Chapter No. 5 Fog computing programming languages and frameworks [06 hrs]

- Middleware and software platforms
- Development and deployment considerations
- **Industrial Internet of Things(IIoT)**
- Performance Evaluation and Metrics in Fog Computing
- Simulation and modelling techniques
- **Applications and Use Cases of Fog Computing**

Self-Learning Topics: Development environments and Frameworks for programming in Fog Computing.

Middleware and software platforms

Edge computing and fog computing have combined in a way to facilitate a wide variety of applications that involve human interactions, which are geographically distributed and have stringent real-time performance requirements . The Internet of Things (IoT) or Internet of Everything (IoT) has introduced edge devices that now obtain information from user environment and need to respond intelligently to changes in real time. The scale of an application has increased from mere single mobile device to a large number of edge devices that are geographically distributed and change locations dynamically. Even though cloud support can be used in processing the data generated by the edge devices, the delay incurred in communication to cloud devices is excessively more than the real-time constraints of some of the latency sensitive applications. With such a large scale of data being generated in geographically distributed locations, sending the code toward the data is in some cases more efficient than processing in the cloud. Fog computing introduced computation solution in the form of fog devices, cloudlets, and mobile edge computing (MEC), which provide computation services in the network edge. It can also meet the real-time requirements of such applications. Apart from the components that pertain to application logic, there are large design components that perform the underlying task of managing the network , computation, and resources of fog and edge architecture (FEA). Due to the dynamically changing context in the edge devices, underlying algorithms for managing the execution and processing data become complex. In addition to the processing and data communication of the distributed application, the control data and algorithm decisions incur excess resources when they are executed on the edge devices.

Need for Fog and Edge Computing Middleware

Fog and edge computing are gaining acceptance due to high availability , low latency and low cost. Domains such as smart cities, virtual reality and entertainment, vehicular systems use edge processing for real-time operations .The efficient design of middleware enables the realization of full potential of fog and edge infrastructures. Middleware handles different tasks such as communication management, network management, task scheduling, mobility management, and security management, thereby reducing the complexity of the distributed mobile application design.

Middleware design of fog/edge infrastructure is challenging because of the stringent application requirements such as
(i) availability of context on the sensing devices;
(ii) cost of data transfer and processing in different tiers of FEA;
(iii) limitations on number of edge devices present and dynamic changes in context and mobility of the devices;
(iv) strict latency constraints.

Including the dynamically changing context of a user and capturing the user interactions patterns can essentially enable intelligent and informed execution of the applications.

Design Goals:

A varied class of mobile applications can utilize FEA middleware. Requirements for emerging applications can be summarized as following:

1. Newer distributed applications increasingly demand a large number of resources and low latency to meet the real-time response constraints. While the use of cloud has eased the implementation of large-scale distributed mobile applications, in many newer applications, the strict real-time response may not be feasible unless processing is done near the edge.

2. Geo-distributed edge applications such as monitoring oil plant and electricity grid management are geographically distributed. Processing enormous sensor generated data streams in real-time constraints require a large processing facility, but also incur huge communication infrastructure or bandwidth. Edge infrastructure can reduce the communication overhead
involved in large data streams.

3. Large-scale distributed management applications such as connected railways and smart grids involve processing huge data in real time to provide control towards reliable operation. Increasing real-time monitoring and analytical processing in edge can adapt the system itself to dynamic faults and changes.

4. Smart and connected applications such as real-time traffic monitoring and connected vehicles can leverage local edge infrastructure for fast and real-time updates and response related to locally sensed data.

Even though the FEA can support different types of applications, common functionalities that are required in such applications can be provided by the middleware.

Following are design goals of FEA middleware.

6.3.1 Ad-Hoc Device Discovery

Data sources in fog/edge may belong to a wide category of devices ranging from IoT sensors, mobile devices to fixed sensors. The data are processed locally or sent to fog/edge devices for further processing. A channel of communication needs to be set up between the requesting devices and ad-hoc discovered devices that perform the application task of acquisition and processing. Once a communication channel is set up, it allows dynamically changing set of devices to join and participate. Given the dynamic nature of participating edge devices that acquire and process the data, the device discovery allows setup of a communication layer to enable further communication between devices.

6.3.2 Run-Time Execution Environment

The middleware provides a platform that executes the application task remotely on the edge devices. Functionality includes code download, remote execution in the edge devices, and delivery of results such that they are available to the requesting device.

6.3.3 Minimal Task Disruption

Task disruption during execution affects the reliability of execution of FEA task. Often it results in reinitialization of the task or undesirable/unavailable results. Device usage patterns, mobility, and network disconnections may cause unexpected changes in the context of the device. This may render the device inappropriate for continuing the execution of sensing or computation task. Anticipatory techniques can be used to minimize the interruption in tasks, thereby promoting intelligent scheduling decisions.

6.3.4 Overhead of Operational Parameters

Establishing communication between ad-hoc edge devices, selection of candidate edge devices, distribution of FEA tasks between multiple edge devices, and managing remote execution in a sequence of FEA tasks incurs additional usage of bandwidth and energy consumption on the edge devices. As these resources are expensive, minimizing these operational parameters is an important aspect of middleware operations. Additionally, several devices may enforce usage limit on their resources that are available for sharing.

6.3.5 Context-Aware Adaptive Design

Innovative contexts such as the mental state of and user activity are now used in mobile applications for sensing useful data. For successful execution in FEA, dynamic changes in the context of the devices as well its environment require the middleware to adapt to these changes. Self-adaptive services can enhance its operations and improve the FEA quality of service.

6.3.6 Quality of Service

Quality of service (QoS) of an architecture is highly dependent on the application. Many edge/fog applications use multidimensional data for achieving specific goals. Acquiring and processing such huge sensor data within real-time constraints is a requirement for these applications. Real-time response is an important QoS measure. Other application specific QoS parameters can be the relevance of the acquired data, its correctness, and uninterrupted data acquisition.

6.4 State-of-the-Art Middleware Infrastructures

Applications in fog and edge computing are discussed in some of the recent works. Real-time data streaming applications include traffic monitoring Waze, smart traffic light systems ,real-time replay in the stadium,and video analytics. Real-time applications that process the requests for emergency rescue in disaster and searching for missing persons.Application of geographically distributed systems such as wind farms and smart vehicle-to-vehicle systems are becoming popular in fog and edge computing. Common requirements of these applications present a need for middleware to support easy design and development of such applications. Middleware features such as security, mobility, context awareness and data analytics addressed in recent research are shown in Table 6.1. Popular IoT platforms such as GoogleFit [15, 16] have a cloud-based IoT middleware for smartphones. Provisioning of sensing services on the mobile devices is discussed in M-Sense. Service oriented middleware like GSN are proposed for processing data in the distributed environment. Further, Carregaet al. propose microservices-based middleware using a user interface.

Table 6.1 Middleware features in fog and edge architectures.

