



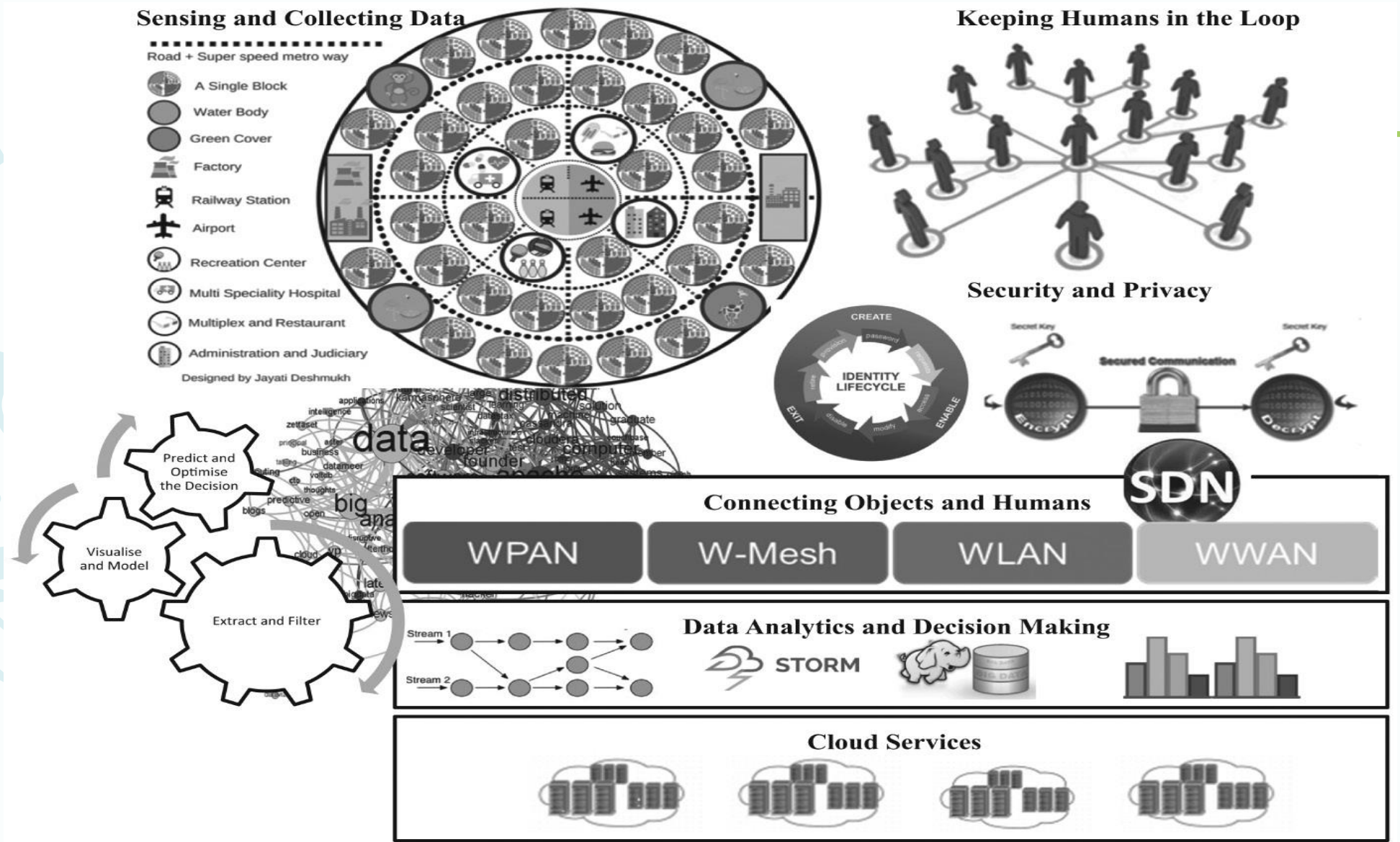
Overview of Internet of Things

PREPARED BY,
SHELLY SHIJU GEORGE
ASSISTANT PROFESSOR

Overview of Internet of Things

- Two important pillars – “Internet” and “Things”
- Every object capable of connecting to the Internet will be in “Things” category.
- Any entity able to communicate with other entities, making it accessible at anytime, anywhere.
- The objects are accessible without any time or place restrictions.
- Pervasive connectivity is a crucial requirement of IoT.
- IoT requires integration of mobile devices, edge devices like routers and smart hubs and humans in the loop as controllers.

IoT Ecosystem



IoT Definition

- Kevin Ashton is accredited for using the term “Internet of Things”.
- IoT is pervasive and autonomous networks of objects where identification and service integration have important and inevitable role.
- IIoT (Industrial IoT) – machines can perform specific tasks such as data acquisition and communication more accurately than humans and thus IIoT came into existence.
- Another characteristics of IoT is “smartness” which can be further categorized into “object smartness” and “network smartness”.
- IoT is enabler for machine-to-machine, human-to-machine and human-with-environment interactions.



IoT improving the Quality of Life

IoT is recognized by the impact on quality of life and businesses , which can revolutionize the way our medical systems and businesses operate by:

- (1) expanding the communication channel between objects by providing a more integrated communication environment in which different sensor data such as location, heartbeat, etc. can be measured and shared more easily.
- (2) Facilitating the automation and control process, whereby administrators can manage each object's status via remote consoles; and
- (3) saving in the overall cost of implementation, deployment, and maintenance, by providing detailed measurements and the ability to check the status of devices remotely.

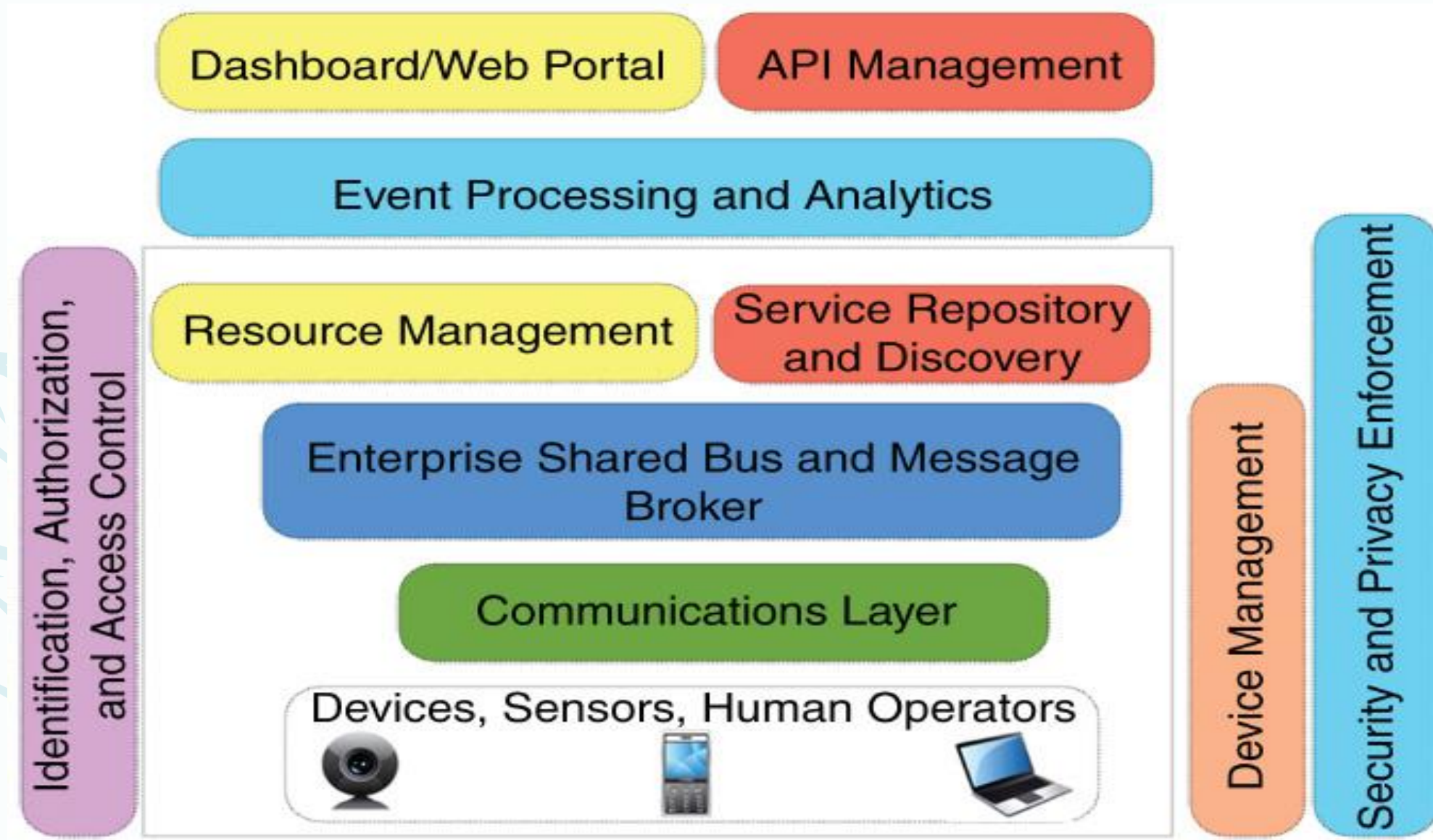


IoT Architectures

The building blocks of IoT are

- sensory devices,
- remote service invocation,
- communication networks,
- and context-aware processing of events

A Reference Architecture for IoT





Different service and presentation layers are shown in this architecture.

- Service layers include event processing and analytics, resource management and service discovery, as well as message aggregation and Enterprise Service Bus (ESB) services built on top of communication and physical layers.
- API management, which is essential for defining and sharing system services and web-based dashboards (or equivalent smartphone applications) for managing and accessing these APIs, are also included in the architecture.
- Due to the importance of device management, security and privacy enforcement in different layers, and the ability to uniquely identify objects and control their access level, these components are prestressed independently in this architecture.

SOA-BASED ARCHITECTURE



- Service-Oriented Architecture (SOA)
- SOA ensures the interoperability among the heterogeneous devices.



A generic SOA consisting of four layers, with distinguished functionalities as follows:

- Sensing layer is integrated with available hardware objects to sense the status of things
- Network layer is the infrastructure to support over wireless or wired connections among things
- Service layer is to create and manage services required by users or applications
- Interfaces layer consists of the interaction methods with users or applications

API-ORIENTED ARCHITECTURE



- Application Programming Interface (API)
- Building APIs for IoT applications helps the service provider attract more customers while focusing on the functionality of their products rather than on presentation.
- It also provides more efficient service monitoring and pricing tools than previous service-oriented approaches.

