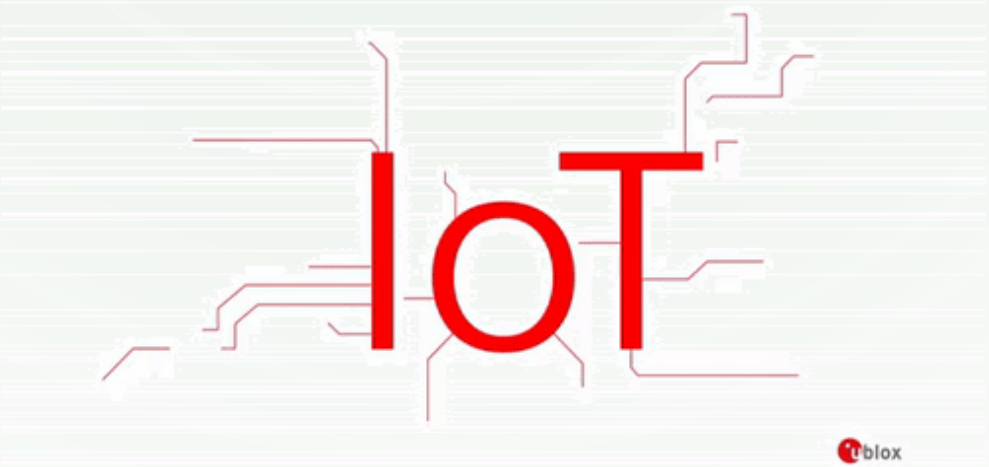
INTERNET OF THINGS

20MCA281

ELECTIVE

L-T-P | 3 1 0
CREDIT - 4



Prepared By: Gloriya Mathew, Asst. Professor, Amal Jyothi College Of Engineering.

MODULE 1 (9 HOURS)

- **Overview of Internet of Things:** Open-source semantic web infrastructure for managing IOT resources in the Cloud - Device/Cloud Collaboration framework for intelligence applications.

Textbook:

- RajkumarBuyya; Amir VahidDastjerdi , “Internet of Things”, Morgan Kaufmann, 2016

