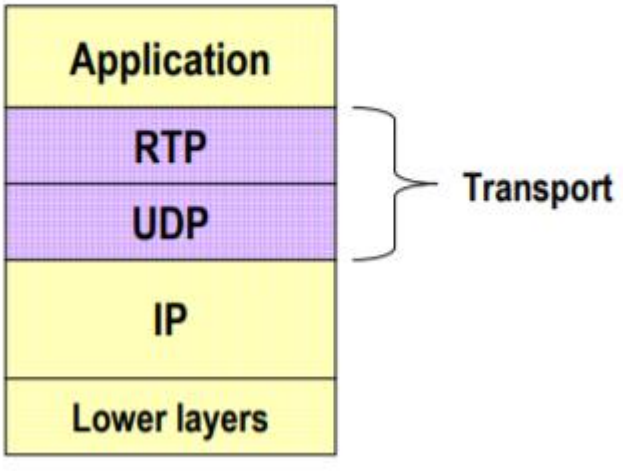
➔ **Zero padding**

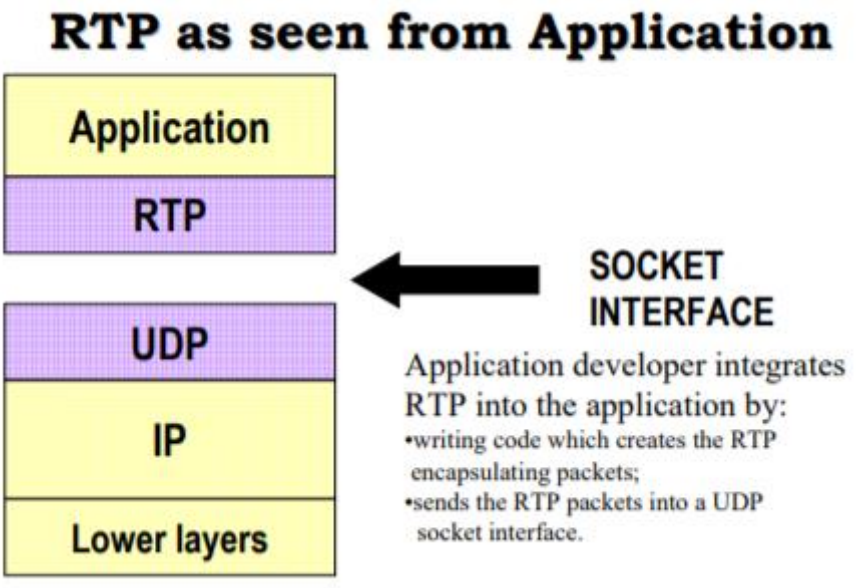
- ⇨ when data size is odd

➔ **checksum disabled**

- ⇨ by source, by setting 0 in the checksum field

RTP: sublayer of Transport





An application programming interface (API) is a specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables. An API specification can take many forms, including an International Standard such as POSIX or vendor documentation such as the Microsoft Windows API, or the libraries of a programming language, e.g. Standard Template Library in C++ or Java API. An API differs from an application binary interface (ABI) in that the former is source code based while the latter is a binary interface. For instance POSIX is an API, while the Linux Standard Base is an ABI.

Language used An API can be:

• language-dependent, meaning it is only available by using the syntax and elements of a particular language, which makes the API more convenient to use.

• language-independent, written so that it can be called from several programming languages. This is a desirable feature for a service-oriented API that is not bound to a specific process or system and may be provided as remote procedure calls or web services. For example, a website that allows users to review local restaurants is able to layer their reviews over maps taken from Google Maps, because Google Maps has an API that facilitates this functionality. Google Maps' API controls what information a third-party site can use and how they can use it.

The term API may be used to refer to a complete interface, a single function, or even a set of APIs provided by an organization. Thus, the scope of meaning is usually determined by the context of usage.

In procedural languages like C language the action is usually mediated by a function call. Hence the API usually includes a description of all the functions/routines it provides. For instance: the math.h include file for the C language contains the definition of the function prototypes of the mathematical functions available in the C language library for mathematical processing (usually called libm). This file describes how to use the functions included in the given library: the function prototype is a signature that describes the number and types of the parameters to be passed to the functions and the type of the return value.