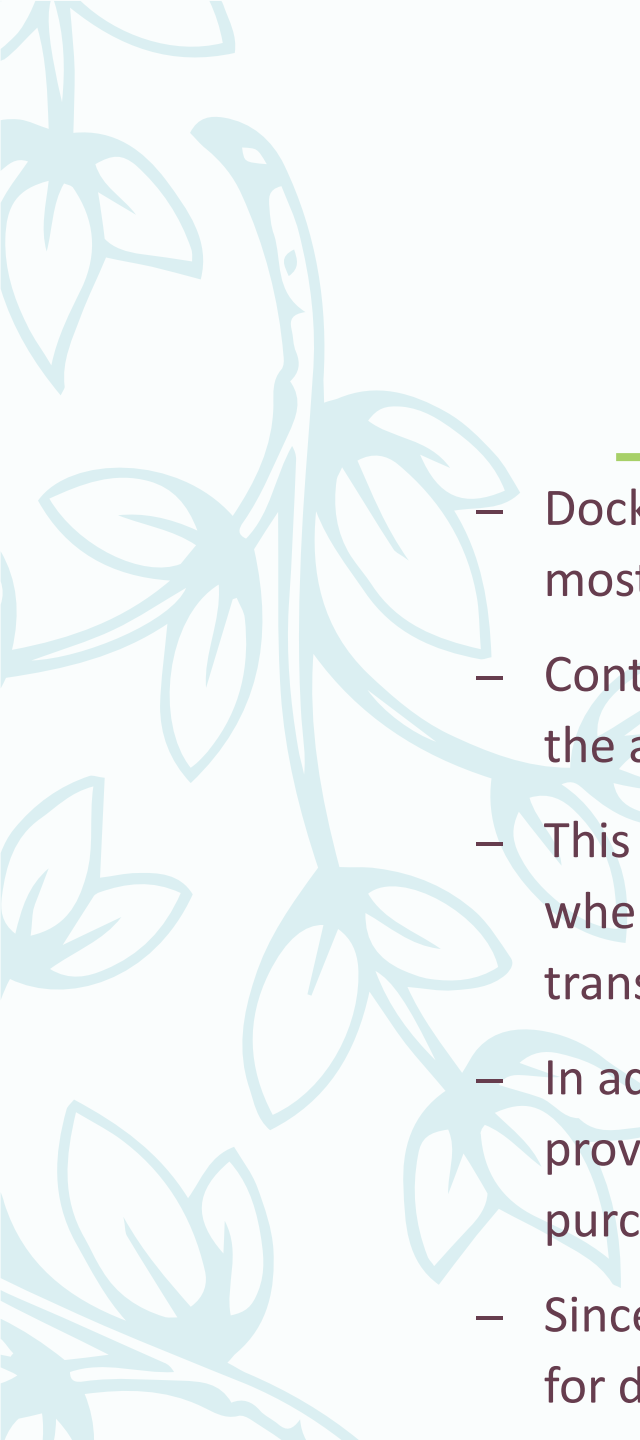


RESOURCE MANAGEMENT

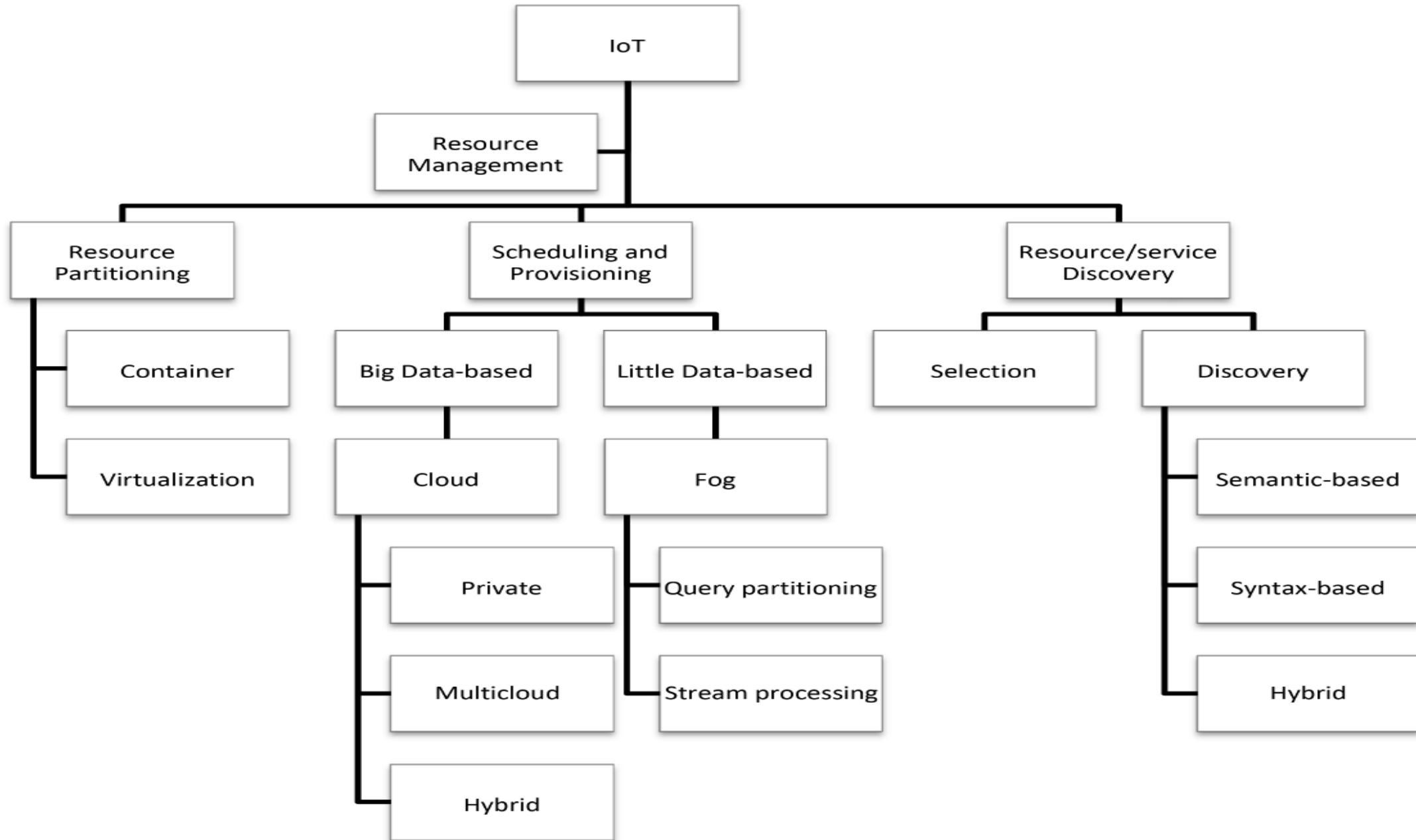
- Large-scale deployment of sensors for a smart city use-case, it is obvious that an efficient resource management module needs considerable robustness, fault-tolerance, scalability, energy efficiency, QoS, and SLA.
- Resource management involves discovering and identifying all available resources, partitioning them to maximize a utility function—which can be in terms of cost, energy, performance, etc., and, finally, scheduling the tasks on available physical resources.

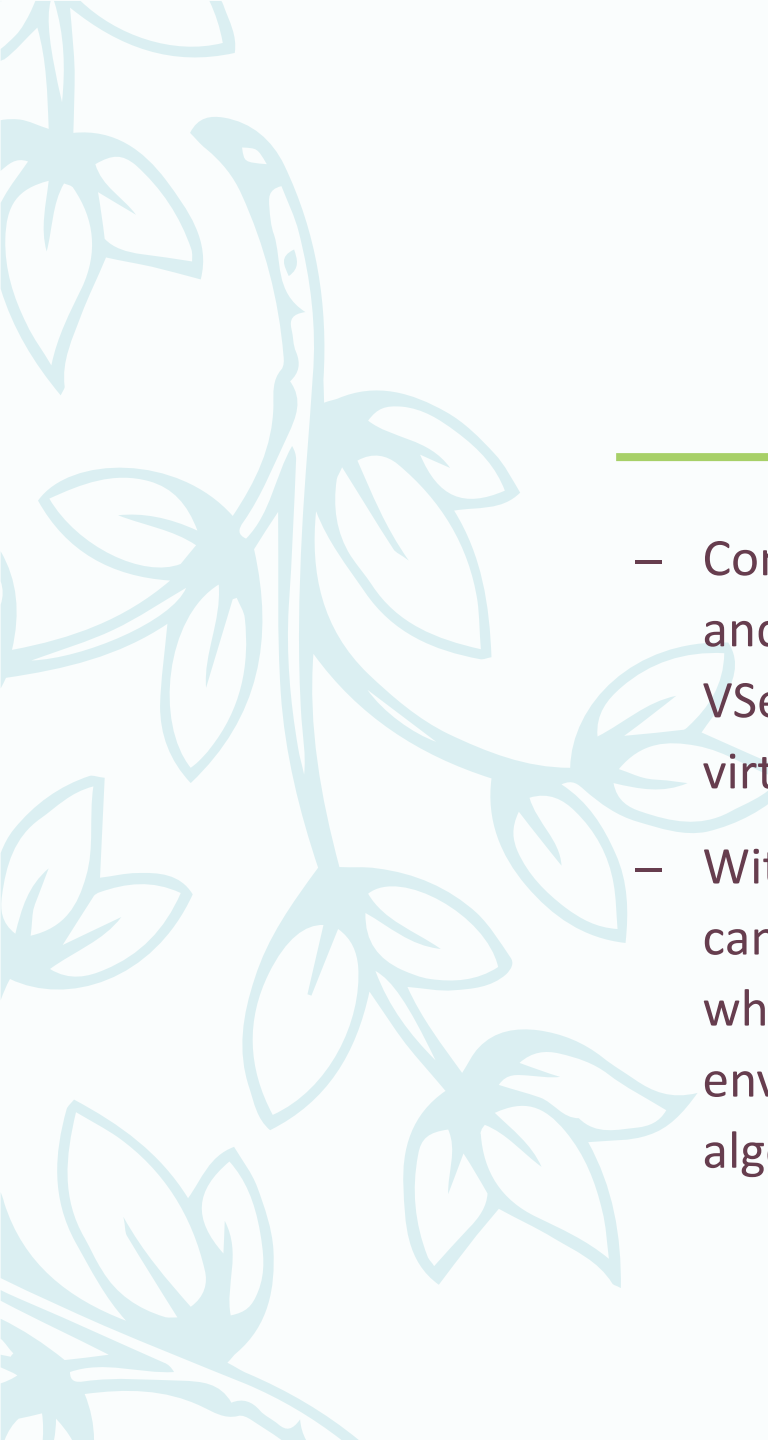
RESOURCE PARTITIONING

- The first step for satisfying resource provisioning requirements in IoT is to efficiently partition the resources and gain a higher utilization rate.
- The hypervisor, is responsible for managing interactions between host and guest VMs, requires a considerable amount of memory and computational capacity, this configuration is not suitable for IoT, where devices often have constrained memory and processing power.
- To address these challenges, the concept of **Containers** has emerged as a new form of virtualization technology that can match the demand of devices with limited resources.

- 
-
- Docker (<https://www.docker.com/>) and Rocket (<https://github.com/coreos/rkt>) are the two most famous container solutions.
 - Containers are able to provide portable and platform-independent environments for hosting the applications and all their dependencies, configurations, and input/output settings.
 - This significantly reduces the burden of handling different platform-specific requirements when designing and developing applications, hence providing a convenient level of transparency for applications, architects, and developers.
 - In addition, containers are lightweight virtualization solutions that enable infrastructure providers to efficiently utilize their hardware resources by eliminating the need for purchasing expensive hardware and virtualization software packages.
 - Since containers, compared to VMs, require considerably less spin-up time, they are ideal for distributed applications in IoT that need to scale up within a short amount of time.