	Devices	Security	Mobility support	Context awareness	Data analytics	Optimized selection of devices
FemtoCloud [21]	Mobile	N	N	Y	Y	Y
Nakamura et al. [22]	Mobile and sensor	N	N	N	Y	N
Aazam et al. [27]	Fog, MEC, Cloud	Y	N	N	Y	Y
Bonomi et al. [9]	Fog, Cloud	N	Y	N	Y	Y
Verbelen et al. [28]	Mobile Cloudlet	N	N	N	N	Y
Cloudaware [26]	Cloudlet	N	Y	Y	N	Y
Hyrax [10]	Cloud	N	Y	N	N	Y
Grewe et al. [14]	MEC	Y	Y	N	Y	Y
Carrega et al. [20]	MEC, Fog	Y	Y	N	Y	Y
Piro et al. [16]	Cloud	Y	Y	Y	Y	Y

In FemtoCloud system, the mobile devices in the edge can be configured to provide the services to requesting devices [21]. "Process on Our Own (PO3)" concept where a data stream generated on each device is processed on itself is proposed by Nakamura et al. [22]. CoTWare middleware proposed by Jaroodi et al. suggests a novel way of integrating things, fog devices, and cloud by using cloud-hosted services to manage processing of IoT data in fog resources [23]. MobiPADs [24] and MobiCon [25] are context-aware middleware solutions that enable adaptive design in mobile applications by reconfiguring the services with respect to dynamic changes in context for edge devices. Recently proposed CloudAware [26] is an example of adaptive middleware for constantly changing context such as connectivity for cloudlet.

6.6.4 Middleware

Components of middleware commonly used in fog and edge applications are discussed in the following subsections.

6.6.4.1 Context Monitoring and Prediction

FEA can adapt to dynamic changes in the user environment using the context-aware design of middleware. This may involve continuous monitoring of relevant context and adaptive actions that are based on changes in the context. Also, recent research shows that several human-dependent contexts have patterns. These patterns can be learned to intelligently manage the operations between multiple devices. Several techniques such as time series, stochastic, or machine learning can be used to model and predict the human-mobile contextual changes.

6.6.4.2 Selection of Participating Devices

FEA employs devices from the environment that can sense and/or process the data acquired in the FEA applications. Selection of the surrogate device can be based on different policies designed in the middleware. Research shows several policies such as fairness-based selection, game theoretic [8], context optimization [44], and resource optimization approaches that are used in surrogate selection. Participating users are selected based on different criterion ranging from simple user context such as the location of the device to selection based on the reputation of user task completion history [45].

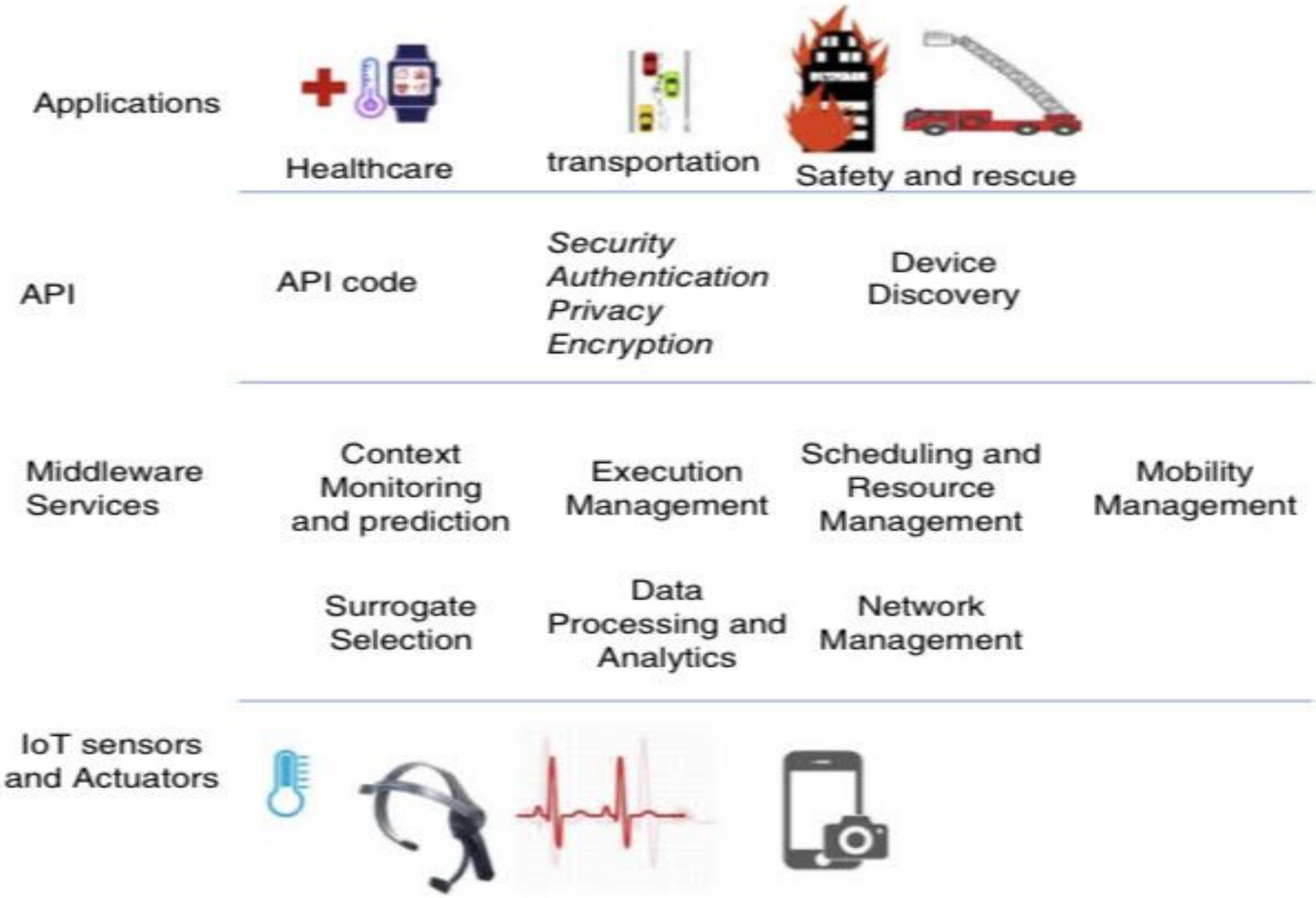


Figure 6.2 Fog and edge computing architecture.

Following are different surrogate selection techniques.

Energy-Aware Selection. Remaining battery is critical to every mobile user and it determines the amount of resources that the device owner may share. Selection of surrogates is a trade-off between the quality of information gathered and the remaining battery on the device with an incentive budget [46]. Delay Tolerance-Based Selection. Real-time applications and streaming data application require the processing to be completed in a given time constraint [12]. Performance-based selection of surrogates is proposed in incentivized schemes by Petri et al. [47]. Context-Aware Selection. Context-aware functionality is used in many mobile applications. Applications are designed to adapt themselves based on changes in context on the mobile device or that of the user. Recently proposed context-aware recruitment scheme focused on improving the mobile selection based on context requirements of the application [44]. In applications like crowdsensing apart from individual context prediction, large-scale activity prediction such as proposed in [48] is now becoming useful. Change in location of mobile users can be modeled using different techniques such as random waypoint model and statistical models such as Markov [49, 50]. Spatiotemporal model of user location is proposed by Wang et al. [51].

6.6.4.3 Data Analytics

FEA introduces the idea of processing near the edge. Extensive analytics may be involved in an application that is processing across different layers in the architecture. Some of the analytics tasks can also be used to extract the essential information from the raw data obtained on the user devices. This not only reduces the processing requirements centrally but also reduces the communication costs. Data analytics module on the user device can be used to send essential data towards a central server [52]. Cloud server/data center may be used to aggregate information and process high-level data analytics tasks. Bonomi et al. [9] discuss processing data analytics tasks in multiple use cases in a fog/edge environment.

6.6.4.4 Scheduling and Resource Management

This engine works continuously to monitor the incoming tasks and their assignment using the surrogate selection policy. It monitors the availability of resources in different layers such as the availability of new, incoming user devices as well as tenant resources such as VMs that process data in fog devices and the cloud.