The behaviour of the functions is usually described in more details in a human readable format in printed books or in electronic formats like the man pages: e.g. on Unix systems the command man 3 sqrt will present the signature of the function sqrt in the form:

SYNOPSIS #include double sqrt(double X); float sqrtf(float X); DESCRIPTION sqrt computes the positive square root of the argument. ... RETURNS On success, the square root is returned. If X is real and positive... That means that the function returns the square root of a positive floating point number (single or double precision) as another floating point number. Hence the API in this case can be interpreted as the collection of theinclude files used by the C language and its human readable description provided by the man pages.

API in object-oriented languages In object-oriented languages, an API usually includes a description of a set of class definitions, with a set of behaviours associated with those classes. This abstract concept is associated with the real functionality exposed, or made available, by the classes that are implemented in terms of class methods (or more generally by all its public components hence all public methods, but also possibly including any internal entity made public, like fields, constants, nested objects, enums...).

The API in this case can be conceived as the totality of all the methods publicly exposed by the classes (usually called the class interface). This means that the API prescribes the methods by which one interacts with/handles the objects derived from the class definitions. More generally, one can see the API as the collection of all the kinds of objects one can derive from the class definitions, and their associated possible behaviours. Again: the use is mediated by the public methods, but in this interpretation, the methods are seen as a technical detail of how the behaviour is implemented. For instance: a class representing a Stack can simply expose publicly two methods push() (to add a new item to the stack), and pop() (to extract the last item, ideally placed on top of the stack).

In this case the API can be interpreted as the two methods pop() and push(), or, more generally, as the idea that one can use an item of type Stack that implements the behaviour of a stack: a pile exposing its top to add/remove elements. The second interpretation appears more appropriate in the spirit of object orientation. This concept can be carried to the point where a class interface in an API has no methods at all, but only behaviours associated with it. For instance, the Java language and Lisp (programming language) API include the interface Serializable, which is a marker interface that requires that each class that implements it should behave in a serialized fashion. This does not

require to have any public method, but rather requires that any class that implements it to have a representation that can be saved (serialized) at any time (this is typically true for any class containing simple data and no link to external resources, like an open connection to a file, a remote system, or an external device). Similarly the behaviour of an object in a concurrent (multi-threaded) environment is not necessarily determined by specific methods, belonging to the interface implemented, but still belongs to the API for that Class of objects, and should be described in the documentation.[2] In this sense, in object-oriented languages, the API defines a set of object behaviours, possibly mediated by a set of class methods. In such languages, the API is still distributed as a library. For example, the Java language libraries include a set of APIs that are provided in the form of the JDK used by the developers to build new Java programs. The JDK includes the documentation of the API in JavaDoc notation. The quality of the documentation associated with an API is often a factor determining its success in terms of ease of use.

API libraries and frameworks An API is usually related to a software library: the API describes and prescribes the expected behaviour while the library is an actual implementation of this set of rules. A single API can have multiple implementations (or none, being abstract) in the form of different libraries that share the same programming interface. An API can also be related to a software framework: a framework can be based on several libraries implementing several APIs, but unlike the normal use of an API, the access to the behaviour built into the framework is mediated by extending its content with new classes plugged into the framework itself. Moreover the overall program flow of control can be out of the control of the caller, and in the hands of the framework via inversion of control or similar mechanisms.

API and protocols An API can also be an implementation of a protocol. In general the difference between an API and a protocol is that the protocol defines a standard way to exchange requests and responses based on a common transport and agreeing on a data/message exchange format, while an API (not implementing a protocol) is usually implemented as a library to be used directly: hence there can be no transport involved (no information physically transferred from/to some remote machine), but rather only simple information exchange via function calls (local to the machine where the elaboration takes place) and data is exchanged in formats expressed in a specific language.[4] When an API implements a protocol it can be based on proxy methods for remote invocations that underneath rely on the communication protocol. The role of the API can be exactly to hide the detail of the transport protocol. E.g.: RMI is an API that implements the JRMP protocol or the IIOP as RMI-IIOP. Protocols are usually shared between different technologies (system based on given computer programming languages in a given operating system) and usually allow the different technologies to exchange information, acting as an abstraction/mediation level between the two worlds. While APIs can be specific to a given technology: hencethe APIs of a given language cannot be used in other languages, unless the function calls are wrapped with specific adaptation libraries.