Taxonomy of Resource Management in IoT



- 
-
- Container-based virtualization, can bring advantages in terms of performance and security by sandboxing applications on top of a shared OS layer. Linux VServer, Linux Containers LXC, and OpenVZ are examples of using OS virtualization in an embedded systems domain.
 - With respect to heterogeneity of devices in IoT, and the fact that many of them can leverage virtualization to boost their utilization rate, task-grain scheduling, which considers individual tasks within different containers and virtualized environments, can potentially challenge current resource-management algorithms that view these layers as black box.




COMPUTATION OFFLOADING

- Code offloading (computation offloading) is another solution for addressing the limitation of available resources in mobile and smart devices.
- The advantages of using code offloading translate to more efficient power management, fewer storage requirements, and higher application performance.
- The proposed combination of VMs and mobile clouds can create a powerful environment for sharing, synchronizing, and executing codes in different platforms.



IDENTIFICATION AND RESOURCE/SERVICE DISCOVERY

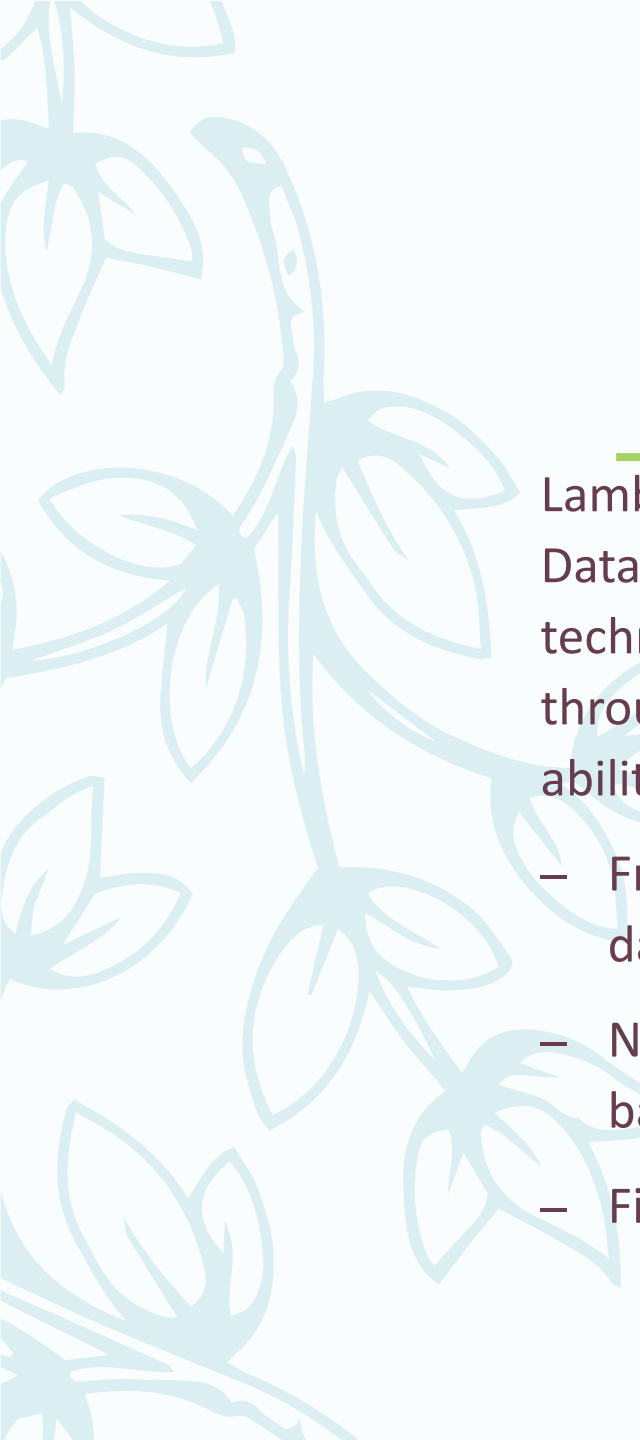
- The discovery module in IoT is twofold.
- The first objective is to identify and locate the actual device, which can be achieved by storing and indexing metadata information about each object.
- The final step is to discover the target service that needs to be invoked.
- Lack of an effective discovery algorithm can result in execution delays, poor user experience, and runtime failures.
- Efficient algorithms that dynamically choose centralized or flooding strategies can help minimize the consumed energy, although other parameters such as mobility and latency should be factored in to offer a suitable solution for IoT, considering its dynamic nature.

- 
-
- In another approach within the fog-computing context, available resources like network bandwidth and computational and storage-capacity metrics are converted to time resources, forming a framework that facilitates resource sharing.
 - Different parameters like energy-consumption level, price, and availability of services need to be included in proposing solutions that aim to optimize resource sharing within a heterogeneous pool of resources.



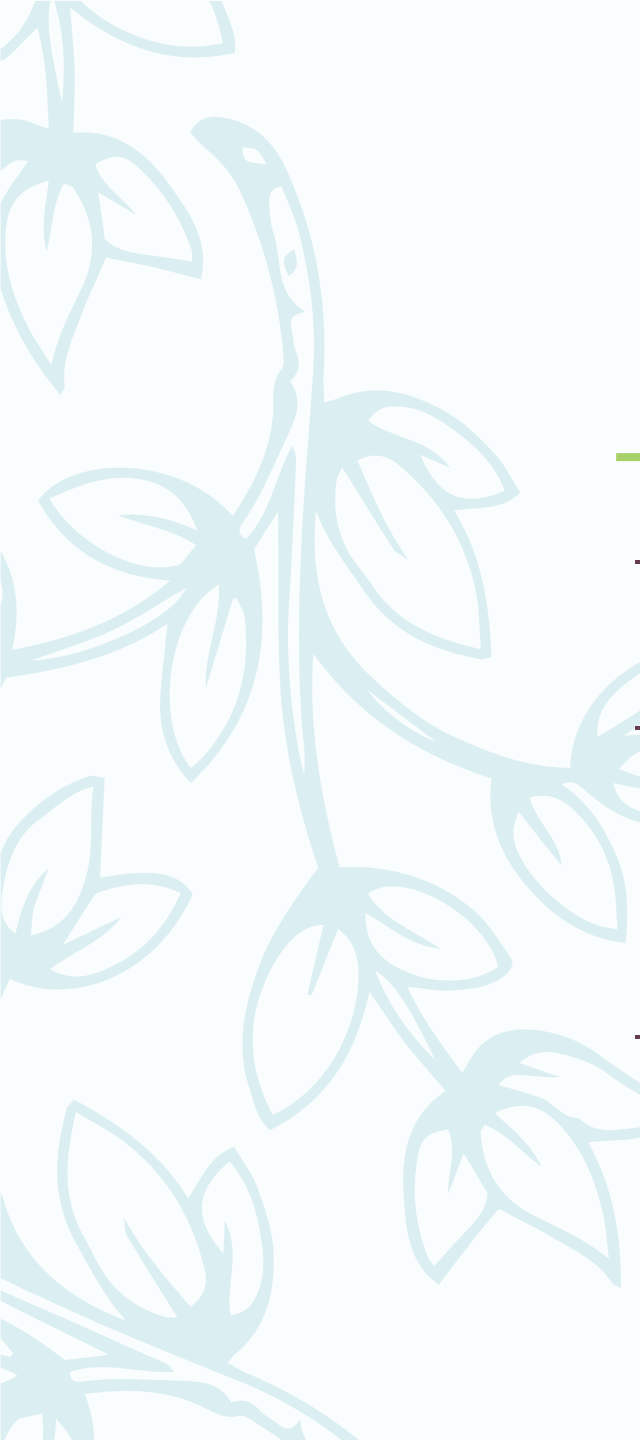
IoT DATA MANAGEMENT AND ANALYTICS

- Special considerations are required to process huge amounts of data originating from, and circulating in, such a distributed and heterogeneous environment.
- Big Data related procedures, such as data acquisition, filtering, transmission, and analysis have to be updated to match the requirements of the IoT data.
- Big Data is characterized by 3Vs, namely velocity, volume, and variety.
- **Batch Processing and Stream Processing** are two major methods used for data analysis.



Lambda Architecture is an exemplary framework proposed by Nathan Marz to handle Big Data processing by focusing on multiapplication support, rather than on data-processing techniques. It has three main layers that enable the framework to support easy extensibility through extension points, scale-out capabilities, low-latency query processing, and the ability to tolerate human and system faults.

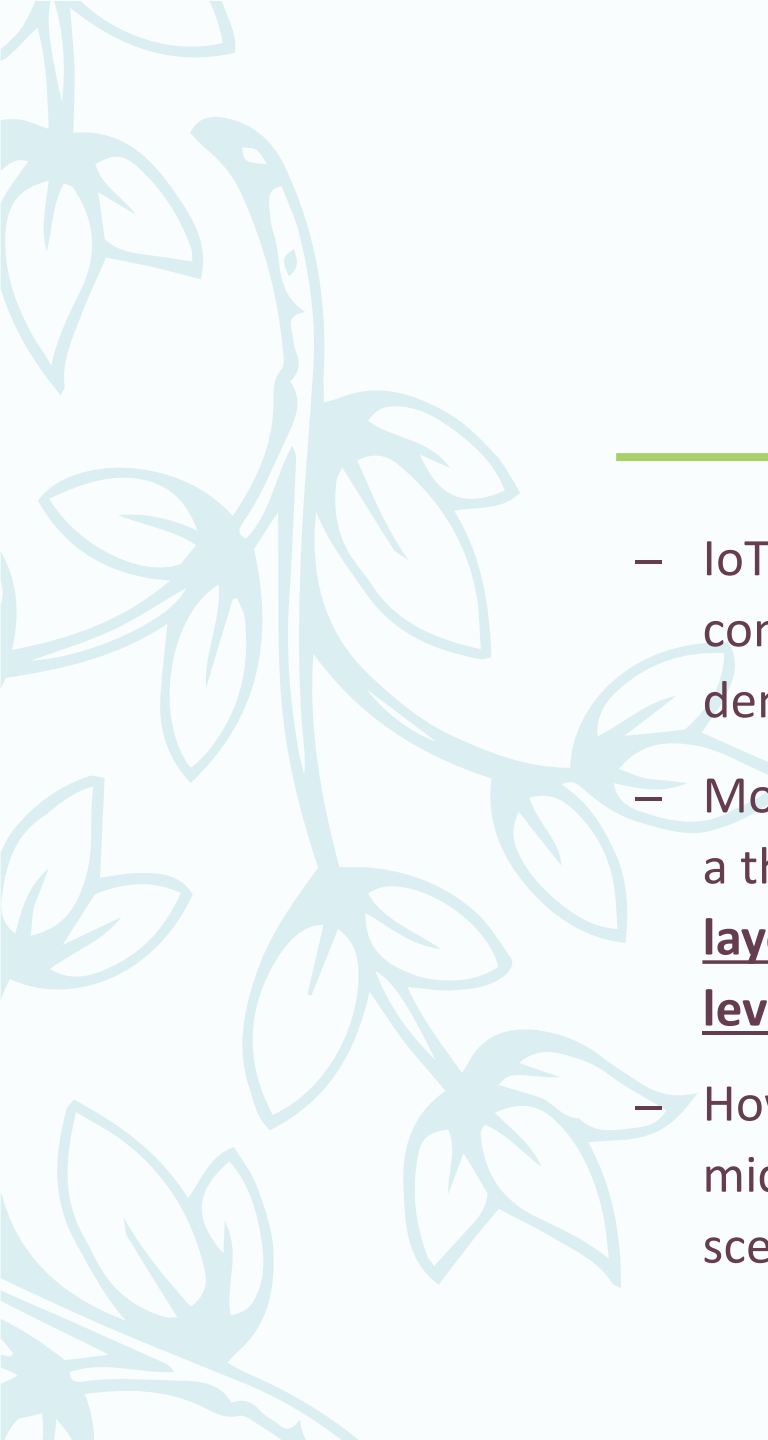
- From a top-down view, the first layer is called “**Batch Layer**” and hosts the master dataset and batch views where precomputed queries are stored.
- Next is the “**Serving Layer**,” which adds dynamic query creation and execution to the batch views by indexing and storing them.
- Finally, the “**Speed Layer**” captures and processes recent data for delay-sensitive queries.