6.6.4.5 Network Management

FEA uses the multitier network to distribute the fog and edge applications.It may use software-defined networking or virtual network topology in multitenant resources in fog and cloud devices. User devices are usually connected using point-to-point network topologies that may either use TCP socket – WiFi connection, WiFi direct, or Bluetooth communication. This module is also responsible for monitoring connection and triggering the connection resume procedures for a lost connection.

6.6.4.6 Execution Management

This module facilitates the application specific code functionality to executeon the edge and fog nodes. Existing work in fog computing proposed the use of a virtual environment [28] or use of private OS stack provided by CISCO iox [53]. Virtualization with migration support on mobile devices is proposed by Bellavista et al. [54]. In some research, the code offload techniques such as DEX compositions in android [55] or .NET may be used. Other works propose plug-in based designs that are downloaded and integrated into the app in runtime [56].

6.6.4.7 Mobility Management

MEC supports mobile edge devices that are constantly on the move. In such cases, the data and the middleware services follow the devices. The idea is commonly known as Follow me Cloud (FMC) [57] and uses Locator/ID separation (LISP) protocol.

6.6.5 Sensor/Actuators

The sensors handle the important task of obtaining real-time data from the environment and user’s surrounding. The information obtained through sensors is used in several forms. Sensor data may be acquired in the FEA application itself. It can also be used to evaluate and extract context information of the device user. In more complex applications, the closed-loop information is acquired and analyzed and further used for taking real-time actions using the actuators.

Q) What is Industrial IoT or IIoT?

IoT and edge computing use cases Industrial IoT, or IIoT, refers to the use of IoT in an industrial context, such as the machines in a factory. Think of the lifecycle of heavy machinery used in a factory.Different people may stress equipment differently over time, and breakdowns are an expected part of operations.IoT sensors can be added to parts of the machinery that are most prone to breaking or overuse. The data from these sensors can be analyzed and used for predictive maintenance, reducing overall downtime.

Autonomous vehicles are an example of why IoT solutions and edge computing need to work together. An autonomous vehicle driving down the road needs to collect and process real-time data about traffic, pedestrians, street signs and stop lights, as well as monitor the vehicle's systems.

If the vehicle needed to stop or turn quickly to avoid an accident, sending data back and forth from the vehicle to the cloud to be processed would take too long. Edge computing brings cloud computing services to the vehicle, allowing the IoT sensors in the vehicle to process the data locally in real-time to avoid an accident.IIoT stands for Industrial Internet of Things, a term for connected devices in manufacturing, energy, and other industrial practices. IIoT is significant for bringing more automation and self-monitoring to industrial machines, helping improve efficiency.

The similarities between IIoT and IoT are:

- Allow for connectivity and communication of devices over the internet.
- They use sensors, devices, and networks to collect data.
- Increase efficiency and automate tasks by allowing devices to communicate and share information.
- They both rely on cloud computing and data analytics. This will help to process large amounts of data
- generated by connected devices.

Features	IIoT	IoT
Devices	Industrial machines and equipment.	Smartwatches, home appliances, cars, and other consumer-facing devices.
Purpose	For specific industrial processes such as monitoring and maintenance.	To improve convenience and efficiency in everyday life.
Security	Requires a high level of security and reliability.	Security and reliability levels vary depending on the device.
Power	High power and expensive devices.	Low-power and low-cost devices.

How is IIoT related to edge computing?

IIoT devices are often deployed in connection with edge computing.Edge computing refers to a strategy of shifting computing resources nearer to the physical location of either the user or the source of the data. By placing computing services closer to these locations, users benefit from faster, more reliable services while companies benefit from the flexibility of hybrid cloud computing. Edge computing is one way that a company can use and distribute a common pool of resources across a large number of locations.

It’s common for IIoT devices to be used for edge computing. For example, in a factory setting,machines that gather data for the purpose of real-time data analytics on site would represent an IIoT use case that supports an edge computing strategy.

Achieving these benefits requires an underlying platform that can unify disparate data systems—especially because manufacturing systems traditionally have been isolated from each other.

Under a unified system, manufacturing sites can deploy artificial intelligence and machine learning (AI/ML) model training through a scalable service platform.The combination of IIoT and edge computing is helping manufacturers solve problems faster by transforming operations,assisting end users in making business decisions, and making plants even more productive.

Q)What’s the difference between IIoT and IoT?

The IoT, or Internet of Things, is a general term for everyday objects which connect to a network,sending and receiving data to and from other devices. The IIoT is a subsection of the IoT.Generally, the IoT is made up of any kind of equipment that takes advantage of Internet connectivity in order to send data and receive data. When that equipment is used for industrial purposes, it is considered IIoT.Consumers IoT devices include products such as connected home thermostats, lights, and door locks. Industrial IoT devices span a wide range of items—everything from water meters to factory machines to sensors on pipelines.

Q) What does IIoT look like in action?

IIoT solutions have a wide variety of use cases, with manufacturing and energy being two of the primary industries to use IIoT.In manufacturing, providing a view of factory conditions is a common example. Sensor data from machinery, analyzed in real time and fed back to control systems, can lead to improved levels of operational and business efficiency.

In energy, companies can use IIoT to better monitor their field assets. IIoT devices can gather real-time data on electrical grid performance, pipeline flow, or emissions monitoring, even with assets distributed across wide geographic areas.

For example, a water and sewage utility service in Italy uses connected self-service water kiosks across a region to gather real-time data on water quality.

Q) How does IIoT automation work?

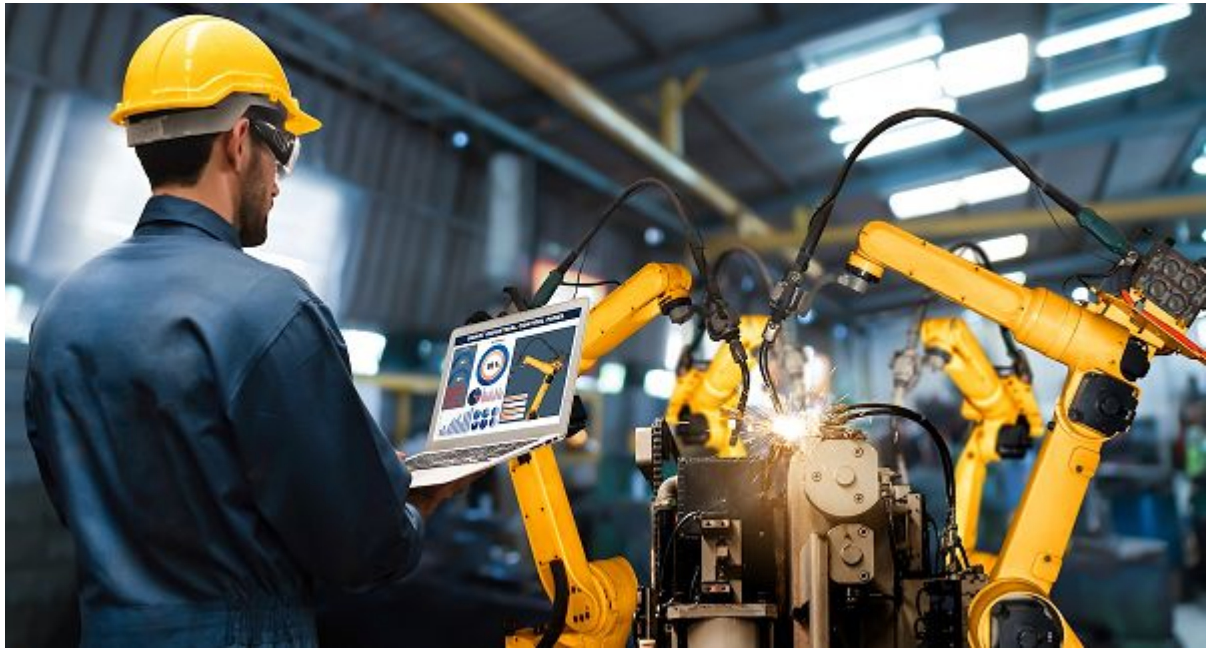
IIoT and automation are tightly linked. Data gathered by IIoT devices can prompt automated tasks that improve efficiency, such as predictive maintenance. Additionally, automation tools can be used to more effectively manage large numbers of IIoT devices.