Object API and protocols

An object API can prescribe a specific object exchange format, an object exchange protocol can define a way to transfer the same kind of information in a message sent to a remote system. When a message is exchanged via a protocol between two different platforms using objects on both sides, the object in a programming language can be transformed (marshalled and unmarshalled) in an object in a remote and different language: so, e.g., a program written in Java invokes a service via SOAP or IIOP written in C# both programs use APIs for remote invocation (each locally to the machine where they are working) to (remotely) exchange information that they both convert from/to an object in local memory. Instead when a similar object is exchanged via an API local to a single machine the object is effectively exchanged (or a reference to it) in memory: e.g. via the memory allocated by a single process, or among multiple processes using shared memory or other sharing technologies like topple spaces. API sharing and reuse via virtual machine Some languages like those running in a virtual machine (e.g. .NET CLI compliant languages in the Common Language Runtime and JVM compliant languages in the Java Virtual Machine) can share APIs. In this case the virtual machine enables the language interoperation thanks to the common denominator of the virtual machine that abstracts from the specific language using an intermediate bytecode and its language binding. Hence this approach maximizes the code reuse potential for all the existing libraries and related APIs.

Implementations The POSIX standard defines an API that allows a wide range of common computing functions to be written in a way such that they may operate on many different systems (Mac OS X, and various Berkeley Software Distributions (BSDs) implement this interface); however, making use of this requires re-compiling for each platform. A compatible API, on the other hand, allows compiled object code to function without any changes to the system implementing that API. This is beneficial to both software providers (where they may distribute existing software on new systems without producing and distributing upgrades) and users (where they may install older software on their new systems without purchasing upgrades), although this generally requires that various software libraries implement the necessary APIs as well. Microsoft has shown a strong commitment to a backward compatible API, particularly within their Windows API (Win32) library, such that older applications may run on newer versions of Windows using an executable-specific setting called "Compatibility Mode".[8] Apple Inc. has shown less concern, breaking compatibility or implementing an API in a slower "emulation mode"; this allows greater freedom in development, at the cost of making older software obsolete. Among Unix-like operating systems, there are many related but incompatible operating systems running on a common hardware platform (particularly Intel 80386-compatible systems). There have been several attempts to standardize the API such that software vendors may distribute one binary application for all these systems; however, to date, none of these have met with much success. The Linux Standard Base is attempting to do this for the Linux platform, while many of the BSD Unites, such as FreeBSD, NetBSD, and OpenBSD, implement various levels of API compatibility for both backward compatibility (allowing programs written for older versions to run on newer distributions of the

system) and cross-platform compatibility (allowing execution of foreign code without recompiling).

The CAN Standard CAN is an International Standardization Organization (ISO) defined serial communications bus originally developed for the automotive industry to replace the complex wiring harness with a two-wire bus. The specification calls for signalling rates up to 1 Mbps, high immunity to electrical interference, and an ability to self-diagnose and repair data errors. These features have led to CAN's popularity in a variety of industries including automotive, marine, medical, manufacturing, and aerospace. The CAN communications protocol, ISO 11898, describes how information is passed between devices on a network, and conforms to the Open Systems Interconnection (OSI) model that is defined in terms of layers. Actual communication between devices connected by the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the lowest two layers of the seven layer OSI/ISO model as the data-link layer and physical layer in Figure 5.3.