- 
-
- To this end, applications utilize pattern detection and data-mining techniques to extract knowledge and make smarter decisions.
 - One of the key limitations in using currently developed datamining algorithms lies in the inherent centralized nature of these algorithms, which drastically affects their performance and makes them unsuitable for IoT environments that are meant to be geographically distributed and heterogeneous.
 - Distributed anomaly-detection techniques that concurrently process multiple streams of data to detect outliers have been well-studied.



IoT AND THE CLOUD

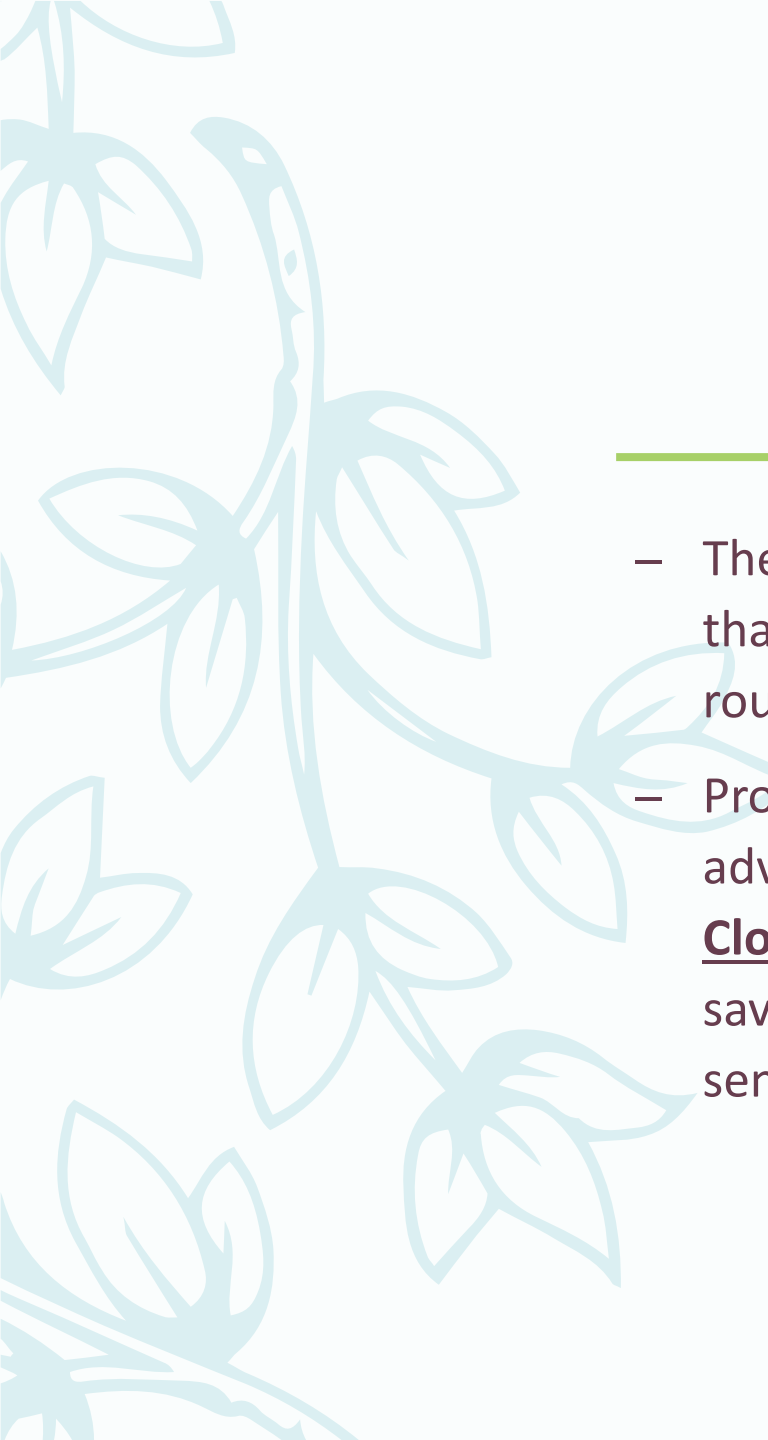
- Cloud computing, due to its on-demand processing and storage capabilities, can be used to analyze data generated by IoT objects in batch or stream format.
- A pay-as-you-go model adopted by all cloud providers has reduced the price of computing, data storage, and data analysis, creating a streamlined process for building IoT applications.
- With cloud's elasticity, distributed Stream Processing Engines (SPEs) can implement important features such as fault-tolerance and autoscaling for bursty workloads.

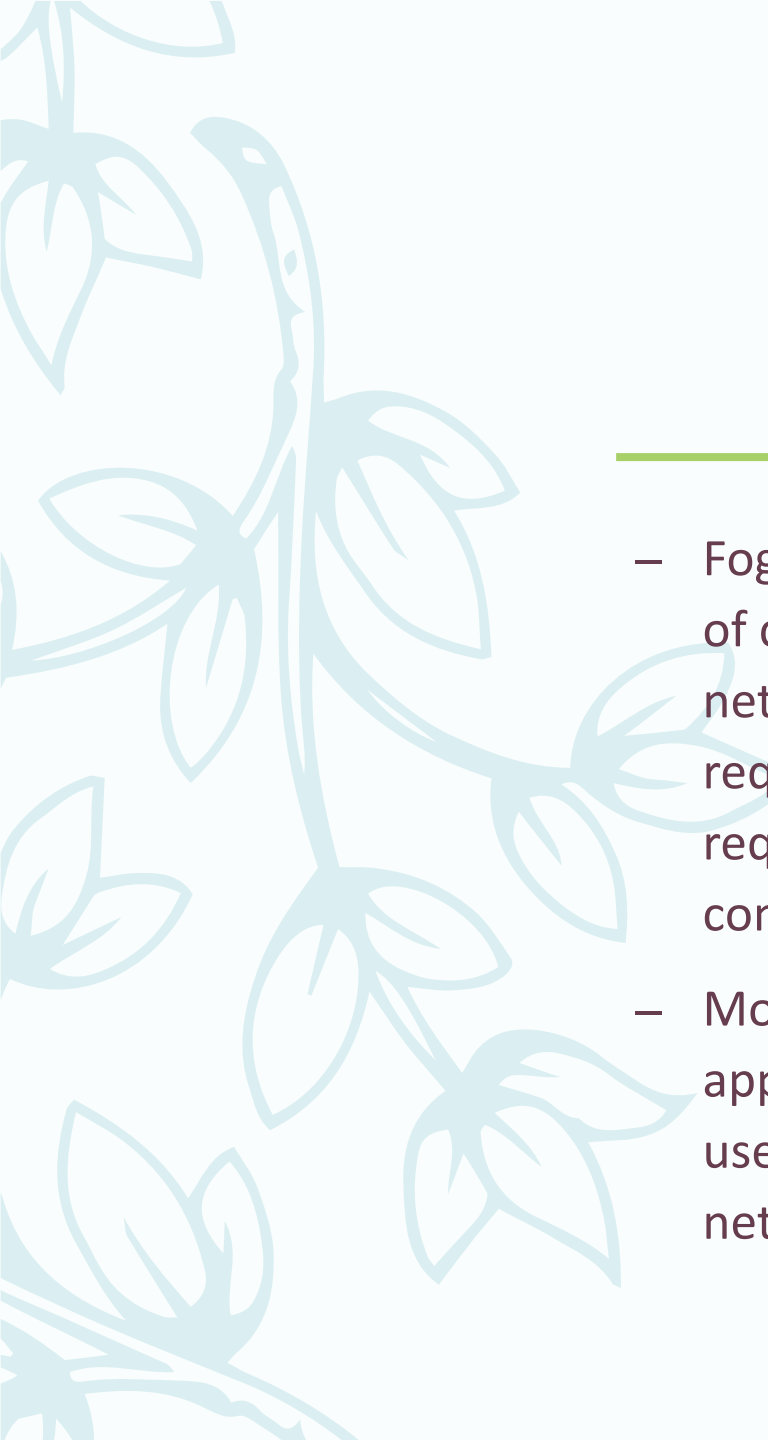
- 
-
- IoT applications can harness cloud services and use the available storage and computing resources to meet their scalability and compute-intensive processing demands.
 - Most of the current design approaches for integrating cloud with IoT are based on a three-tier architecture, where **the bottom layer consists of IoT devices, middle layer is the cloud provider, and top layer hosts different applications and high-level protocols.**
 - However, using this approach to design and integrate cloud computing with an IoT middleware limits the practicality and full utilization of cloud computing in scenarios where minimizing end-to-end delay is the goal.




REAL-TIME ANALYTICS IN IoT AND FOG COMPUTING

- Current data-analytics approaches mainly focus on dealing with Big Data, however, processing data generated from millions of sensors and devices in real time is more challenging.
- Proposed solutions that only utilize cloud computing as a processing or storage backbone are not scalable and cannot address the latency constraints of real-time applications.
- Real-time processing requirements and the increase in computational power of edge devices such as routers, switches, and access points lead to the emergence of the Edge Computing paradigm.

- 
-
- The Edge layer contains the devices that are in closer vicinity to the end user than the application servers, and can include smartphones, smart TVs, network routers, and so forth.
 - Processing and storage capability of these devices can be utilized to extend the advantages of using cloud computing by creating another cloud, known as **Edge Cloud**, near application consumers, in order to: decrease networking delays, save processing or storage cost, perform data aggregation, and prevent sensitive data from leaving the local network

- 
-
- Fog Computing is a term coined by Salvatore Stolfo and applies to an extension of cloud computing that aims to keep the same features of Cloud, such as networking, computation, virtualization, and storage, but also meets the requirements of applications that demand low latency, specific QoS requirements, Service Level Agreement (SLA) considerations, or any combination of these .
 - Moreover, these extensions can ease application development for mobile applications, Geo-distributed applications such as WSN, and large-scale systems used for monitoring and controlling other systems, such as surveillance camera networks.