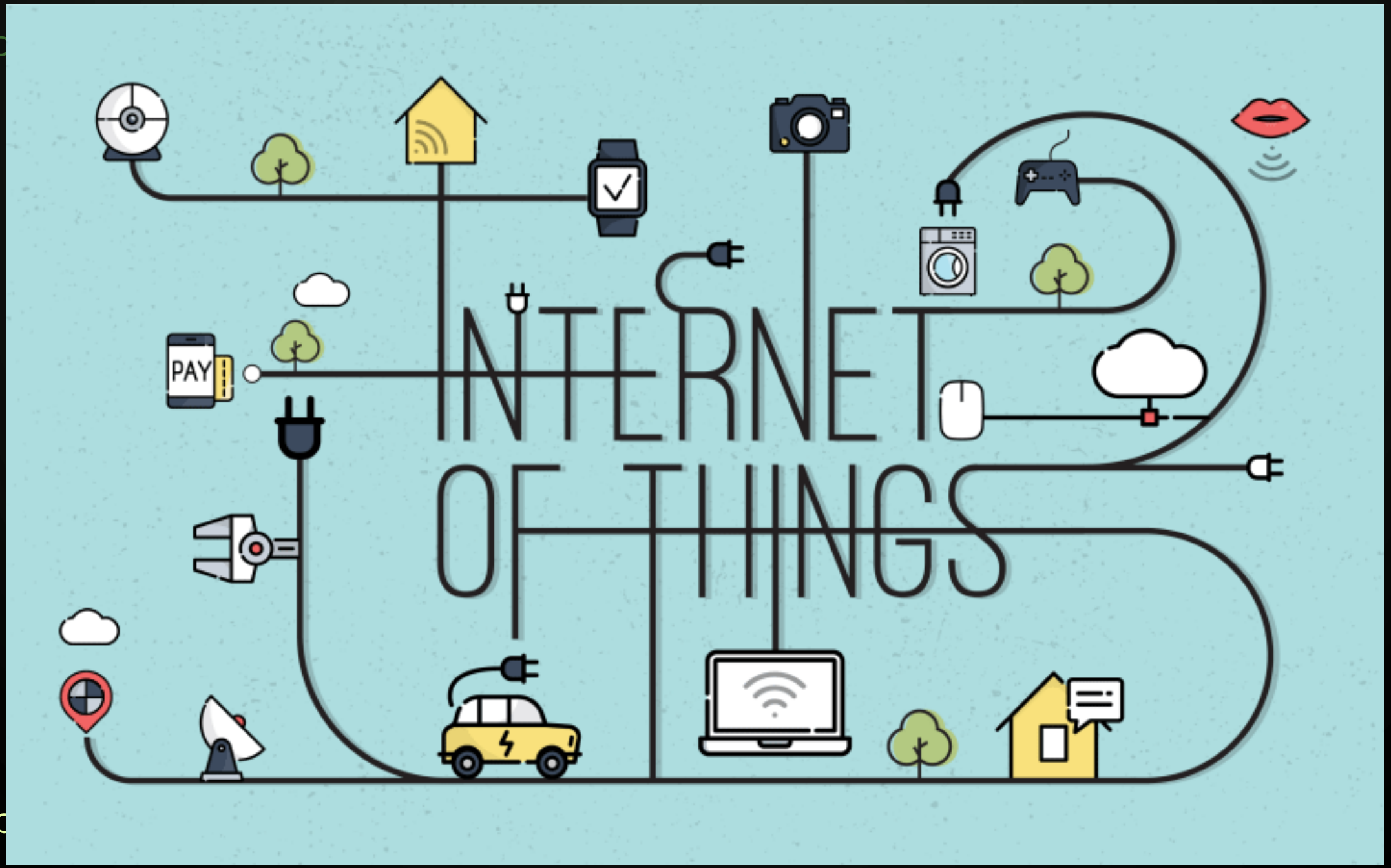
MODULE 1: SYLLABUS

- Internet of things- definition, evolution. Applications -Smart home applications, Health care, Elder care, Traffic surveillance.
- SOA -Based Architecture, API oriented Architecture, Resource Management.
- Computational Offloading, Identification and Resource/Service Discovery, IOT Data Management and Analytics, IOT and the CLOUD.
- Open IOT architecture for IOT/Cloud convergence, Sensor middleware, Cloud computing infrastructure, Directory service, Global Scheduler, Local Scheduler component, Service delivery and utility manager.

MODULE 1: SYLLABUS

- Workflow of open IOT platform, Scheduling process and IOT Services lifecycle, State diagram of the Open IOT Services lifecycle within the scheduler module Scheduling and resource management, Resource optimization schemes, Caching technique Service creation flowchart, Comparison of cost - with cache server and public cloud data-score.
- Runtime adaptation engine, Device/cloud collaboration framework.
- Applications of device/cloud collaboration, Semantic QA cache


IOT
INTERNET
OF THINGS


INTERNET
OF THINGS
PAY

Internet Of Things (IOT)

- The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Internet Of Things (IOT)

- IOT describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.
- These devices range from ordinary household objects to sophisticated industrial tools.
- With more than 7 billion connected IoT devices today;

i. IOT EMERGENCE

- Kevin Ashton is accredited for using the term "Internet of Things" for the first time during a presentation in 1999 on supply-chain management.
- He believes the "things" aspect of the way we interact and live within the physical world that surrounds us needs serious reconsideration, due to advances in computing, Internet, and data-generation rate by smart devices.

ii. Internet of Everything

- Internet of Everything (IoE) is used by Cisco to refer to people, things, and places that can expose their services to other entities

iii. Industrial IoT

- Also referred to as Industrial Internet , Industrial IoT (IIoT) is another form of IoT applications favored by big high-tech companies.
- The fact that machines can perform specific tasks such as data acquisition and communication more accurately than humans has boosted IIoT's adoption.
- Machine to machine (M2M) communication, Big Data analysis, and machine learning techniques are major building blocks when it comes to the definition of IIoT.
- These data enable companies to detect and resolve problems faster, thus resulting in overall money and time savings.

iv. Smartness in IoT

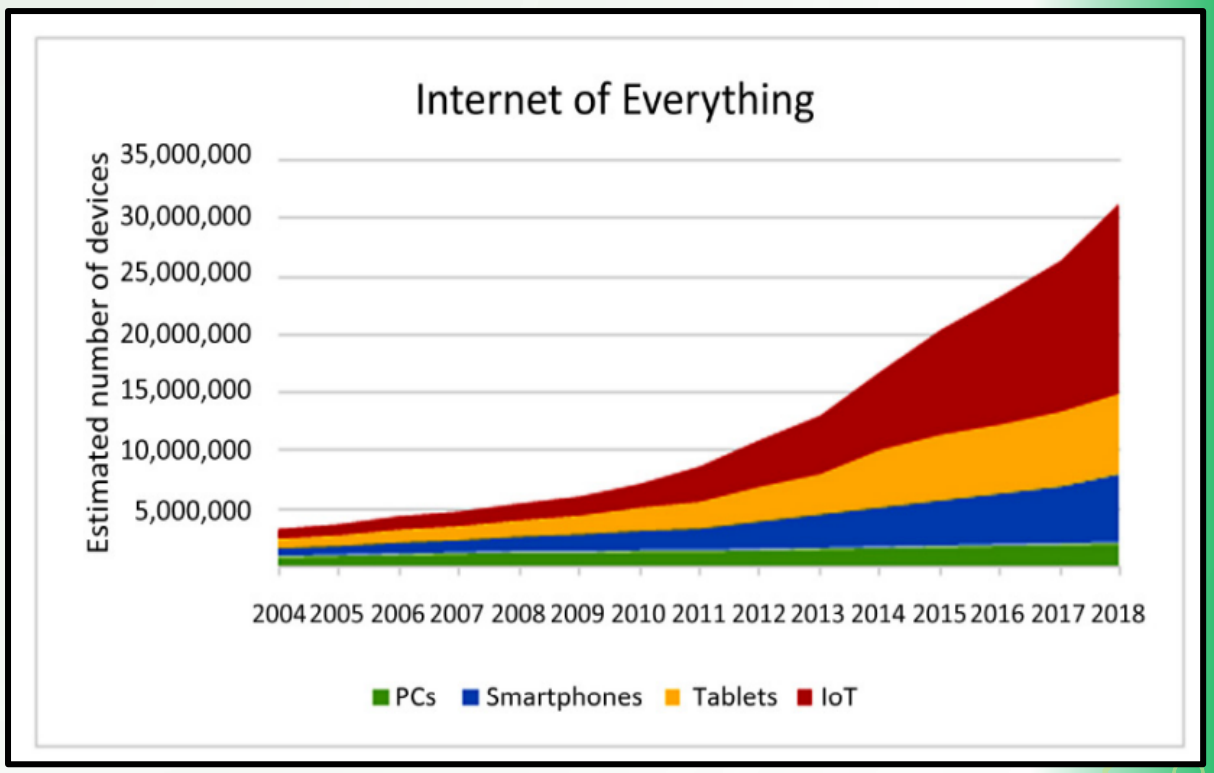
- Another characteristic of IoT, is "smartness."
- "Object Smartness" and "Network Smartness."
- A smart network is a communication infrastructure characterized by the following functionalities:

i. Standardization and openness of the communication standards used, from layers interfacing with the physical world (ie, tags and sensors), to the communication layers between nodes and with the Internet;

ii. Object addressability (direct IP address) and multifunctionality (ie, the possibility that a network built for one application (eg, road-traffic monitoring) would be available for other purposes (eg, environmental-pollution monitoring or traffic safety)

v. Market Share

- potential market of IoT is growing with a fast rate.
- cooperation and information-sharing between leading companies in IoT, such as Microsoft, IBM, Google, Samsung, Cisco, Intel, ARM, Fujitsu, Ecobee Inc., in addition to smaller businesses and start-ups, will boost IoT adoption and market growth.
- IoT growth rate with an estimated number of active devices until 2018 is depicted in Fig.



vi. Human In The Loop

- Involving human in the loop of IoT offers numerous advantages to a wide range of applications, including emergency management, healthcare, etc.

vii. Improving The Quality Of Life

- IoT is also 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 → "Web of Things" and "Internet of Everything.

IOT - Applications

- There are numerous IoT applications , which can be created to be specific to almost every industry.
- IoT applications are using **Artificial Intelligence** and **Machine Learning** to add intelligence to devices.

IOT - Applications

- ✓ Smart home applications
- ✓ Health care
- ✓ Elder care
- ✓ Traffic surveillance.