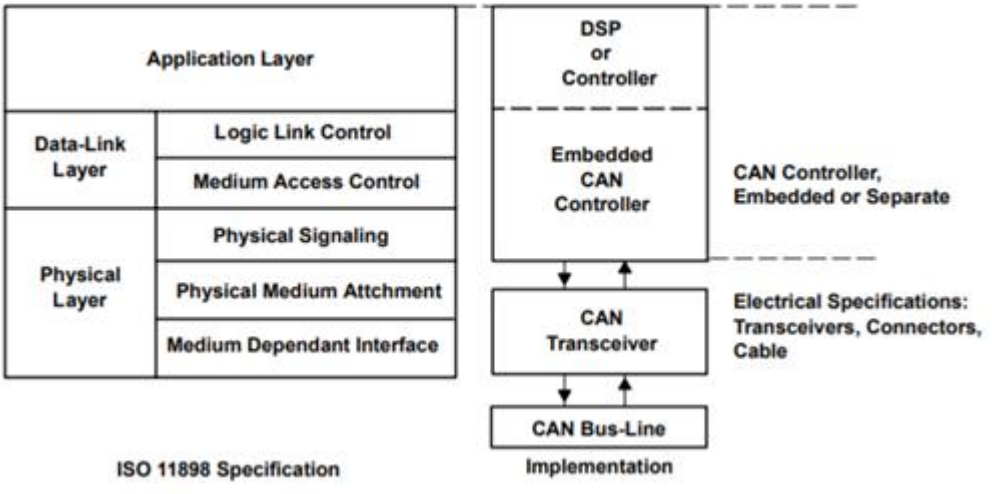


Fig. 5.3 : The Layered ISO 11898 Standard Architecture

In Figure 5.3, the application layer establishes the communication link to an upper-level application specific protocol such as the vendor independent CAN open protocol. This protocol is supported by the international users and manufacturers group, CAN in Automation (CiA). Additional CAN information is located at the CiA website, can-cia.de. There are many similar emerging protocols dedicated to particular applications like industrial automation or aviation. Examples of industry-standard CAN-based protocols are KVASER's CAN Kingdom, Allen-Bradley's DeviceNet and Honeywell's Smart Distributed System (SDS).

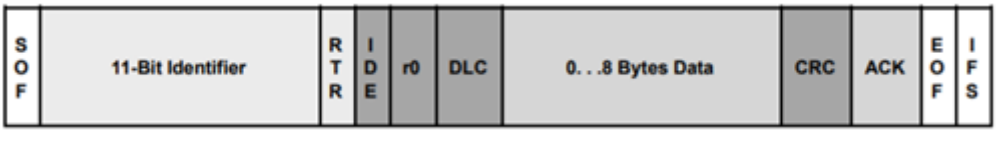


Fig. 5.4 : Standard CAN : 11-bit Identifier

RTR—The single remote transmission request (RTR) bit is dominant when information is required from another node. All nodes receive the request, but the identifier determines the specified node. The responding data is also received by all nodes and used by any node interested. In this way all data being used in a system is uniform.

• IDE—A dominant single identifier extension (IDE) bit means that a standard CAN identifier with no extension is being transmitted.

• r0—Reserved bit (for possible use by future standard amendment).

• DLC—The 4-bit data length code (DLC) contains the number of bytes of data being transmitted.

• Data—Up to 64 bits of application data may be transmitted.

• CRC—The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for error detection.

• ACK—Every node receiving an accurate message overwrites this recessive bit in the original message with a dominate bit, indicating an error-free message has been sent. Should a receiving node detect an error and leave this bit recessive, it discards the message and the sending node repeats the message after rearbitration. In this way each node acknowledges (ACK) the integrity of its data. ACK is 2 bits, one is the acknowledgement bit and the second is a delimiter.

• EOF—This end-of-frame (EOF) 7-bit field marks the end of a CAN frame (message) and disables bit–stuffing, indicating a stuffing error when dominant. When 5 bits of the same logic level occur in succession during normal operation, a bit of the opposite logic level is stuffed into the data.

• IFS—This 7-bit inter-frame space (IFS) contains the amount of time required by the controller to move a correctly received frame to its proper position in a message buffer area.

Industrial IoT Requirements

Essentially, the Industrial IoT involves integrating and then improving the operation of multiple remote assets. Several aspects differentiate the IIoT from consumer applications. The most important difference concerns network security. Security is often cited as the primary barrier to IIoT deployment and growth. IIoT applications involve capital assets that are expensive or impossible to replace and these must be operated securely to protect not only the assets but also human lives and the environment. Thus, security is an absolute must. IIoT applications cannot compromise network security, provide additional cyber-attack vectors, or require specialized configurations such as VPNs or open ports in network firewalls.