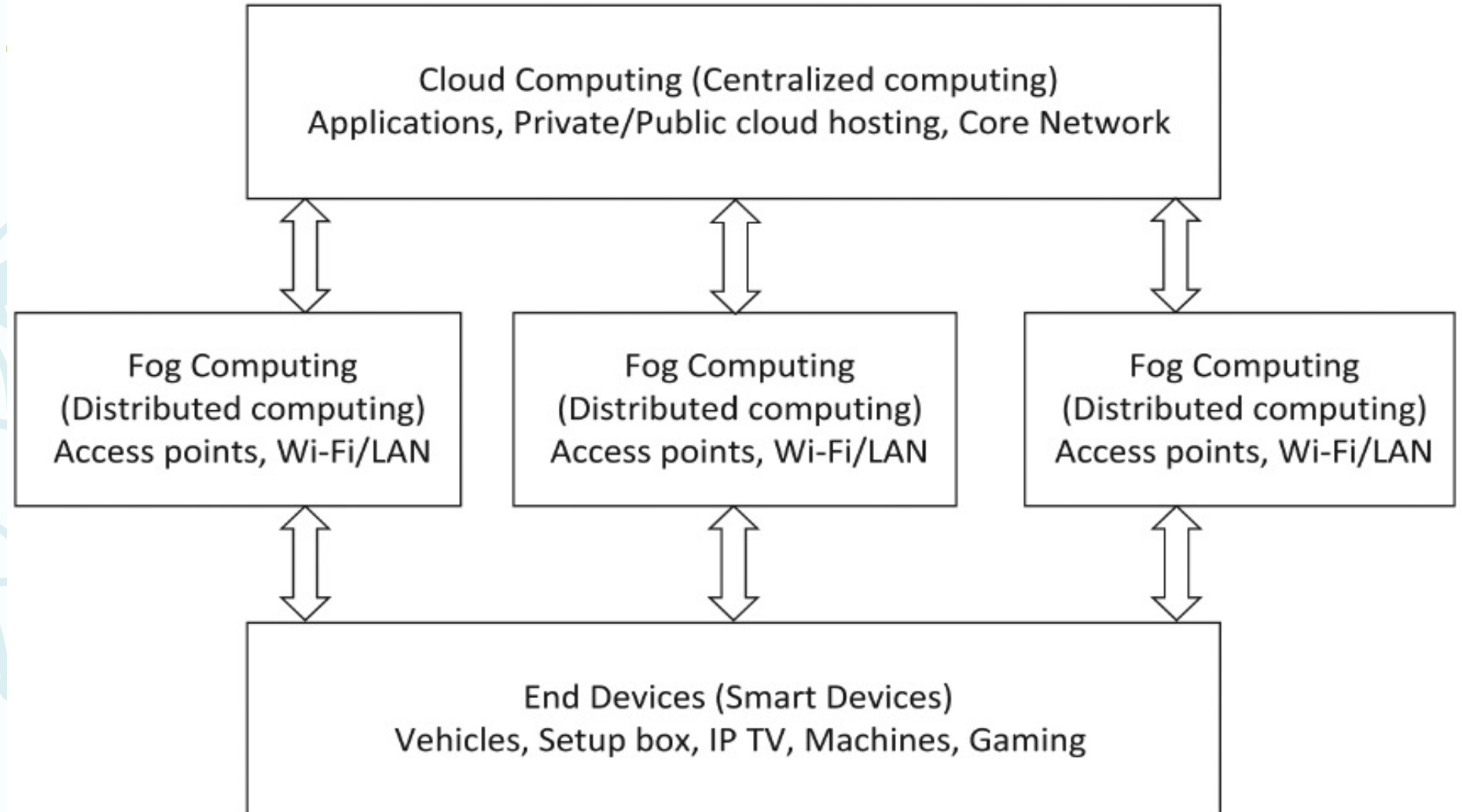
The following requirements should be fulfilled in an efficient real-time Stream Processing Engine (SPE):

- Data fluidity, which refers to processing data on-the-fly without the need for costly data storage
- Handling out-of-order, missing, and delayed streams
- Having a repeatable and deterministic outcome after processing a series or bag of streams
- Keeping streaming and stored data integrated by using embedded database systems
- Assuring high availability, using real-time failover and hot backup mechanisms
- Supporting autoscaling and application partitioning

Cloud Versus Fog

	Fog	Cloud
Response Time	Low	High
Availability	Low	High
Security Level	Medium to hard	Easy to medium
Service Focus	Edge devices	Network/enterprise core services
Cost for each device	Low	High
Dominant architecture	Distributed	Central/distributed
Main content generator - consumer	Smart devices- humans and devices	Humans-end devices

Typical Fog Computing Architecture





COMMUNICATION PROTOCOLS

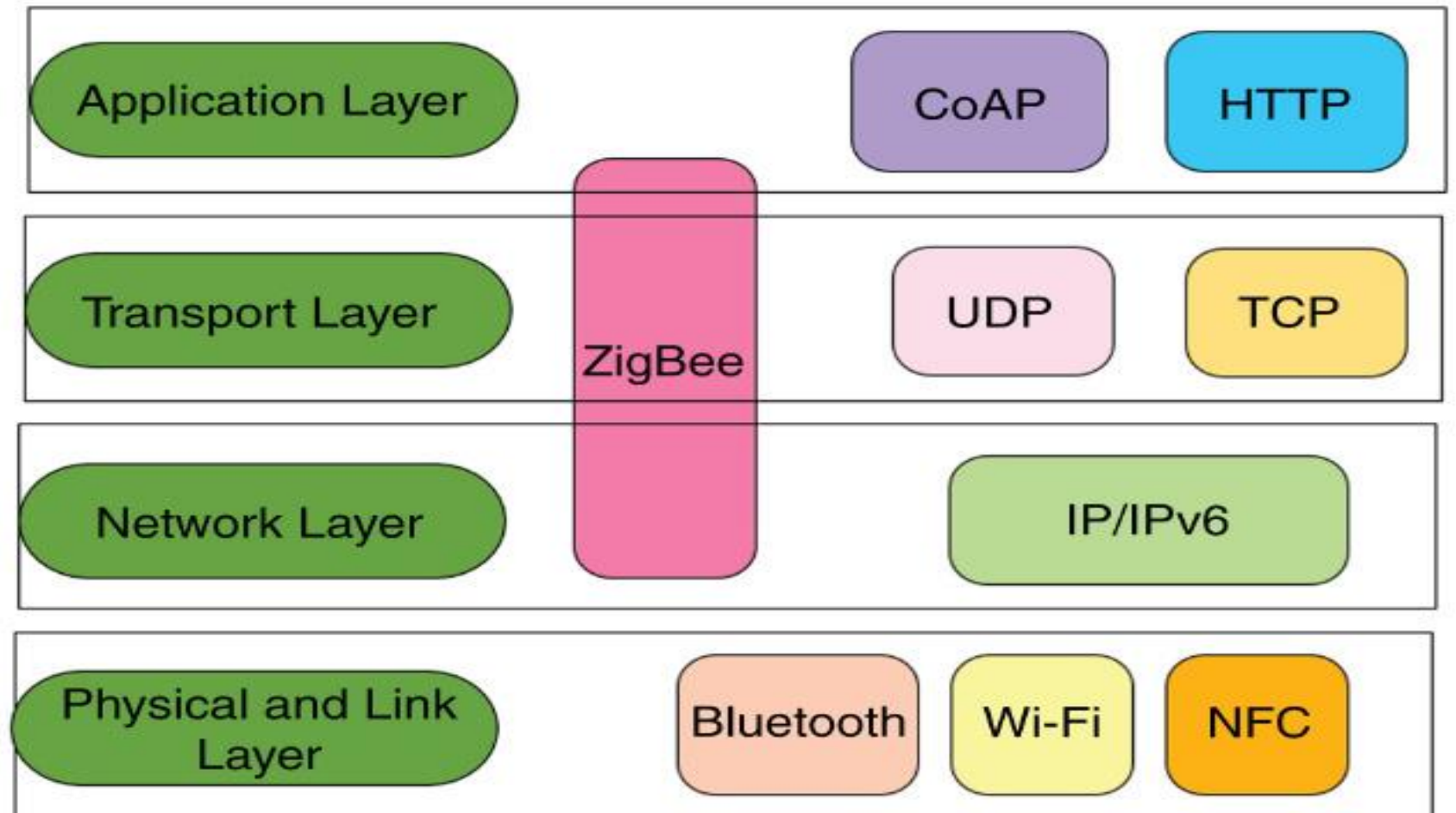
- From the network and communication perspective, IoT can be viewed as an aggregation of different networks, including mobile networks (3G, 4G, CDMA, etc.), WLANs, WSN, and Mobile Adhoc Networks (MANET).
- Seamless connectivity is a key requirement for IoT. Network-communication speed, reliability, and connection durability will impact the overall IoT experience.
- With the emergence of high-speed mobile networks like 5G, and the higher availability of local and urban network communication protocols such as Wi-Fi, Bluetooth, and WiMax, creating an interconnected network of objects seems feasible, however, dealing with different communication protocols that link these environments is still challenging.

NETWORK LAYER

Based on the device's specification (memory, CPU, storage, battery life), the communication means and protocols vary. However, the commonly used communication protocols and standards are listed below:

- RFID (eg, ISO 18000 series that comes with five classes and two generations, and covers both active and passive RFID tags)
- IEEE 802.11 (WLAN), IEEE 802.15.4 (ZigBee), Near Field Communication (NFC), IEEE 802.15.1 (Bluetooth)
- Low-power Wireless Personal Area Networks (6LoWPAN) standards by IETF
- M2M protocols such as MQTT and CoAP
- IP layer technologies, such as IPv4, IPv6, etc.