Using the example of factory machinery, a machine can be programmed to respond to data from onboard sensors—such as noting an increase in vibrations—and automatically take an action—such as alerting an operator that maintenance is required. In ways such as this, IIoT-driven automation can minimize downtime and reduce overall maintenance costs.

Automation can also help with the challenge of managing a large number of IIoT devices, especially ones scattered across large geographic areas—at the edge. Just as automation software can manage servers and network devices, it can also be used to keep IIoT devices updated and validated.

1. Different Industries Manufacturing

IIoT enables connected devices to gather and analyze data for manufacturing processes. The sensor data can track machine performance. It can also detect potential issues before they cause problems.Production is more efficient and flexible when devices and machines communicate with robots.



Agriculture

One specific industrial internet of things example application in agriculture is Precision Agriculture. Precision Agriculture uses devices like drones to gather data from the farming process.

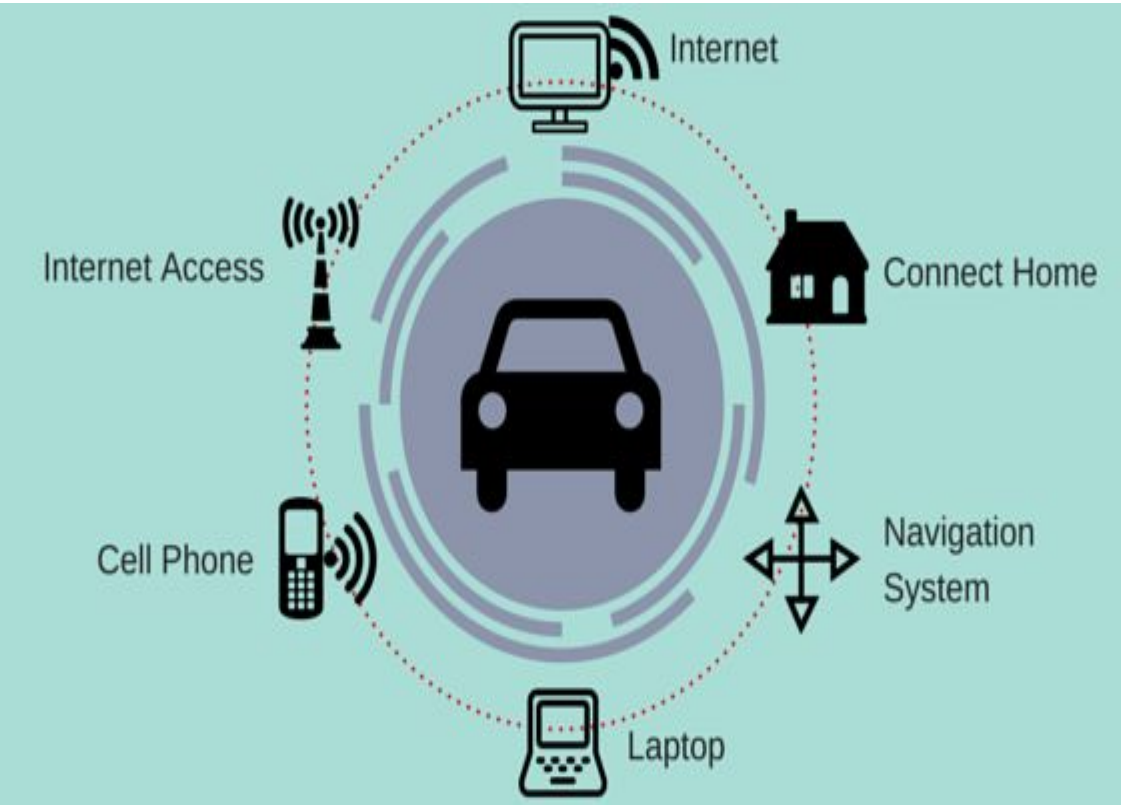
This data can help to optimize crop production, reducing waste and improving efficiency. For example, sensor data can track soil moisture and nutrient levels.

Another critical industrial internet of things example in agriculture applications is automation and robotics in farming. Autonomous tractors help plant and harvest, reduce waste, and increase crop yields.

Automotive

An industrial internet of things example applicable in the automotive industry is the presence of "Connected Cars." Connected cars use various devices, such as sensors, cameras, and GPS, to scan data from the vehicle.

This data is to improve safety, reduce emissions, and optimize performance. For example, sensor data can track tire pressure. It can also predict traffic patterns to optimize routing. The data is then used for predictive analytics to improve fuel efficiency and reduce emissions.



2. Industrial IoT Examples in Action

The internet of things transforms industries. They allow machines and devices to communicate in sync. By integrating IoT with industrial processes, companies can improve efficiency, productivity, and safety. Some examples of industrial IoT use cases in action include:

- In the Manufacturing Industry:

  GE has implemented IIoT technology to optimize production processes. This has led to a reduction in downtime and the number of defects. It also increases productivity.

- In the Transportation Industry:

  Routes and supply chain operations are better using IIoT sensors and tracking devices. For example, industrial IoT use cases can be observed in companies like Daimler and Volvo.IIoT helps to improve the efficiency of their vehicles. Daimler uses IIoT sensors to check the health of truck engines. It also helps to predict when they will need maintenance.

- In the Agriculture Industry:

  John Deere is using IIoT technology to improve the efficiency of its farming equipment.(An American corporation that manufactures agricultural machinery, heavy equipment, forestry machinery, diesel engines, drivetrains used in heavy equipment, and lawn care equipment).For example, the development of tractors that a farmer can control in a remote setting. Based on data from sensors, their speed and direction can adjust to suit the farmer.

- In the Healthcare Industry:

  IoT equipment can check patients' condition from a far. It also improves outcomes in a remote setting. **Medtronic** is one of the best industrial IoT use cases in the medical field. They use IIoT technology to track patients' vital signs. It then sends an alert to healthcare professionals if any changes need attention.

Q) What are the features of IIoT?