Beyond security, Industrial IoT applications also demand scalability. Pilot programs do not always indicate how scalable a solution is, but installations must be easy to replicate and per-unit costs must decrease rapidly as the size of the deployment grows. Without this, the architecture is impractical.

Finally, performance is an important criterion for industrial applications. While consumer IoT

applications often involve very low data rates and may tolerate very high latency, industrial applications must support bursts of data, higher data rates, and low latency. While this may not always be the case, being able to turn up a data rate in industrial applications could unlock much more value.

Skkynet

Enter Skkynet. Skkynet originated over 20 years ago as industrial data sharing software. During that long life, Skkynet has developed into a full-service Industrial IoT middleware, addressing the many new requirements of the IIoT. Skkynet software consists of only three major components:

- DataHub - This is the agent software that runs near or on the physical assets. DataHub requires only a Windows platform for operation. It provides real-time connectivity to and from any number of data sources including the most popular industrial protocols. It also provides connectivity to the major cloud service platforms. For many applications, DataHub can be used by itself in multiple locations to deliver highly secure OPC UA services that extend beyond enterprise firewalls. DataHub also provides a remote configuration tool that creates a single point of configuration for any network of remote DataHub instances.
- DataHub ETK - This embedded software tool kit enables device developers to build DataHub services within industrial devices that run embedded software, rather than Windows. ETK enables these small-footprint devices to provide DataHub functionality directly.
- SkkyHub - This cloud software-as-a-service (SaaS) offering works with installations of DataHub and ETK. The subscription service provides both the software and a near-infinitely scalable backbone for the exchange of real-time data between systems, devices, and client web browsers. The service enables bi-directional supervisory control integration and data sharing among multiple users. Its high scalability derives from the vast resources of commercial public clouds. Perhaps the most important features are that SkkyHub requires no VPN or open firewall ports and no dedicated end user programming.

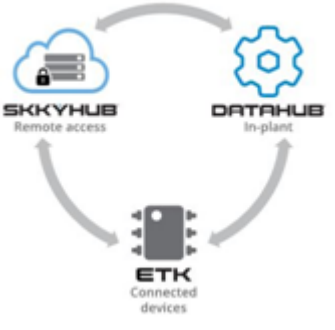


Fig. 5.5 : SKKYNET Components

These three components can be assembled quickly and in different configurations to fit various Industrial IoT application requirements. Not all components are required for any particular solution. Indeed, a number of valuable applications can be configured using DataHub alone.

One illustration of the usefulness of Skkynet software is the number of firms that have become "white label" partners. These partners embed Skkynet software within their own Industrial IoT products and services, which of course carry their own brand. This enables firms such as equipment OEMs to deliver advanced services that leverage their own unique in-house expertise. By using Skkynet, they can do this without being concerned about the details of the secure remote connectivity they need. Instead, they can focus on application areas where they alone can deliver greater value to their IoT customers. One client told ARC, "We are not aware of other off-the-shelf software that can do this."

Returning to the wind farm example that opened this discussion, the figure below shows how Skkynet middleware can enable multiple parties to collaborate for remote assets. For internal use, Datahub services on two wind farms serve the operational needs and the IT requirements of the wind farm operator. At the same time, DataHub feeds the cloud where selected sets of data are shared with wind turbine OEMs for analytics and with other third-party suppliers. None of these parties has direct network access to the wind farm site, and each is authorized to subscribe to a restricted data set, relevant to their particular role or service. This architecture is extremely useful in cases where multiple OEMs need access to operational information and in situations where capital assets are owned by a joint venture but operated by a single firm. In both cases, detailed operational data must be shared among several parties, but access to the capital asset itself must be restricted to the operator.