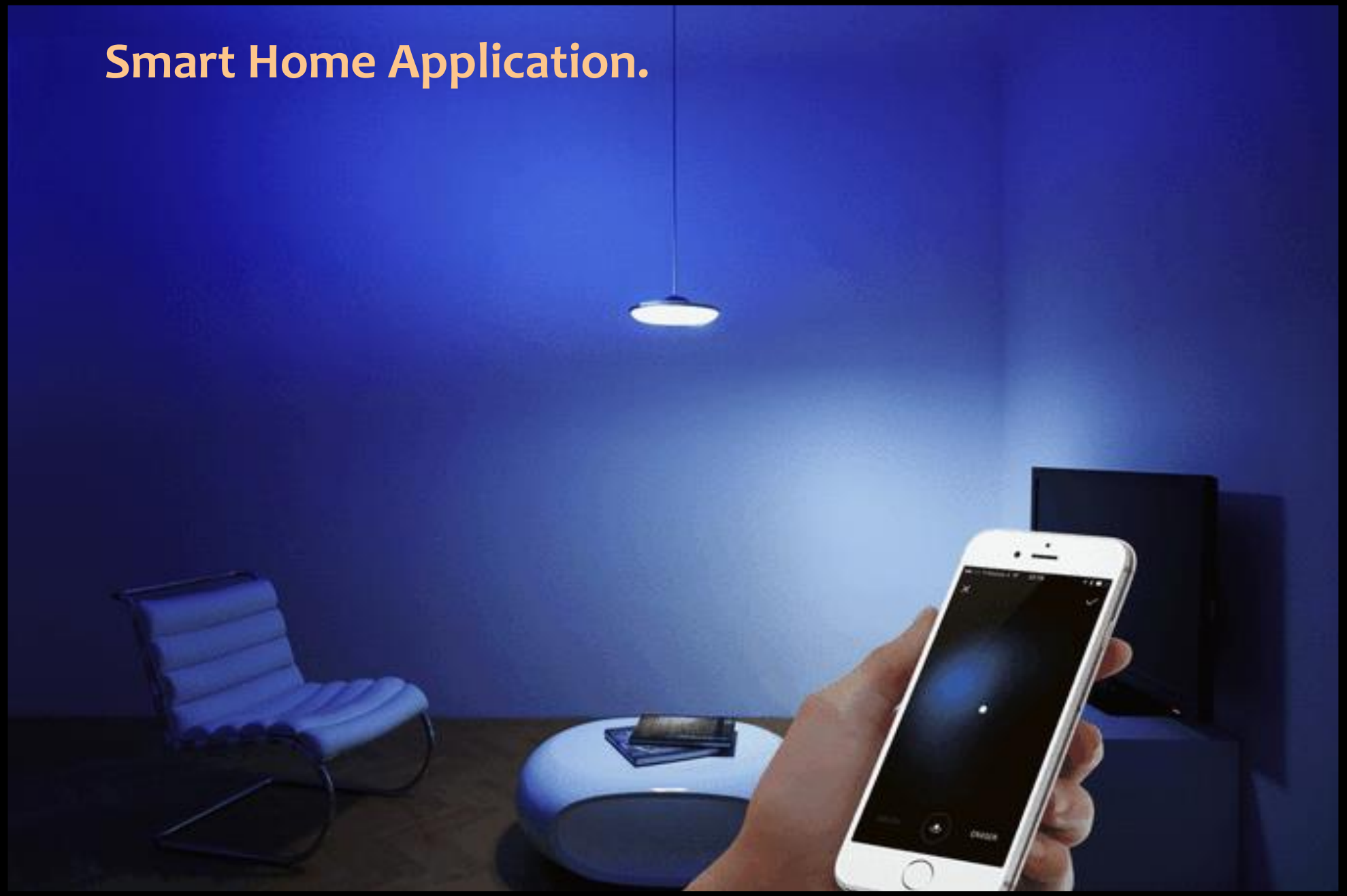
Smart Home Applications

- Smart Home devices collect and share information with one another in an integrated platform and automate their actions based on the owner's preferences.

- **Smart Thermostats**: which monitor and control home temperatures to the comfort of owner.
- **Smart lighting**: lighting adjusts themselves based on the user preferences as well as external lighting.
- **Smart kitchen** : can check for smoke and carbon monoxide or temperature/humidity levels- special programs monitor if the users have enough products in the refrigerator, calculate nutritional value of the meals.
- **Security systems**: With special sensors, controllers can automatically lock the door, turn off electronic devices- users can check their home state remotely through app.

Smart Home Applications

Smart Home Application.



Smart Home Application.

02

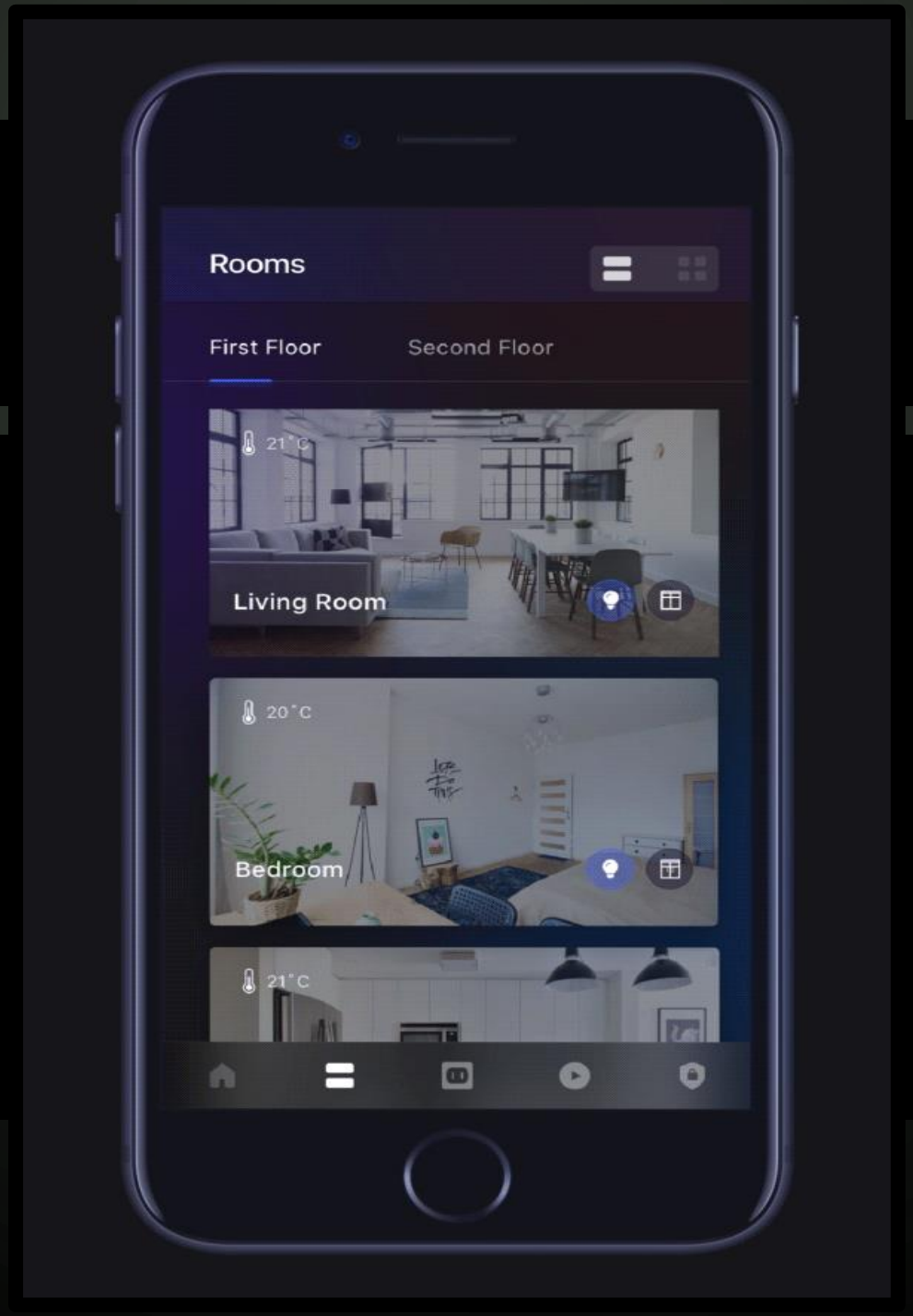
03

CONTROL PANEL

In this app you can controle all smart devices that you have at your home. Just add all your devices to the app and manage them according to your needs.