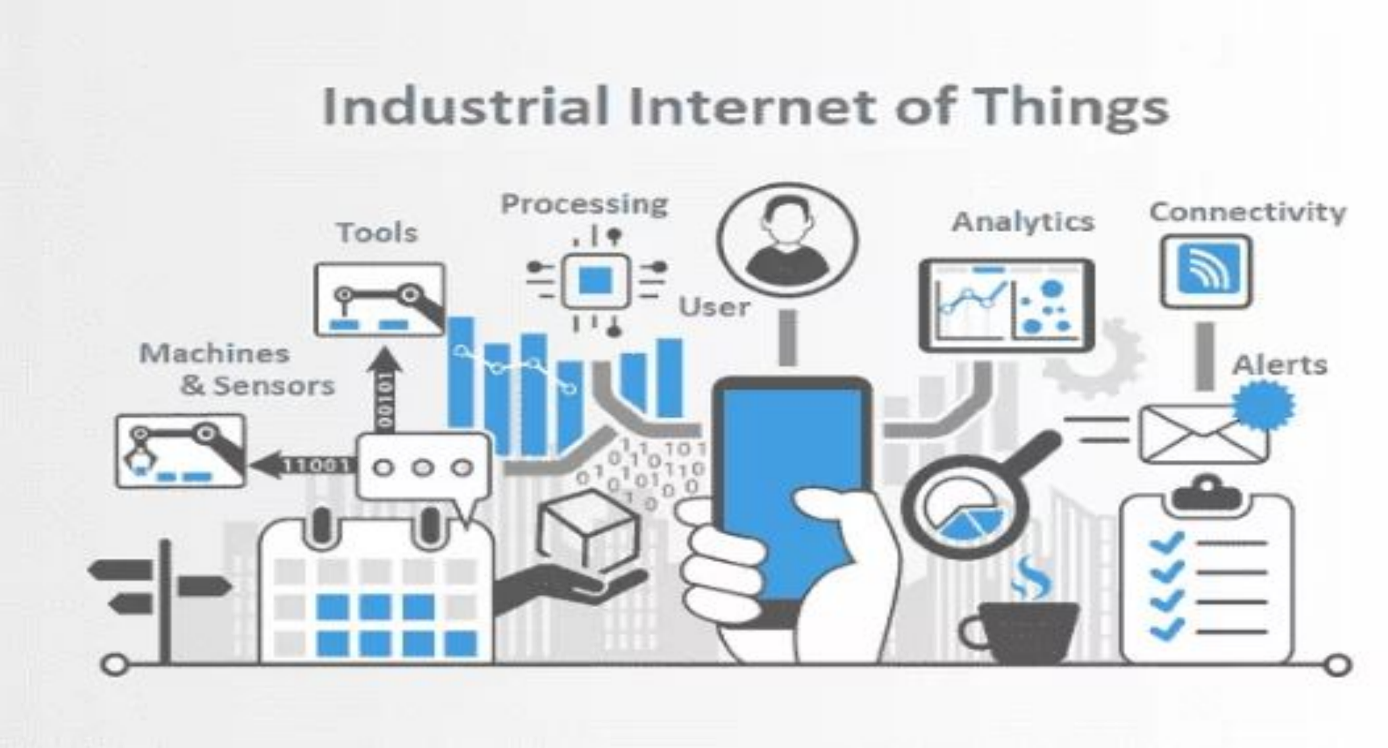
3. The main features of Industrial Internet of Things (IIoT) solutions

Businesses depend on the internet of things to increase efficiency and productivity. The main features of IIoT are:

Smart Alerts and Notifications

Smart alerts and notifications are critical features of IoT. In manufacturing, it allows for real-time monitoring and control of manufacturing assets. Preset conditions trigger equipment failure alerts. It also helps with production process deviations or safety hazards.

These are then sent to the relevant personnel via various communication channels (e.g., SMS, email, push notifications). IIoT allows for quick response and resolution of issues by manufacturing personnel. It improves total efficiency and reduces downtime in manufacturing operations.



Condition Monitoring & Remote Maintenance

It allows for real-time monitoring and control of equipment and other manufacturing assets. This technology allows the collecting of data and analyzing it to detect potential issues.

Remote maintenance enables technicians to access and diagnose equipment. This leads to a reduction of costs associated with on-site repairs. It allows for proactive maintenance and improved equipment uptime. It also increases efficiency in manufacturing operations.

Location Tracking & Geofencing

Enterprise's asset includes equipment, tools, and personnel. This technology can be used to track the location of these assets, as well as set up virtual boundaries that trigger alerts when crossed.

This enables manufacturers to improve operational efficiency and increase productivity in manufacturing.

Data Processing and Analyzing

IIoT allow for analyzing large amounts of data from various manufacturing equipment and personnel. This data can be used to optimize production processes and reduce downtime. It can help to recognize patterns. It can also help to determine trends in the data. This will improve quality control and increase efficiency in manufacturing operations.

Benefits of Industrial Internet of Things (IIoT)

- ❑ **It reduces manual processes:** Allows for device automation when connected to the internet. This leads to a reduction in manual intervention and increases efficiency.
- ❑ **Reduce errors:** IIoT collects and analyzes data in real-time. This makes it easy to identify and correct errors quickly, reducing the risk of costly mistakes.
- ❑ **Optimize operational processes:** IIoT helps to optimize and analyze data. This leads to an increase in the efficiency of operational processes and reduces costs.
- ❑ **Reduce costs:** IIoT reduces costs by automating manual processes. It also reduces errors and optimizes operational processes.
- ❑ **Worker Safety and Productivity:** IIoT helps to automate dangerous and repetitive tasks. This improves worker safety and increases productivity.
- ❑ **Sustainability:** IIoT can help companies reduce their environmental impact. This can be done by optimizing operational processes and automating manual tasks.

Challenges of IIoT

The Industrial Internet of Things (IIoT) has the potential to revolutionize various industries. They also introduce several security challenges. For example:

- ❑ **Network security:** IoT devices are connected to the internet. This becomes a potential entry point for hackers to gain access to a company's network.
- ❑ **Data security:** IoT devices collect large amounts of data. This increases the risk of data breaches and unauthorized access to sensitive information.
- ❑ **Privacy:** IoT devices contain large amounts of personal data. This can be used for malicious purposes if it falls into the wrong hands.

Market Segments of IIoT

The Industrial Internet of Things (IIoT) market is a rapidly growing and diverse field that can be segmented in various ways. This allows companies better to understand the specific needs of their target customers. They can also develop products and services that meet those needs.

The IIoT market segments are based on:

- **Device and Technology:** They include sensors and industrial robots.
- **Connectivity Technology:** They include Wi-Fi, Bluetooth, Zigbee, Z-Wave, and cellular networks.
- **Software:** They include platform software, analytics software, security software, and others.
- **Vertical:** It is present in manufacturing, transportation, healthcare, agriculture, and others.

Why is IoT Growing So Rapidly?

The Internet of Things (IoT) is increasing due to a combination of factors. These factors include:

Advancements in technology.

- The increasing availability of data.
- The growing demand for connected devices.
- Declining costs of IoT devices.
- Connectivity options.
- Increase in IoT application.
- Government support.
- Growing demand for automation.
- Development of new technologies like 5G.

All these are making it easier and more cost-effective to connect devices to the internet.Additionally, it is possible to extract valuable insights from IoT data. This can be done using the growing availability of data. The increasing capabilities of analytics software make IoT popular.

IoT for Smart Cities: Use Cases and Implementation Strategies

Table of contents

- Use Cases
- Road traffic
- Smart parking
- Public transport
- Utilities
- Street lighting
- Waste management
- Environment
- Public safety
- Iterative implementation
- Adapting IoT strategy

https://www.scnsoft.com/blog/iot-for-smart-city-use-cases-approaches-outcomes#road-traffic

Road traffic

Smart cities ensure that their citizens get from point A to point B as safely and efficiently as possible. To achieve this, municipalities turn to IoT development and implement smart traffic solutions.

Smart traffic solutions use different types of sensors, as well as fetch GPS data from drivers' smart phones to determine the number, location and the speed of vehicles. At the same time, smart traffic lights connected to a cloud management platform allow monitoring green light timings and automatically alter the lights based on current traffic situation to prevent congestion. Additionally, using historical data, smart solutions for traffic management can predict where the traffic could go and take measures to prevent potential congestion.

Smart parking

With the help of GPS data from drivers' smartphones (or road-surface sensors embedded in the ground on parking spots), smart parking solutions determine whether the parking spots are occupied or available and create a real-time parking map. When the closest parking spot becomes free, drivers receive a notification and use the map on their phone to find a parking spot faster and easier instead of blindly driving around.

Public transport

The data from IoT sensors can help to reveal patterns of how citizens use transport. Public transportation operators can use this data to enhance traveling experience, achieve a higher level of safety and punctuality. To carry out a more sophisticated analysis, smart public transport solutions can combine multiple sources, such as ticket sales and traffic information.