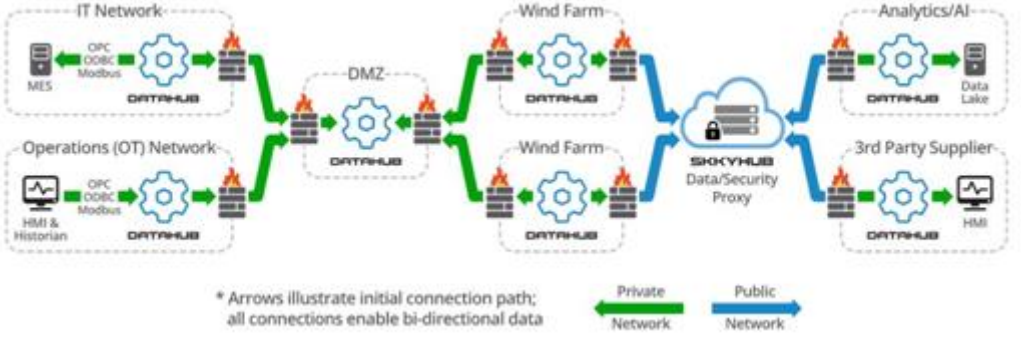


Fig. 5.6 : Skkynet Multi-party Wind Farm Solution

Recommendations

- Equipment OEMs should normally partner for the high-performance wide area connectivity required for advanced aftermarket services.

- Owner-operators should recognize that middleware has made the leap into the IoT age, and so it should be considered for a role in Industrial IoT strategies.
- System and service suppliers should note that wide area data services can be incorporated within their own branded IoT services.

Identity Management (IdM)

IdM manages an identity's lifecycle through a combination of processes, organizational structure, and enabling technologies.

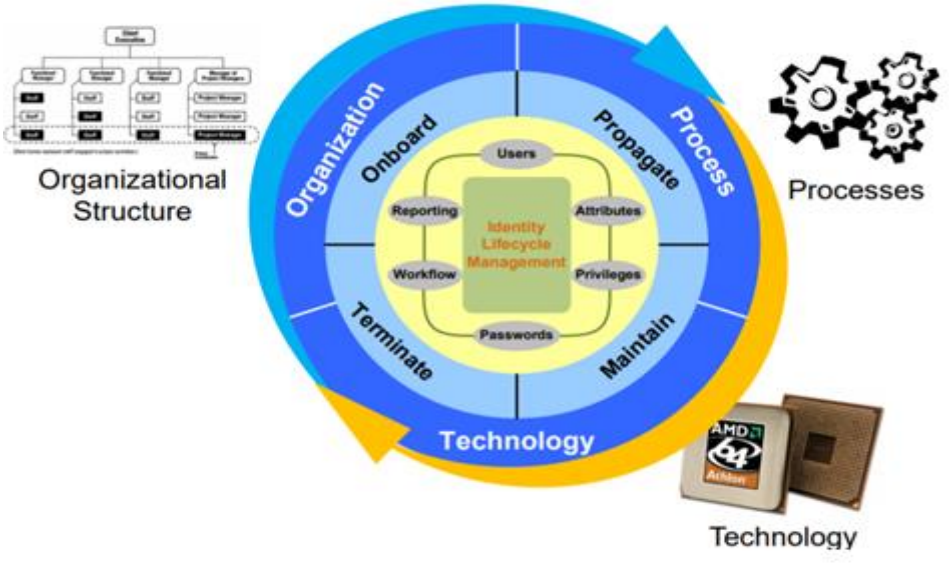
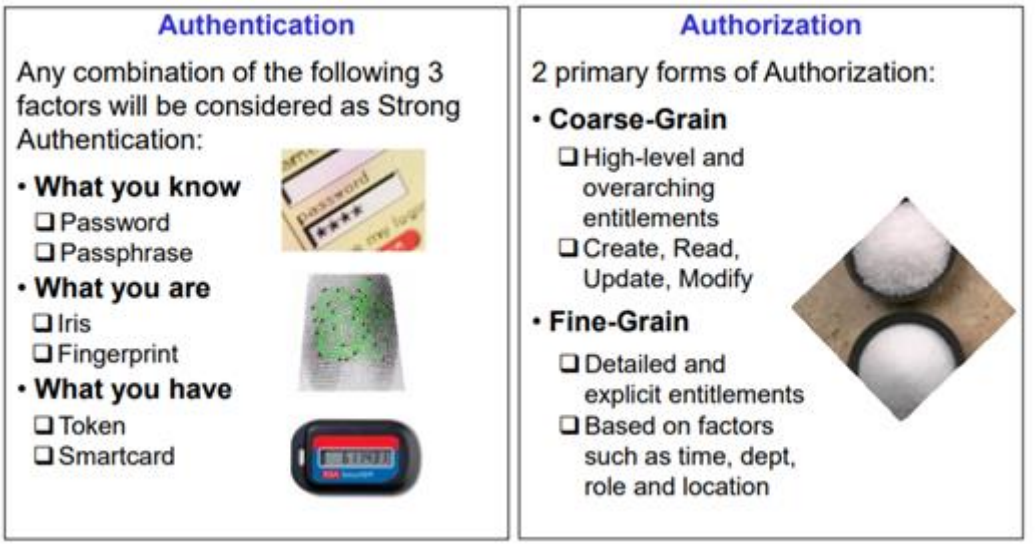