You can change the shades and the amount of light in your room. Or you can use standard presets for morning, day or night.

Smart Home Application.



Health Care Applications

- **IoT for patients**: Devices in the form of wearable devices like fitness bands, blood pressure watches, ECG headbands etc.
- **IoT for physicians**: Data collected from the IoT devices can help physicians identify the best treatment process for the patients.
- **IoT for hospitals**: IoT devices tagged with the sensors are used for tracking real time location of medical equipments- Deployment of medical staff at different locations can also be analyzed real time.
- **IoT for health insurance companies**: The insurance companies can leverage data captured through health monitoring devices, which enable them to detect fraud claims.

Health Care

PARKS ASSOCIATES

NEXT EVOLUTION OF HEALTH IoT: MOVING TOWARDS THE EDGE

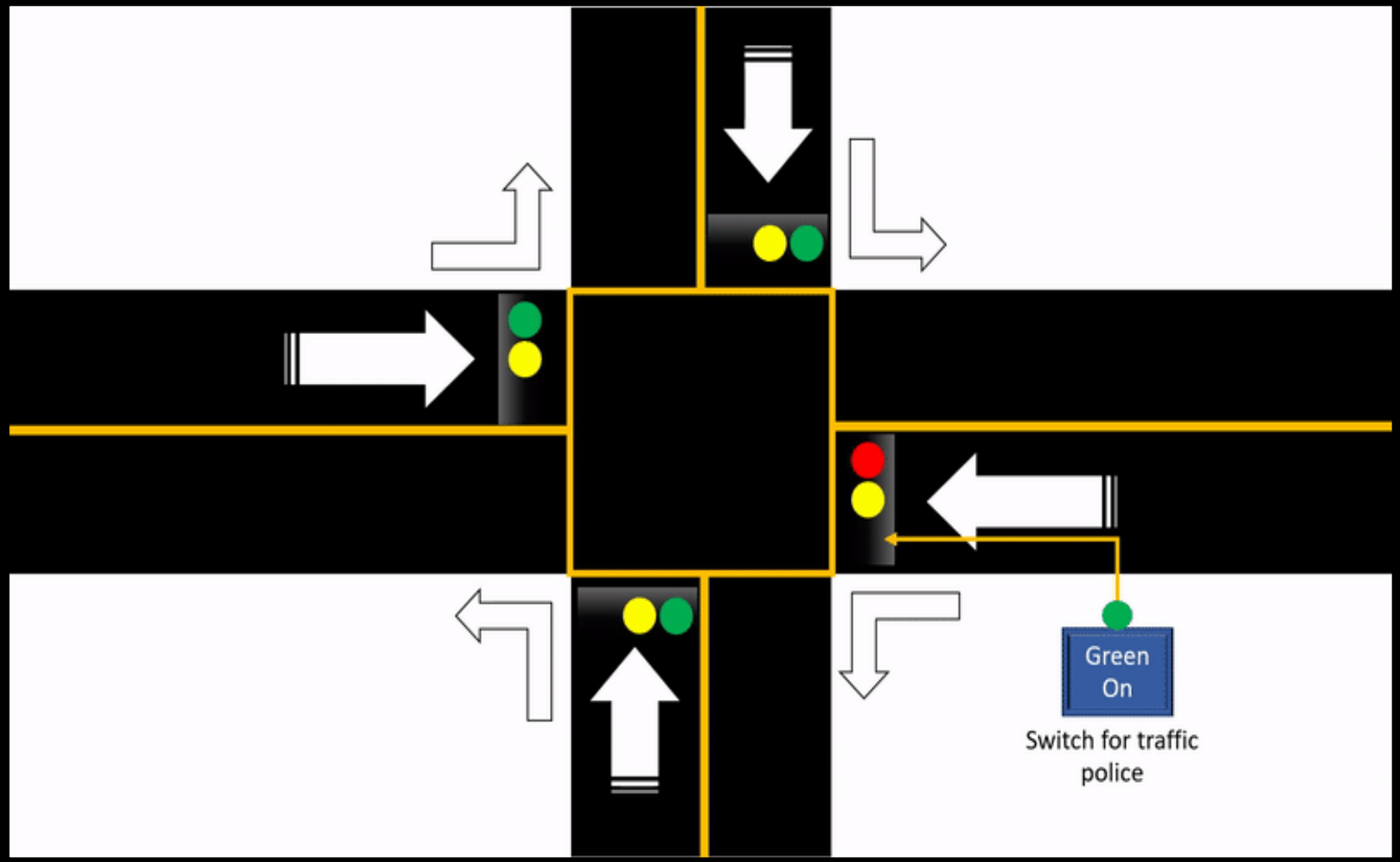
Health Care System

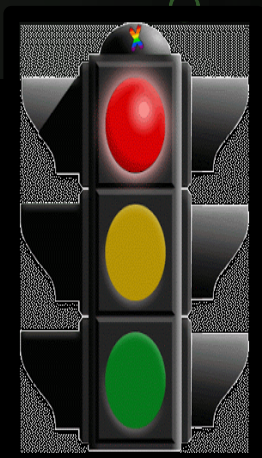

HOSPITAL

Elder Care


16:25
Google Fit
Today 29 April
74/100
Move Min
23/20
Heart Pts
8225
steps
1,876
Cals
2.2
miles
Heart rate
97 bpm • 3 min ago
13:17
0/100
Move Min
0/20
Heart Pts
0
steps
0
Cals
0
miles
100 Move Minutes a day
150
100
Fri
Sat
Sun
Mon
Tue
Wed
Home
Journal
Profile
74/100
Move Min
23/20
Heart Pts
y activity
Week
Month
Today, April 29
16 mins at 10:30 am
30
15
4
8
12
16
20
24
Move Minutes
Heart Points
ute back home
pm • Strava
time run
am
Plaça de Gràcia

Traffic Surveillance





Traffic Management System

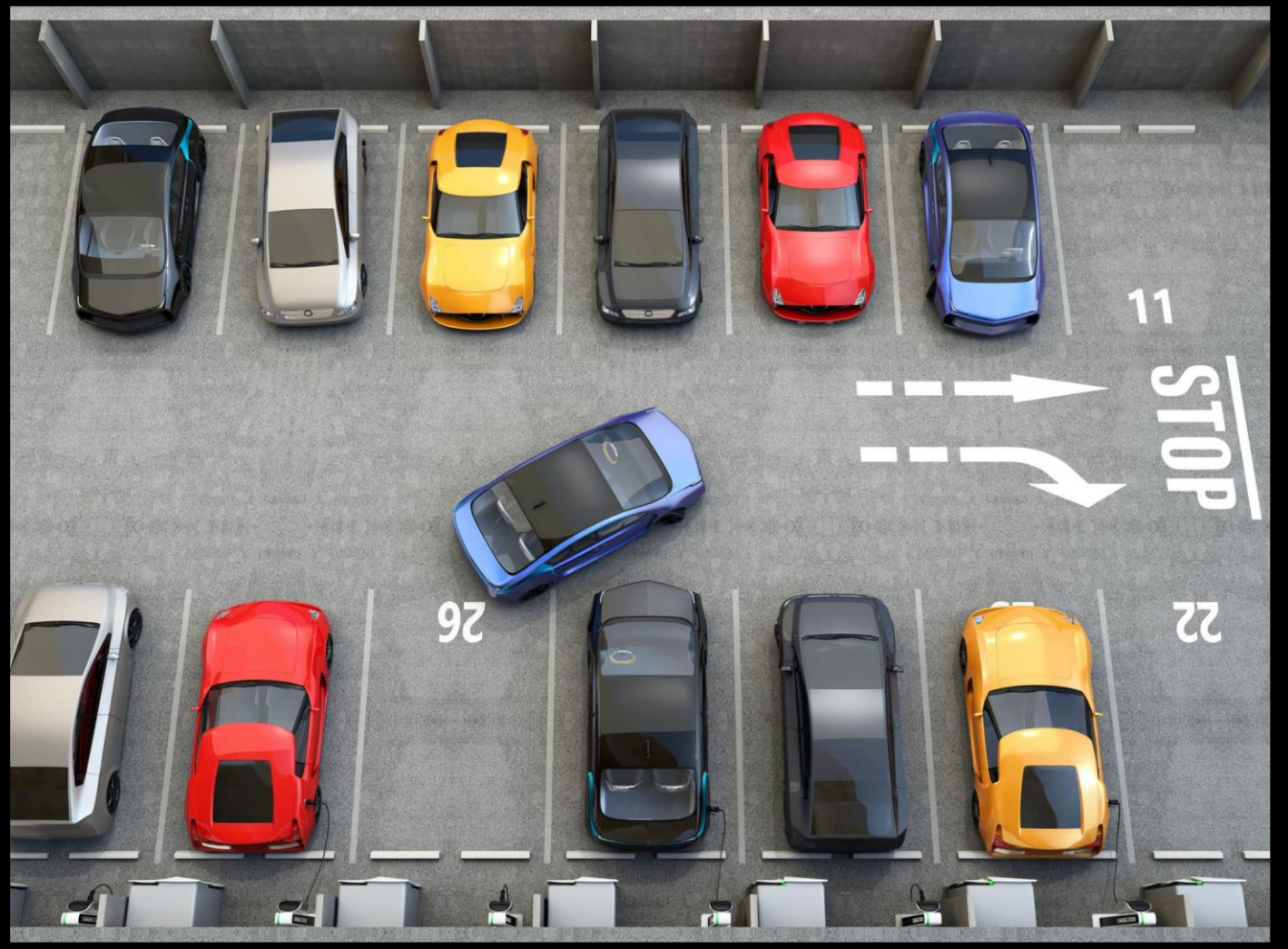
Traffic Surveillance

- **Smart traffic control**
- **Vehicle control**
- **Smart parking**
- **Connected car**
- **Drowsiness alerts**
- **Electronic toll collection system**
- **Logistics and fleet management**
  - and many others…