Use of Various Protocols in IoT Communication Layers





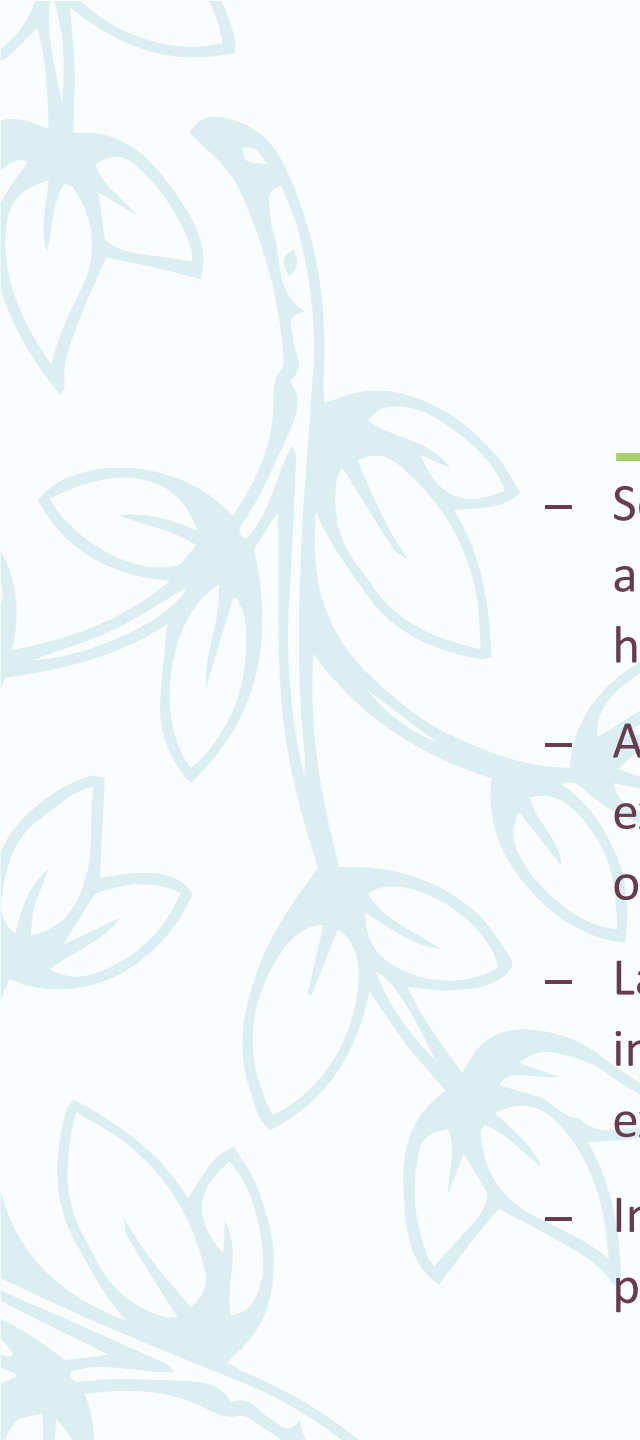
TRANSPORT AND APPLICATION LAYER

- Segmentation and poor coherency level, which are results of pushes from individual companies to maximize their market share and revenue, has made developing IoT applications cumbersome.
- Universal applications that require one-time coding and can be executed on multiple devices are the most efficient.




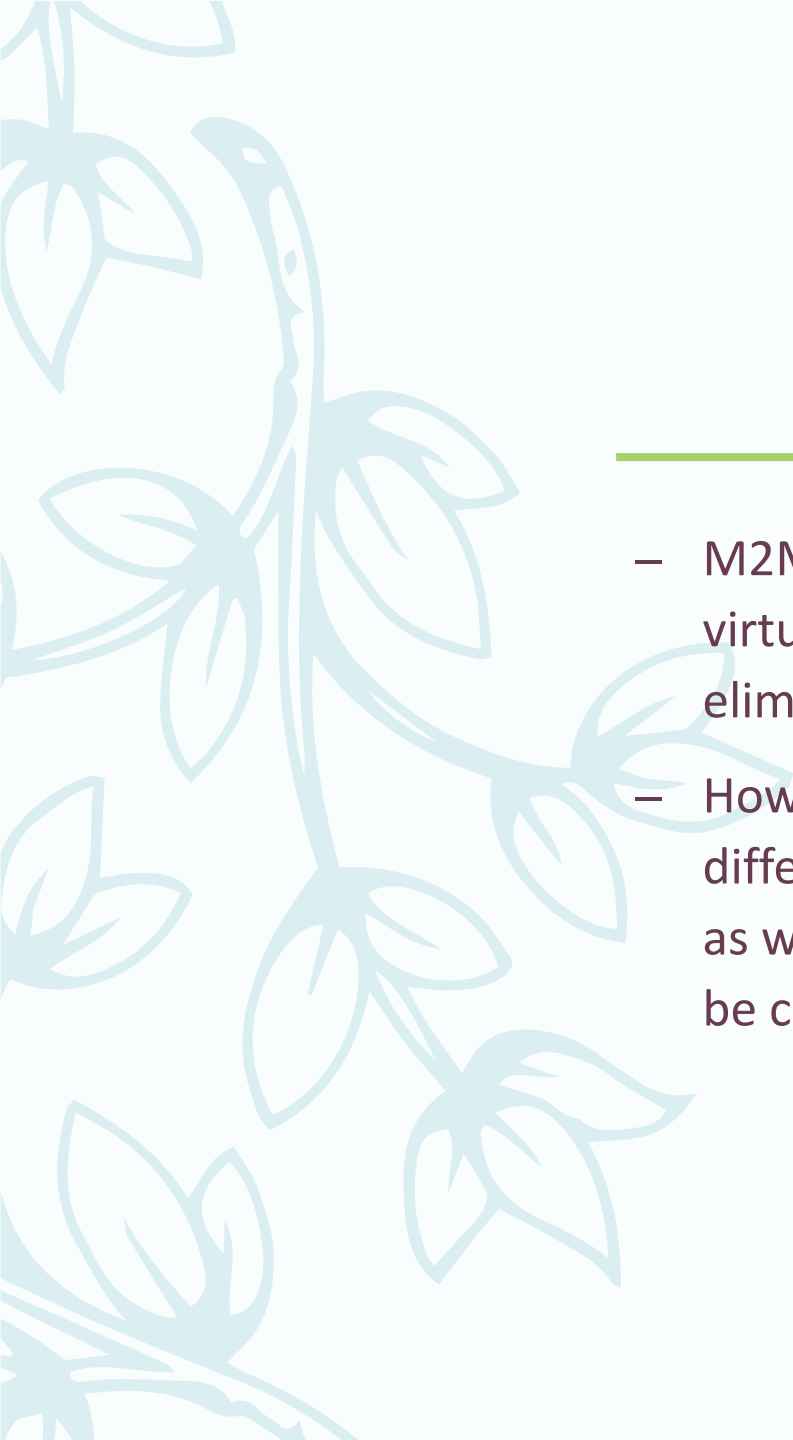
Protocols in IoT can be classified into three categories:

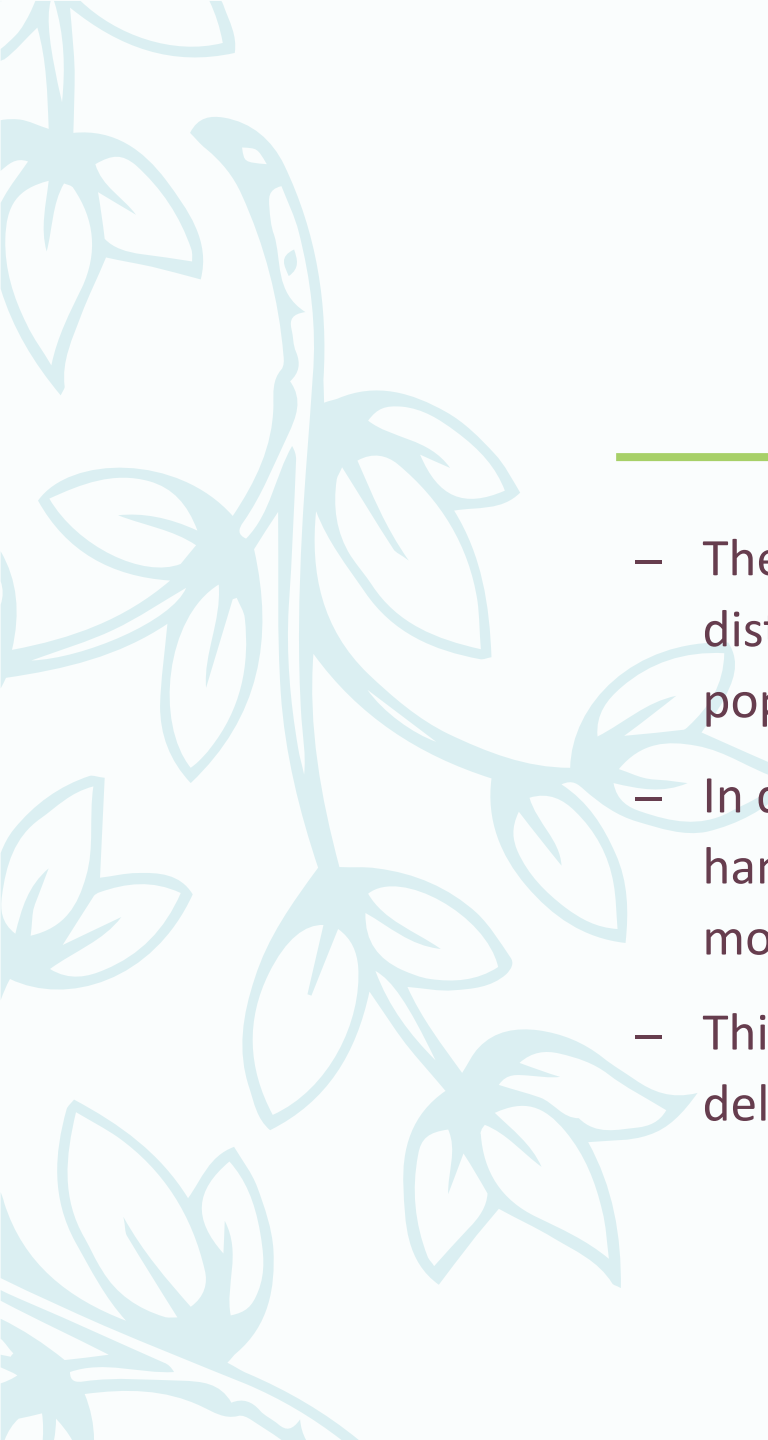
1. **general-purpose protocols** like IP and SNMP (Simple Network Management Protocol) that have been around for many years and are vastly used to manage, monitor, configure network devices, and establish communication links;
2. **lightweight protocols** such as CoAP (Constrained Application Protocol) that have been developed to meet the requirements of constrained devices with tiny hardware and limited resources;
3. **device- or vendor-specific protocols** and APIs that usually require a certain build environment and toolset.