Fig 5.7 : Identity management

Access Management (AM)

AM Primarily focuses on Authentication and Autherization



Uniting Identity and Access Management

Identity and Access management are tightly coupled by the governance and consumption of identity data.

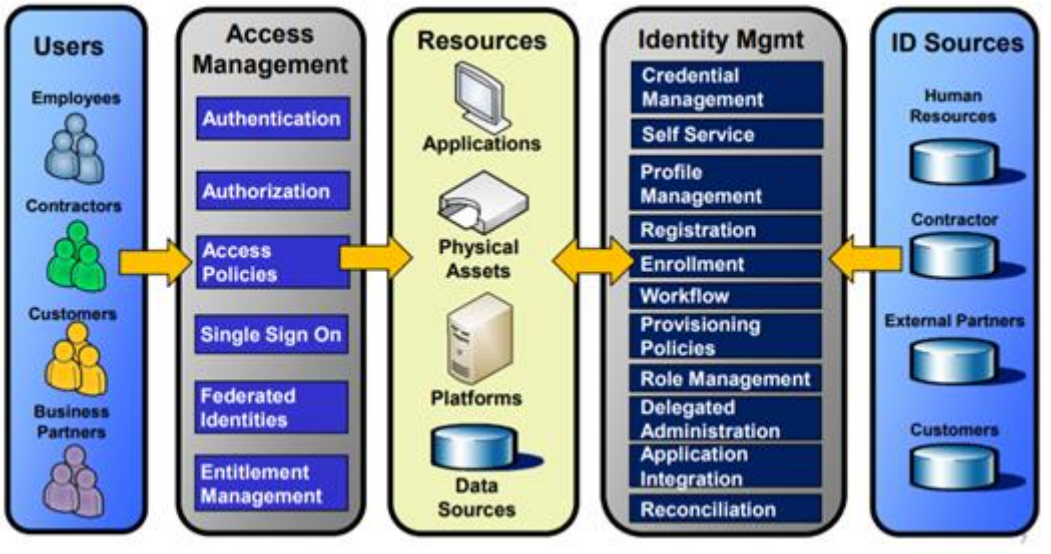


Fig. 5.8 : Uniting Identity and Access Management

Typical IT Architecture

- Multiple Identity stores
- Multiple administration points
- Redundant data synchronization and replication
- Users must authenticate to each application