In London, for instance, some train operators predict the loading of train passenger cars on their trips in and out of the city. They combine the data from ticket sales, movement sensors, and CCTV cameras installed along the platform. Analyzing this data, train operators can predict how each car will load up with passengers. When a train comes into a station, train operators encourage passengers to spread along the train to maximize the loading. By maximizing the capacity use, train operators avoid train delays.

Utilities

IoT-equipped smart cities allow citizens to save money by giving them more control over their home utilities. IoT enables different approaches to smart utilities:

Smart meters & billing

With a network of smart meters, municipalities can provide citizens with cost-effective connectivity to utilities companies' IT systems. Now, smart connected meters can send data directly to a public utility over a telecom network, providing it with reliable meter readings. Smart metering allows utilities companies to bill accurately for the amount of water, energy and gas consumed by each household.

Revealing consumption patterns

A network of smart meters enables utilities companies to gain greater visibility and see how their customers consume energy and water. With a network of smart meters, utilities companies can monitor demand in real time and redirect resources as necessary or encourage consumers to use less energy or water at times of shortage.

Remote monitoring

IoT smart city solutions can also provide citizens with utility management services. These services allow citizens to use their smart meters to track and control their usage remotely. For instance, a householder can turn off their home central heating using a mobile phone. Additionally, if a problem (e.g., a water leakage) occurs, utilities companies can notify householders and send specialists to fix it.

Street lighting

IoT-based smart cities make maintenance and control of street lamps more straightforward and cost-effective. Equipping streetlights with sensors and connecting them to a cloud management solution helps to adapt lighting schedule to the lighting zone.

Smart lighting solutions gather data on illuminance, movement of people and vehicles, and combine it with historical and contextual data (e.g., special events, public transport schedule, time of day and year, etc.) and analyze it to improve the lighting schedule. As a result, a smart lighting solution “tells” a streetlight to dim, brighten, switch on or switch off the lights based on the outer conditions.

For instance, when pedestrians cross the road, the lights around the crossings can switch to a brighter setting; when a bus is expected to arrive at a bus stop, the streetlights around it can be automatically set brighter than those further away, etc.

Waste management

Most waste collection operators empty containers according to predefined schedules. This is not a very efficient approach since it leads to the unproductive use of waste containers and unnecessary fuel consumption by waste collecting trucks.

IoT-enabled smart city solutions help to optimize waste collecting schedules by tracking waste levels, as well as providing route optimization and operational analytics.

Each waste container gets a sensor that gathers the data about the level of the waste in a container. Once it is close to a certain threshold, the waste management solution receives a sensor record, processes it, and sends a notification to a truck driver’s mobile app. Thus, the truck driver empties a full container, avoiding emptying half-full ones.

Environment

IoT-driven smart city solutions allow tracking parameters critical for a healthy environment in order to maintain them at an optimal level. For example, to monitor water quality, a city can deploy a network of sensors across the water grid and connect them to a cloud management platform. Sensors measure pH level, the amount of dissolved oxygen and dissolved ions. If leakage occurs and the chemical composition of water changes, the cloud platform triggers an output defined by the users. For example, if a Nitrate (NO3-) level exceeds 1 mg/L, a water quality management solution alerts maintenance teams of contamination and automatically creates a case for field workers, who then start fixing the issue.

Another use case is monitoring air quality. For that, a network of sensors is deployed along busy roads and around plants. Sensors gather data on the amount of CO, nitrogen, and sulfur oxides, while the central cloud platform analyzes and visualizes sensor readings, so that platform users can view the map of air quality and use this data to point out areas where air pollution is critical and work out recommendations for citizens.

Public safety

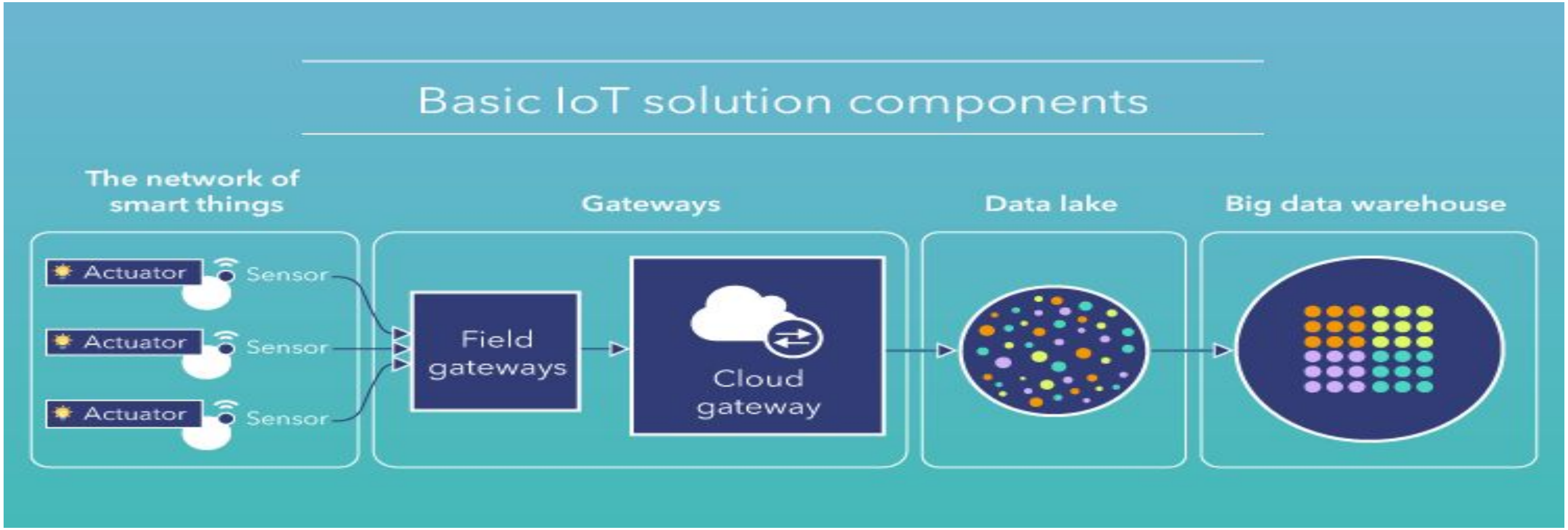
For enhancing public safety, IoT-based smart city technologies offer real-time monitoring, analytics, and decision-making tools. Combining data from acoustic sensors and CCTV cameras deployed throughout the city with the data from social media feed and analyzing it, public safety solutions can predict potential crime scenes. This will allow the police to stop potential perpetrators or successfully track them.

For example, more than 90 cities across the United States use a gunshot detection solution. The solution uses connected microphones installed throughout a city. The data from microphones passes over to the cloud platform, which analyzes the sounds and detects a gunshot. The platform measures the time it took for the sound to reach the microphone and estimates the location of the gun. When the gunshot and its location are identified, cloud software alerts the police via a mobile app.

Iterative approach to implementing smart city solutions

The range of smart city applications is highly diverse. What they have in common is the approach to implementation. Whether municipalities plan to automate waste collection or improve street lighting, they should start with the foundation – a basic smart city platform. If a municipality prefers to expand the range of smart city services in the future, it will be possible to upgrade the existing architecture with new tools and technologies without having to rebuild it.

Here is a six-step implementation model to follow for creating an efficient and scalable IoT architecture for a smart city.