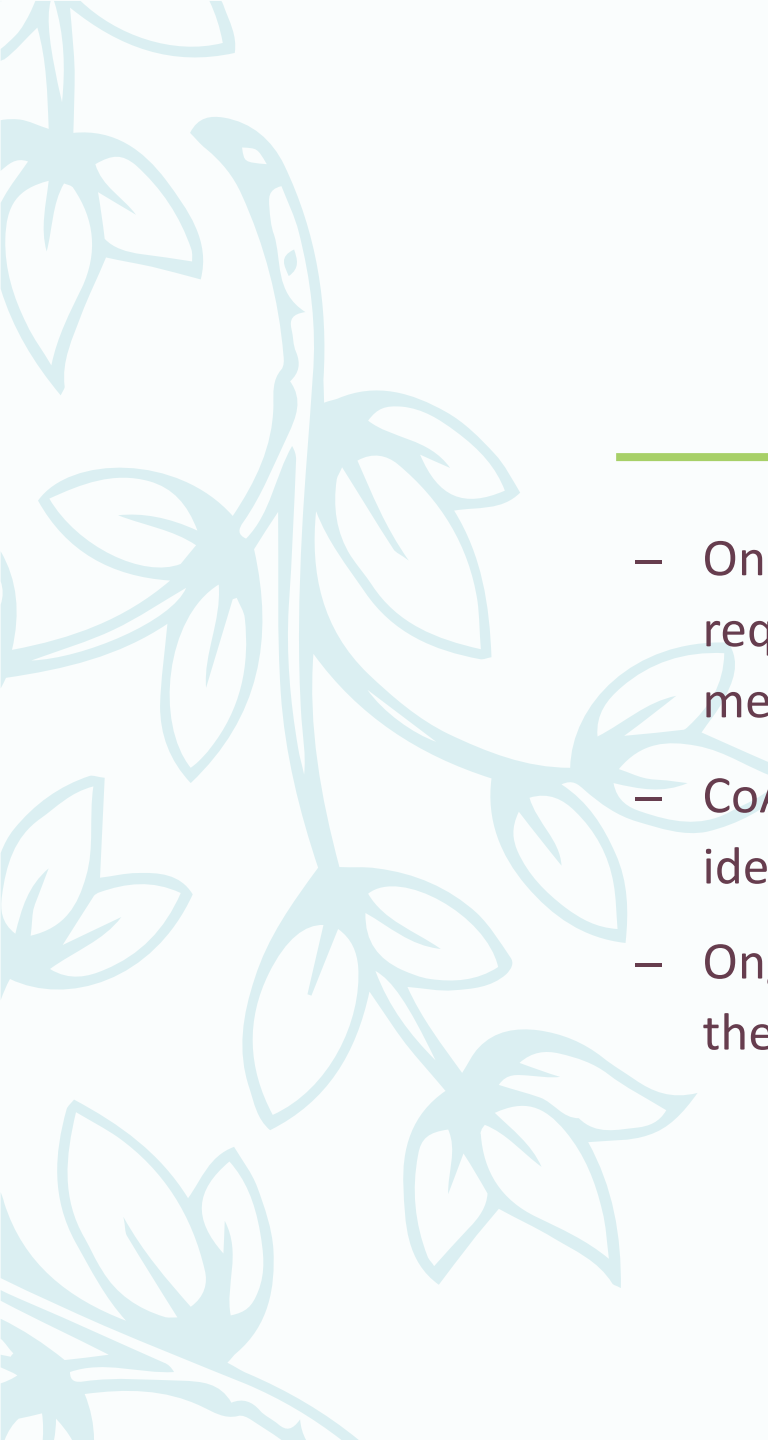
- 
-
- Selecting the right protocols at the development phase can be challenging and complex, as factors such as future support, ease of implementation, and universal accessibility have to be considered.
 - Additionally, thinking of other aspects that will affect the final deployment and execution, like required level of security and performance, will add to the sophistication of the protocol-selection stage.
 - Lack of standardization for particular applications and protocols is another factor that increases the risk of poor protocol selection and strategic mistakes that are more expensive to fix in the future.
 - In order to enhance their adoption, it is important to make sure that communication protocols are well documented; sensors and smart devices limit their usage in IoT.

IoT Communication Protocols Comparison					
Protocol Name	Transport Protocol	Messaging Model	Security	Best-Use Cases	Architecture
AMQP	TCP	Publish/Subscribe	High-Optional	Enterprise integration	P2P
CoAP	UDP	Request/Response	Medium-Optional	Utility field	Tree
DDS	UDP	Publish/Subscribe and Request/Response	High-Optional	Military	Bus
MQTT	TCP	Publish/Subscribe and Request/Response	Medium-Optional	IoT messaging	Tree
UPnP	---	Publish/Subscribe and Request/Response	None	Consumer	P2P
XMPP	TCP	Publish/Subscribe and Request/Response	High-Compulsory	Remote management	Client Server
ZeroMQ	UDP	Publish/Subscribe and Request/Response	High-Optional	CERN	P2P

- 
-
- AMPQ - Advanced Message Queuing Protocol
 - CoAP - Constrained Application Protocol
 - DDS - Data Distribution Service
 - MQTT - MQ Telemetry Transport
 - UpnP - Universal Plug and Play
 - XMPP - Extensible Messaging and Presence Protocol
 - ZeroMQ - ØMQ, 0MQ or ZMQ, ZeroMQ Message Transport Protocol (ZMTP)

- 
-
- M2M communication aims to enable seamless integration of physical and virtual objects into larger and geographically distributed enterprises by eliminating the need for human intervention.
 - However, to achieve this, the enforcement of harmony and collaboration among different communication layers (physical, transport, presentation, application), as well as the approaches used by devices for message storage and passing, can be challenging.

- 
-
- The publish/subscribe model is a common way of exchanging messages in distributed environments, and, because of simplicity, it has been adopted by popular M2M communication protocols like MQTT.
 - In dynamic scenarios, where nodes join or leave the network frequently and handoffs are required to keep the connections alive, the publish/subscribe model is efficient.
 - This is because of using push-based notifications and maintaining queues for delayed delivery of messages.

- 
-
- On the other hand, protocols like HTTP/REST and CoAP only support the request/response model, in which a pulling mechanism is used to fetch new messages from the queue.
 - CoAP also uses IPv6 and 6LoWPAN protocols in its network layer to handle node identification.
 - Ongoing efforts are still being made to merge these protocols and standardize them, as to support both publish/subscribe and request/response models.



INTERNET OF THINGS APPLICATIONS

- Industry-focused applications include logistics and transportation, supply-chain management , fleet management, aviation industry, and enterprise automation systems.
- Healthcare systems, smart cities and buildings, social IoT, and smart shopping are a few examples of applications that try to improve the daily life of individuals, as well as the whole society.
- Disaster management, environmental monitoring, smart watering, and optimizing energy consumption through smart grids and smart metering are examples of applications that focus on environment.
- In a broader magnitude, 54 different IoT applications under the following categories: smart environment, smart cities, smart metering, smart water, security and emergencies, retail, logistics, industrial control, smart agriculture, smart animal farming, domestic and home automation, and eHealth.

MONITORING AND ACTUATING

- Monitoring devices via APIs can be helpful in multiple domains. The APIs can report power usage, equipment performance, and sensor status, and they can perform actions upon sending predefined commands.
- Real-time applications can utilize these features to report current system status, whereas managers and developers have the option to freely call these APIs without the need for physically accessing the devices.
- Smart metering, and in a more distributed form, smart grids, can help in identifying production or performance defects via application of anomaly detection on the collected data, and thus increase the productivity.



BUSINESS PROCESS AND DATA ANALYSIS

The level of IoT adoption through Big Data analytics usage to the following categories:

- ***Society level***, where IoT mainly influences and improves government services by reducing cost and increasing government transparency and accountability
- ***Industry level***, in which manufacturing, emergency services, retailing, and education have been studied as examples
- ***Organizational level***, in which IoT can bring the same type of benefits as those mentioned in society level
- ***Individual level***, where daily life improvements, individual efficiency, and productivity growth are marked as IoT benefits



INFORMATION GATHERING AND COLLABORATIVE CONSUMPTION

- Social Internet of Things (SIoT) is where IoT meets social networks, and, to be more precise, it promises to link objects around us with our social media and daily interaction with other people, making them look smarter and more intractable.
- SIoT concept, motivated by famous social media like Facebook and Twitter, has the potential to affect many people's lifestyles.
- Another advantage is using the humans and their relationships, communities, and interactions for effective discovery of IoT services and objects.




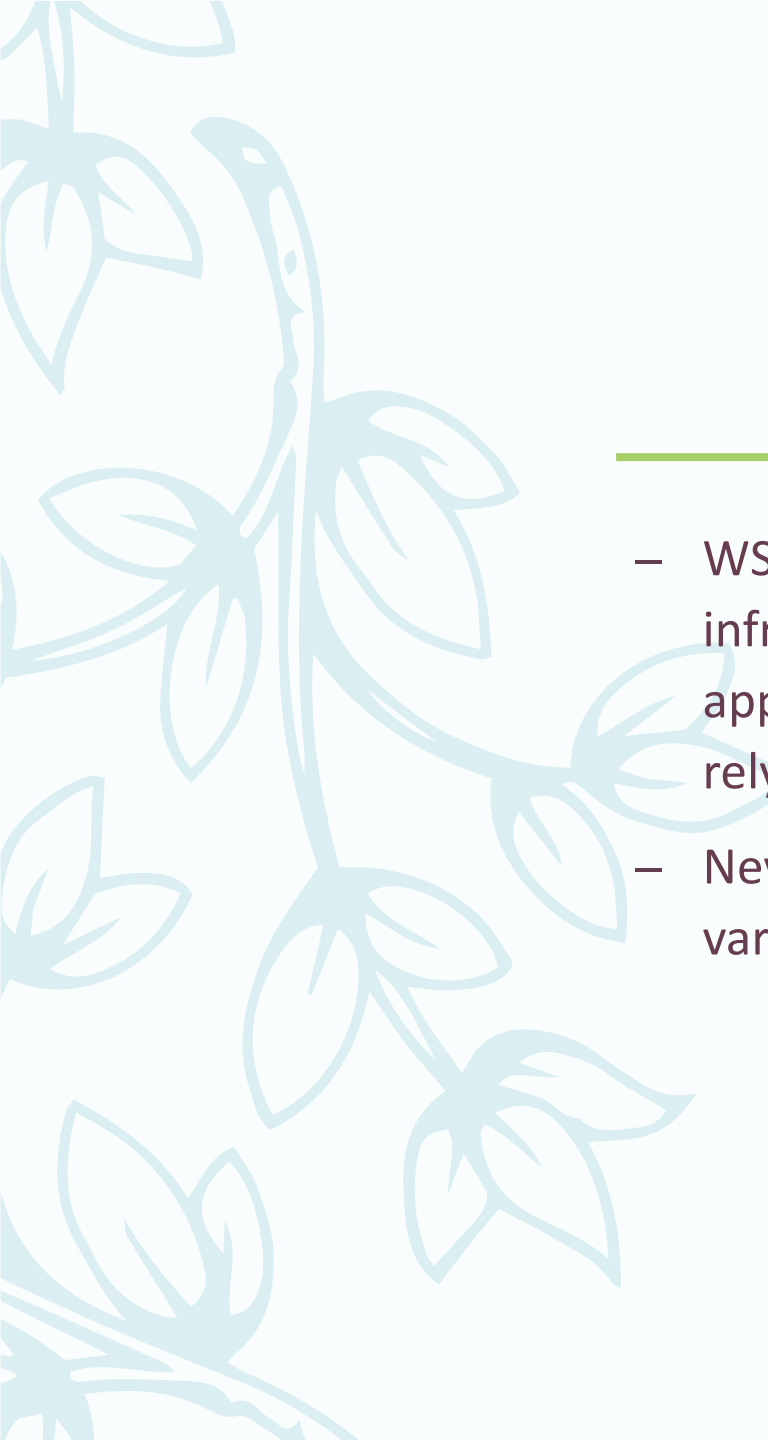
SECURITY

- As adoption of IoT continues to grow, attackers and malicious users are shifting their target from servers to end devices.