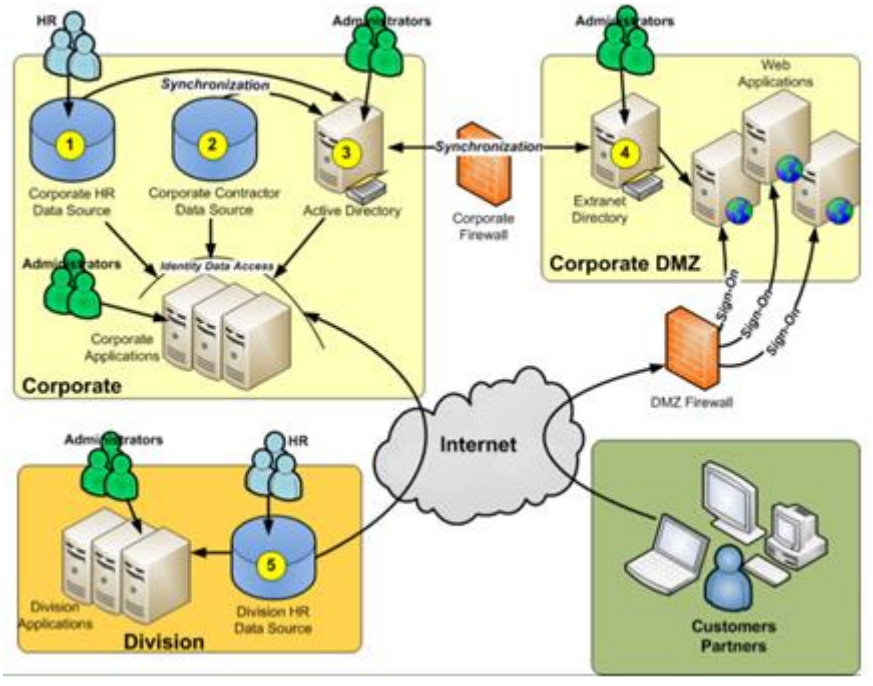


Fig. 5.9 : IT Architecture

I & AM Architecture

- Single Identity Store
- Ability to present multiple data views
- Single Administration Point
- Reduced replication and synchronization
- Single Sign-On

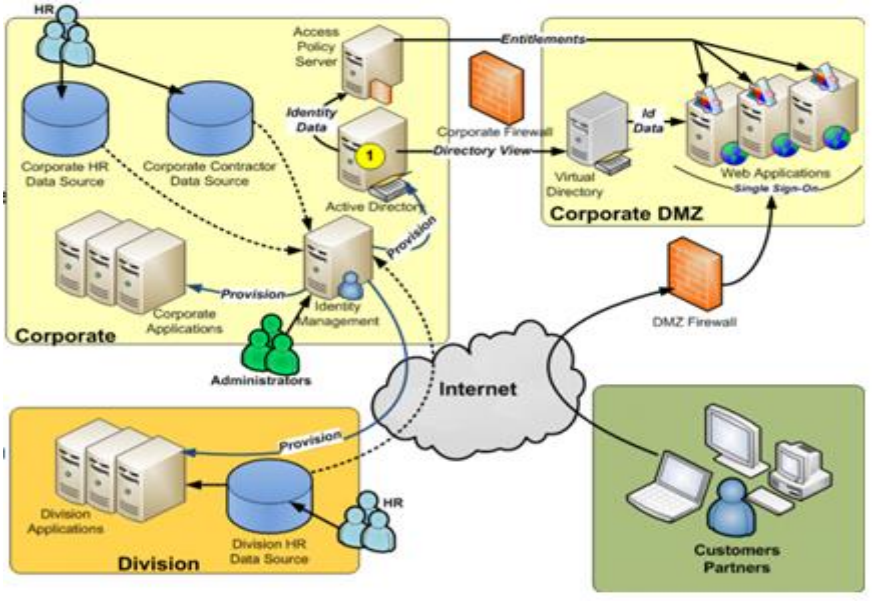


Fig. 5.10 : I & AM Architecture

Text/Reference Books

1. S. Misra, A. Mukherjee, and A. Roy, Introduction to IoT. Cambridge University Press, 2020
2. S. Misra, C. Roy, and A. Mukherjee, Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.2020
3. Dr. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015

Question Bank

PART-A

1. TCP/IP or HTTP which one is more suitable for long range transmission? Judge and state the reason.
2. List the security models employed in IOT.
3. Define multi drop mode.
4. Point out the primary reason for introducing CoAP over TCP and TLS
5. Demonstrate the middleware pattern.
6. List the examples of an Application Programming Interface.
7. Identify the different frames on a CAN bus
8. Summarize the different components of web services

PART-B

1. Explain in detail the architecture of TCP/IP Protocol in IIOT.
2. Discuss MQTT Middleware protocol with AUTOSAR with a neat sketch
3. Describe the seven software design patterns and explain with an example.
4. Describe in details about Application Programming Interface (API)
5. Demonstrate web service and how does a web service work