Stage 1: basic IoT-based smart city platform

To be able to scale, smart city implementation should start with designing a basic architecture – it will serve as a springboard for future enhancements and allow adding new services without losing functional performance. A basic IoT solution for smart cities includes four components:

The network of smart things

A smart city – as any IoT system – uses smart things equipped with sensors and actuators. The immediate goal of *sensors* is to collect data and pass it to a central cloud management platform. *Actuators* allow devices to act - alter the lights, restrict the flow of water to the pipe with leakage, etc.

Gateways

Any IoT system comprises two parts – a “tangible” part of IoT devices and network nodes and a cloud part. The data cannot simply pass from one part to the other. There must be doors – *field gateways*. Field gateways facilitate data gathering and compression by preprocessing and filtering data before moving it to the cloud. *The cloud gateway* ensures secure data transmission between field gateways and the cloud part of a smart city solution.

Data lake

The main purpose of a data lake is to store data. Data lakes preserve data in its raw state. When the data is needed for meaningful insights, it’s extracted and passed over to the big data warehouse.

Big data warehouse

A big data warehouse is a single data repository. Unlike data lakes, it contains only structured data. Once the value of data has been defined, it’s extracted, transformed and loaded into the big data warehouse. Moreover, it stores contextual information about connected things, e.g., when sensors were installed, as well as the commands sent to devices’ actuators by control applications.

Stage 2: Monitoring and basic analytics

With data analytics, it is possible to monitor devices' environment and set rules for control applications (we cover them at stage 4) to carry out a particular task.

For example, analyzing the data from soil moisture sensors deployed across a smart park, cities can set rules for the electronic valves to close or open based on the identified moisture level. The data collected with sensors can be visualized on a single platform dashboard, allowing users to know the current state of each park zone.

Stage 3: Deep analytics

Processing IoT-generated data, city administrations can go beyond monitoring & basic analytics and identify patterns and hidden correlations in sensor data. Data analytics uses advanced techniques like machine learning (ML) and statistical analysis. ML algorithms analyze historical sensor data stored in the big data warehouse to identify trends and create predictive models based on them. The models are used by control applications that send commands to IoT devices' actuators. Here is how it applies in practice.

Unlike a traditional traffic light that is programmed to display a particular signal for a definite period, a smart traffic light can adapt signal timings to the traffic scenario. ML algorithms are applied to historical sensor data to reveal traffic patterns and adjust signal timings, helping to improve average vehicle speed and avoid congestions.

Stage 4: Smart control

Control applications ensure better automation of smart city objects by sending commands to their actuators. Basically, they "tell" actuators what to do to solve a particular task. There are rule-based and ML-based control applications. Rules for rule-based control applications are defined manually, while ML-based control applications use models created by ML algorithms. These models are identified based on data analysis; they are tested, approved and regularly updated.

Stage 5: Instant interacting with citizens via user applications

Along with the possibility of automated control, there should always be an option for users to influence the behavior of smart city applications (for example, in case of emergency). This task is carried out by user applications.

User applications allow citizens to connect to the central smart city management platform to monitor and control IoT devices, as well as receive notifications and alerts. For example, using GPS data from drivers' smartphones, a smart traffic management solution identifies a traffic jam. To prevent even bigger congestion, the solution automatically sends a notification to the drivers in the area, encouraging them to take a different route.

At the same time, employees at a traffic control center who use a desktop user app receive a 'congestion alert.' To relieve the congestion and re-route part of the traffic, they send a command to the traffic lights' actuators to alter the signals.

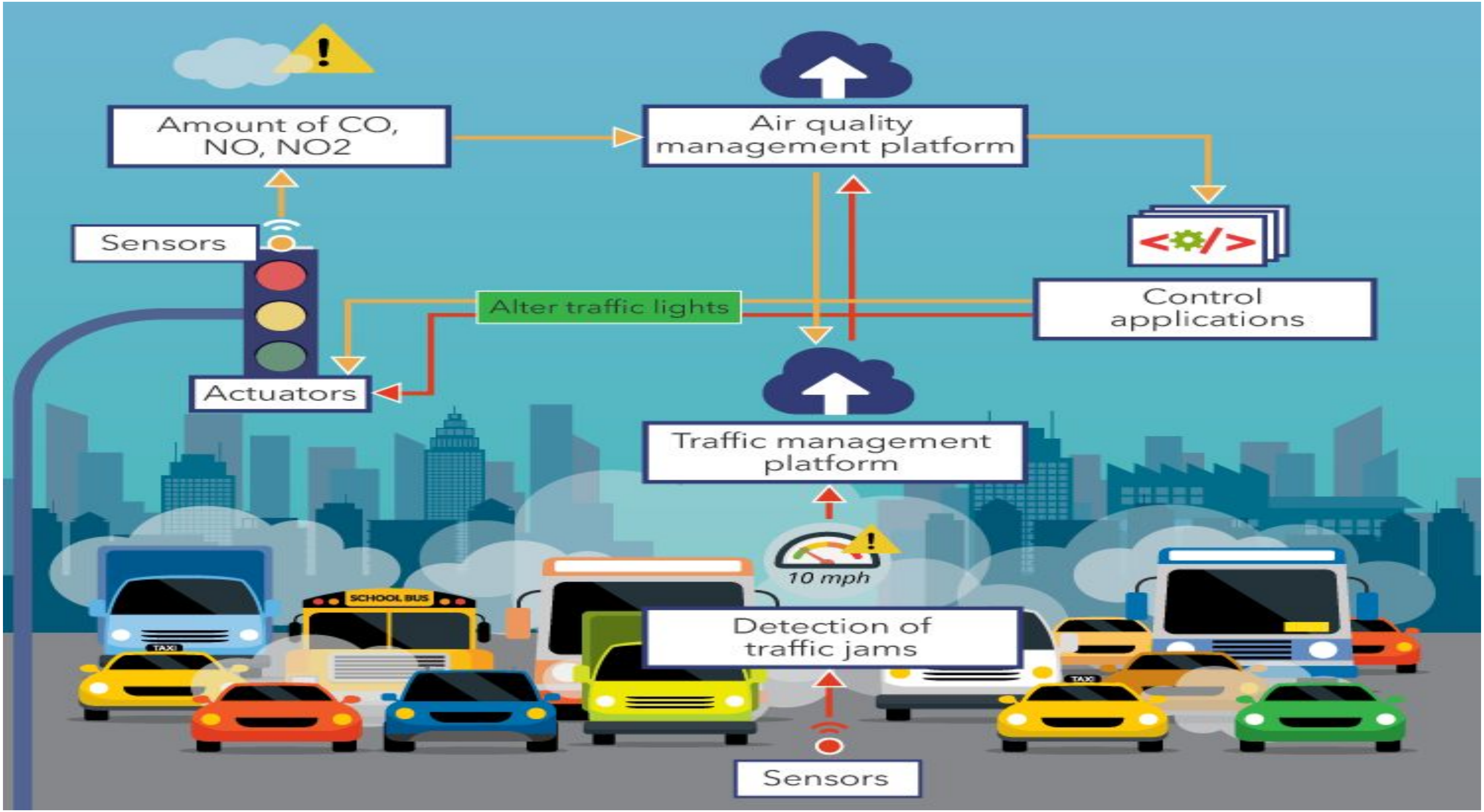
Stage 6: Integrating several solutions

Achieving “smartness” is not a one-time action – it is a continuous process. Implementing IoT-based smart city solutions today, municipalities should think of services they might like to implement tomorrow. It implies not only increasing the number of sensors but, more importantly, the number of functions. Let’s illustrate this functional scalability with the example of a smart city solution for traffic monitoring.

A city deploys a traffic management solution to detect traffic jams in real time and manage traffic lights to reduce traffic in the areas with intensive traffic. After some time, the city decides to ensure city traffic doesn’t harm the environment and integrates the traffic management solution with a smart air quality monitoring solution. Cross-solution integration allows controlling both traffic and air quality in the city dynamically.