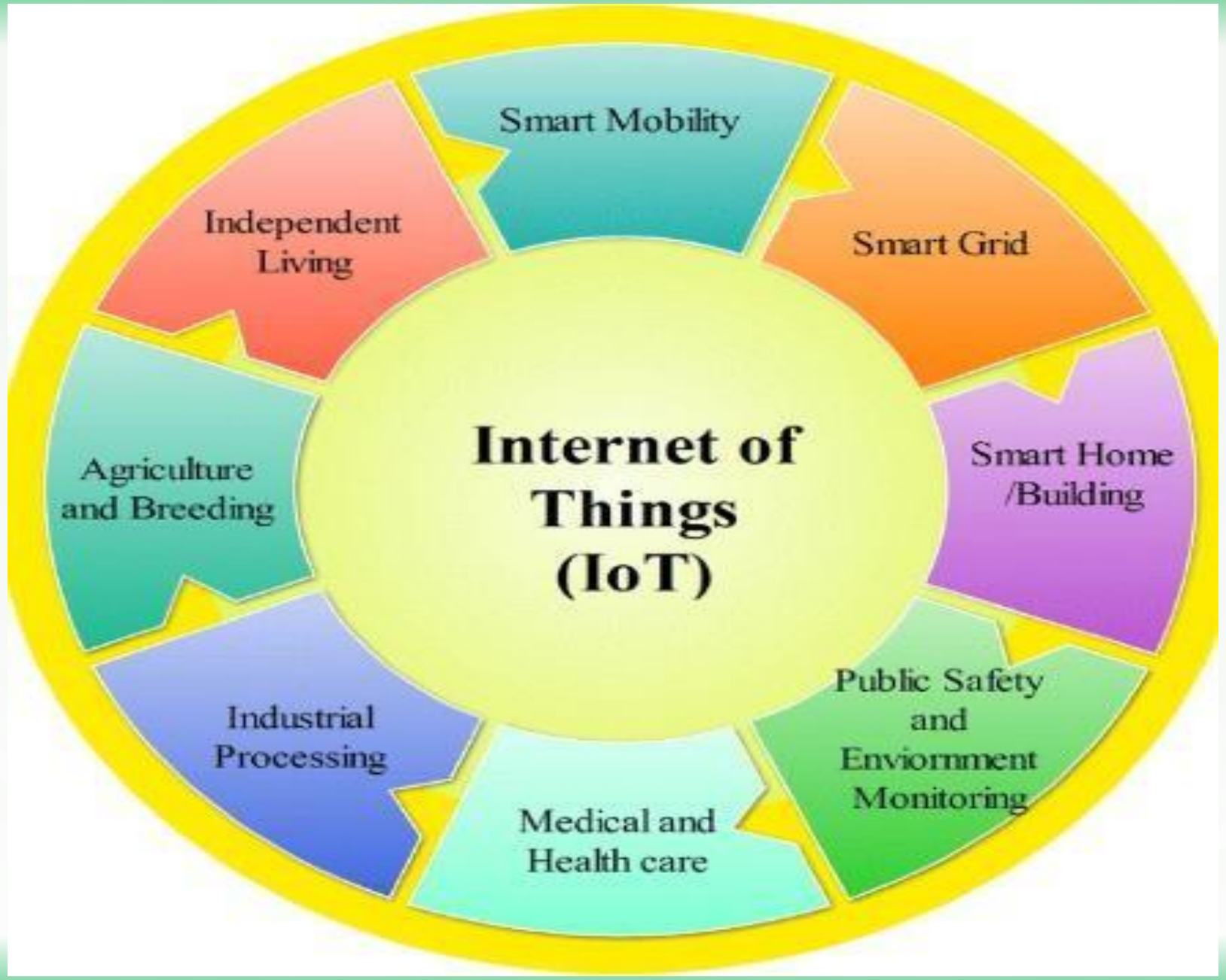
IoT Based Intelligent Traffic Signal Monitoring System for Cities