There are several reasons for this.


- First, in terms of physical accessibility, smart devices and sensors are far less protected than servers, and having physical access to a device gives the attackers an advantage to penetrate with less hassle.
- Second, the number of devices that can be compromised are far more than the number of servers. Moreover, since devices are closer to the users, security leads to leaking of valuable information and has catastrophic consequences.
- Finally, due to heterogeneity and the distributed nature of IoT, the patching process is more consuming, thus opening the door for attackers.

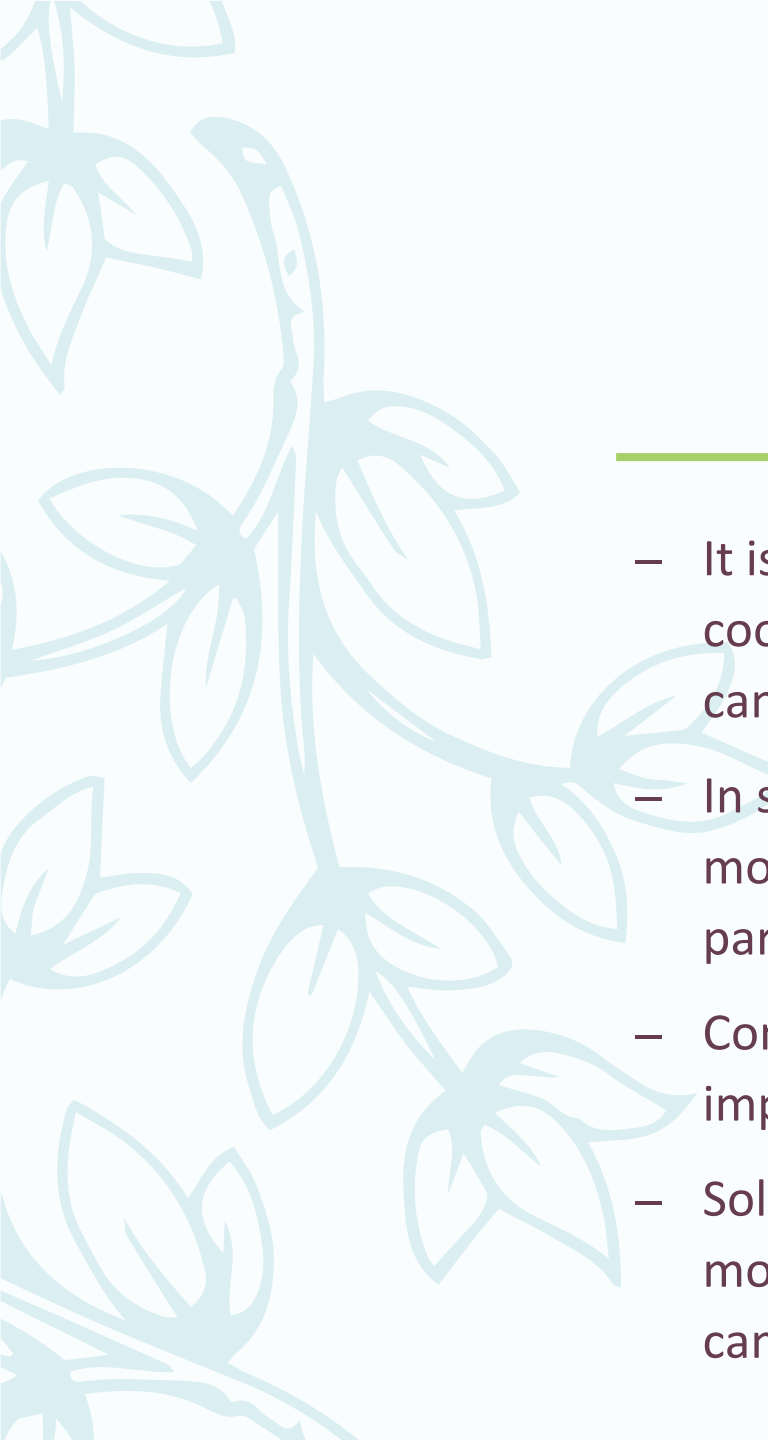
- 
-
- In an IoT environment, resource constraints are the key barrier for implementing standard security mechanisms in embedded devices.
 - Furthermore, wireless communication used by the majority of sensor networks is more vulnerable to eavesdropping and man-in-the-middle (proxy) attacks.
 - Cryptographic algorithms need considerable bandwidth and energy to provide end-to-end protection against attacks on confidentiality and authenticity.
 - Solutions have been proposed in RFID and WSN context to overcome aforementioned issues by considering light cryptographic techniques.
 - With regard to constrained devices, symmetric cryptography is applied more often, as it requires fewer resources; however, public key cryptography in the RFID context has also been investigated.

- 
-
- WSN with RFID tags and their corresponding readers were the first infrastructure for building IoT environments, and, even now, many IoT applications in logistics, fleet management, controlled farming, and smart cities rely on these technologies.
 - Nevertheless, these systems are not secure enough and are vulnerable to various attacks from different layers.

IDENTITY MANAGEMENT AND AUTHENTICATION

- When talking about billions of connected devices, methods for identifying objects and setting their access level play an important role in the whole ecosystem.
- Consumers, data sources, and service providers are essential parts of IoT; identity management and authentication methods applied to securely connect these entities affect both the amount of time required to establish trust and the degree of confidence.
- IoT's inherent features, such as dynamism and heterogeneity, require specific consideration when defining security mechanisms.
- For instance, in Vehicular Networks (VANETs), cars regularly enter and leave the network due to their movement speed; thus, not only do cars need to interact and exchange data with access points and sensors along the road, but they also need to communicate with each other and form a collaborative network.

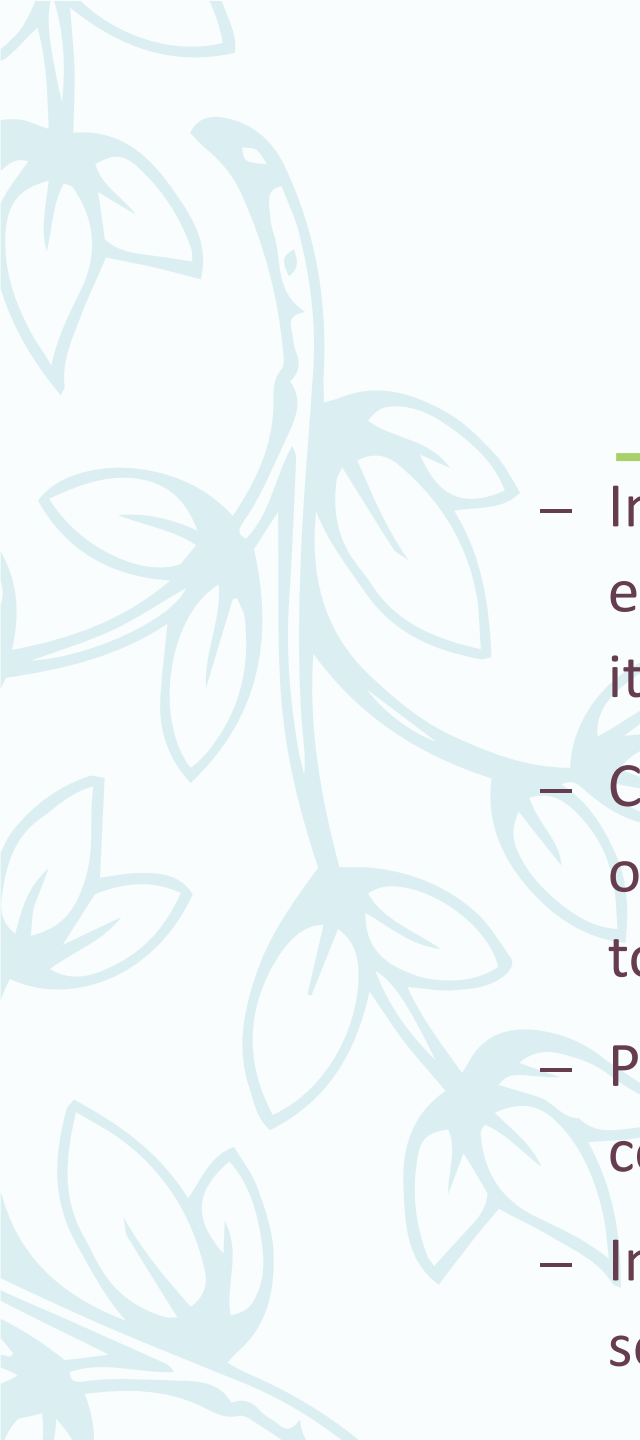
- 
-
- Devices or objects in IoT have to be uniquely identified.
 - There are various mechanisms, such as ucode, which generate 128-bit codes and can be used in active and passive RFID tags, and also Electric Product Code (EPC), which creates unique identifiers using Uniform Resource Identifier (URI) codes.
 - Being able to globally and uniquely identify and locate objects decreases the complexity of expanding the local environment and linking it with the global markets.

- 
-
- It is common for IoT sensors and smart devices to share the same geographical coordinates and even fall into same type or group, hence identity management can be delegated to local identity management systems.
 - In such environments, local identity management systems can enforce and monitor access-control policies and establish trust negotiations with external partners.
 - Context aware pairing of devices and automatic authentication is another important requirement for dynamic environments like IoT.
 - Solutions that implement a zero-interaction approach to create simpler yet more secure procedures for creating a ubiquitous network of connected devices can considerably impact IoT and its adoption.



PRIVACY

- Data generation rate has drastically increased in recent years, and consequently concerns about secure data storage and access mechanisms has be taken more seriously.
- With sensors capable of sensing different parameters, such as users' location, heartbeat, and motion, data privacy will remain a hot topic to ensure users have control over the data they share and the people who have access to these data.

- 
-
- In distributed environments like IoT, preserving privacy can be achieved by either following a centralized approach or by having each entity manage its own inbound/outbound data, a technique known as privacy-by-design.
 - Considering the latter approach, since each entity can access only chunks of data, distributed privacy-preserving algorithms have been developed to handle data scattering and their corresponding privacy tags.
 - Privacy-enhancing technologies are good candidates for protecting collaborative protocols.
 - In addition, to protect sensitive data, rapid deployable enterprise solutions that leverage containers on top of virtual machines can be used.



STANDARDIZATION AND REGULATORY LIMITATIONS

- Standardization and the limitation caused by regulatory policies have challenged the growth and adoption rate of IoT and can be potential barriers in embracing the technology.
- Defining and broadcasting standards will ease the burden of joining IoT environments for new users and providers.
- Additionally, interoperability among different components, service providers, and even end users will be greatly influenced in a positive way, if pervasive standards are introduced and employed in IoT.

- 
-
- Even though more organizations and industries make themselves ready to embrace and incorporate IoT, increase in IoT growth rate will cause difficulties for standardization.
 - Strict regulations about accessing radio frequency levels, creating a sufficient level of interoperability among different devices, authentication, identification, authorization, and communication protocols are all open challenges facing IoT standardization.