For that, traffic lights or street lights along the roads can be equipped with sensors that monitor air quality. Sensors measure the amount of CO, NO, and NO2 in the air and pass data records to a central air quality management platform for processing. If the amount of harmful gases in the air is critical, control applications apply rules or use models to take an output action, e.g., ‘alter traffic lights.’ Before that, there is a need to make sure that altering traffic lights won’t cause accidents or blockages in other areas. It is possible due to the integration of the traffic management solution to the air quality management solution. The traffic management platform performs real-time analysis and identifies if it is possible to alter the traffic lights. If altering the lights is acceptable, control applications send a command to the traffic lights’ actuators, which execute the command.

Applying an iterative approach helps municipalities to reduce implementation costs, get a faster pay-off and make the benefits of smart solutions visible for citizens sooner.


Amount of CO, NO, NO2
Air quality management platform
Sensors
Control applications
Alter traffic lights
Actuators
Traffic management platform
10 mph
SCHOOL BUS
TAXI
Detection of traffic jams
Sensors

Adapting IoT implementation strategy to the city size

Iterative approach can be leveraged in cities of different sizes. In larger ones, it helps to deal with the scale and complexity of implementation; in smaller ones, it helps to reduce investments in smart solutions and use constrained infrastructure resources more reasonably. However, starting a smart project in a smaller city, municipalities have some more points to consider.

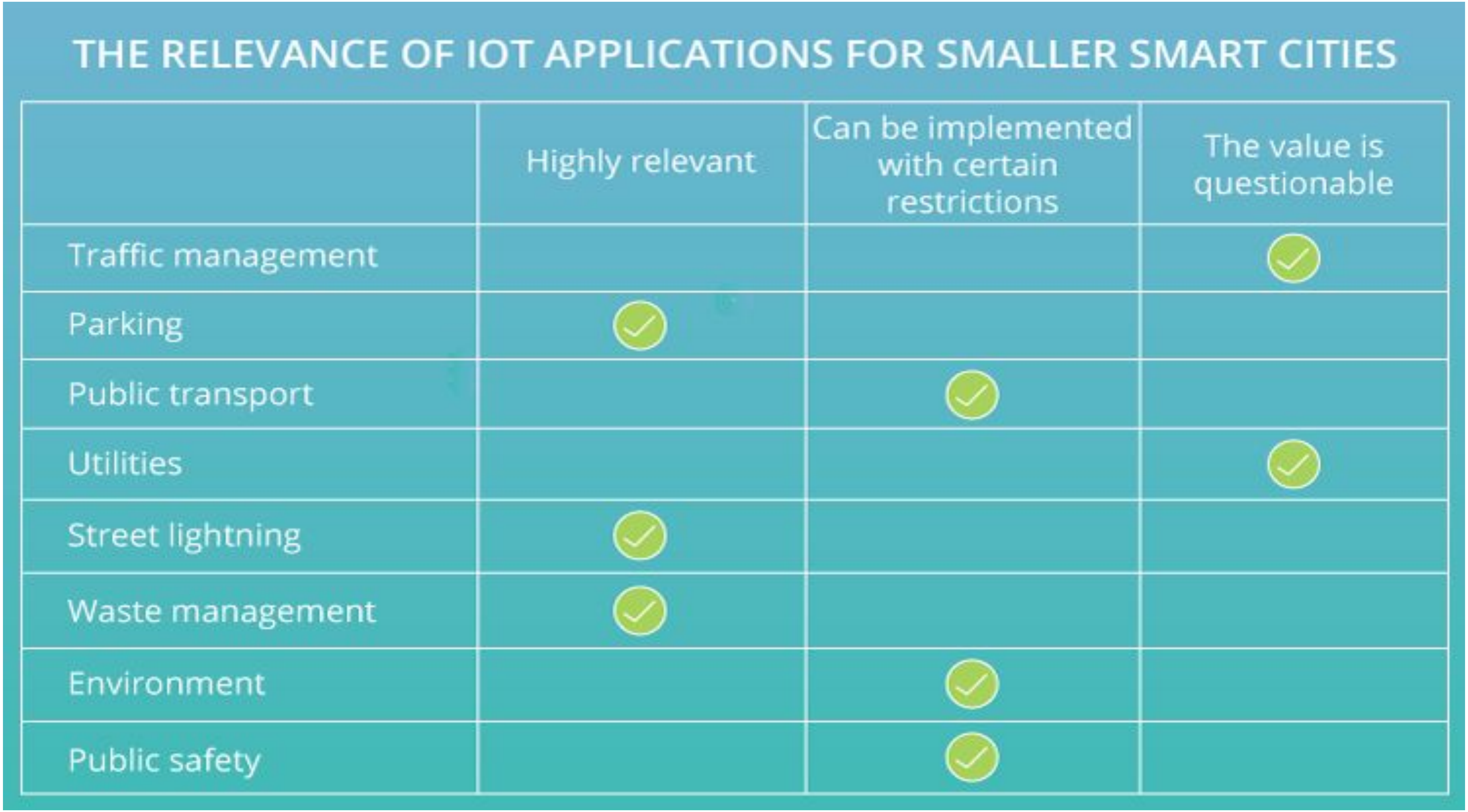
On the way to smartness, midsized and small cities face many barriers, including budgetary and procurement shortages, limited resources for public services, under-resourced IT infrastructure, etc. However, it doesn't mean a smaller city cannot be a smart city.

Starting a smart initiative in a city of medium or small size, it makes sense to begin with the projects that do not require huge investments and deliver tangible return on investments, such as smart parking or waste management, and use the established infrastructure to implement new services.

For example, the town of Vail, CO has less than 6,000 inhabitants but boasts an extensive smart infrastructure. The town started smart city development with connected streetlights. Later, they used the established infrastructure to broaden the range of services and topped it with smart parking and irrigation systems.

To determine which applications are a good fit for smaller cities, we've analyzed them by the volume of investments, required infrastructure, pay-off period, the visibility of benefits for citizens and came up with the following table:

Another non-trivial way to enhance the affordability and accessibility of smart applications is sharing a common platform with a larger city. The cloud nature of IoT-enabled smart city solutions is suitable for that. This way, smart city solutions of both large and smaller smart cities are connected to and managed via a single cloud platform. By sharing the platform based on open data, several smart cities form a common urban ecosystem. One of the examples of such sharing is the Iberian Smart Cities Network, which currently includes 111 cities in Portugal and Spain. The network comprises cities of different sizes, which cooperate in multiple areas including smart energy, mobility, environment, and transport.



THE RELEVANCE OF IOT APPLICATIONS FOR SMALLER SMART CITIES

	Highly relevant	Can be implemented with certain restrictions	The value is questionable
Traffic management			✓
Parking	✓		
Public transport		✓	
Utilities			✓
Street lightning	✓		
Waste management	✓		
Environment		✓	
Public safety		✓	

Conclusion:

IoT helps cities connect and manage multiple infrastructure and public services. From smart lighting and road traffic to connected public transport and waste management – the range of use cases is highly diverse. What they have in common is the outcomes. Applying IoT solutions leads to reduced costs for energy, optimized use of natural resources, safer cities, and a healthier environment.

However, to enjoy these benefits, municipalities should take a consistent approach to design a functional and scalable smart city architecture. Well-designed, it will allow to reduce investments in IoT development and hasten the implementation of smart city solutions, still leaving space for expansion.

dreamstime

Mile stone of 100% Syllabus

Prof.Yasmeen Attar

All the best

GOD BLESS YOU

Prof.Yasm