MANTRA

Traffic Management System – Vehicle Classification & Counting



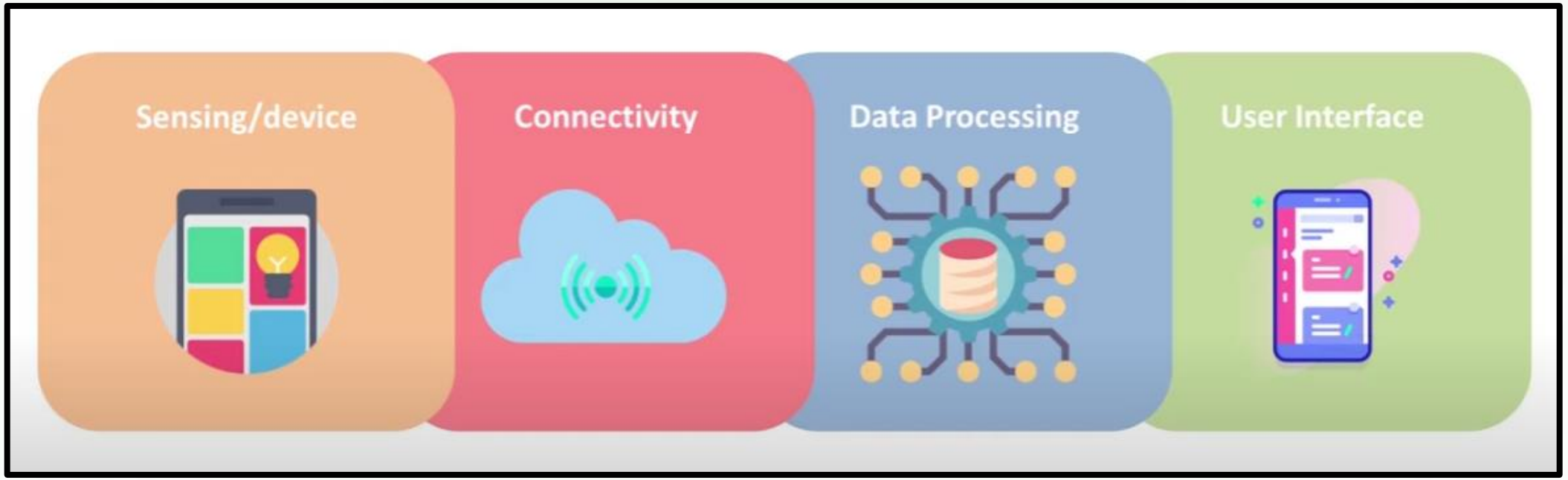
IOT BASED SMART PARKING SYSTEM


Smart Mobility
Smart Grid
Smart Home /Building
Public Safety and Enviornment Monitoring
Medical and Health care
Industrial Processing
Agriculture and Breeding
Independent Living
Internet of Things (IoT)

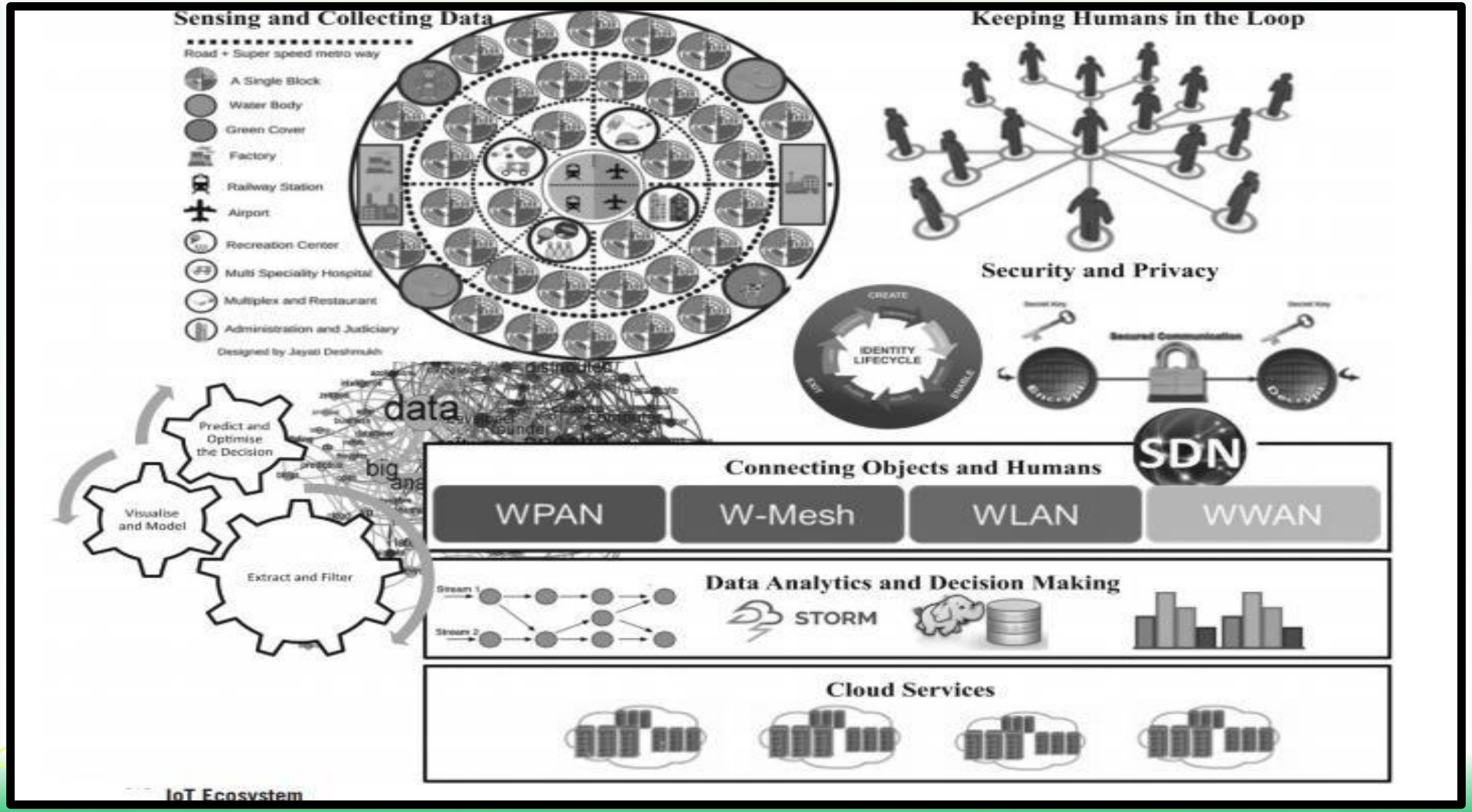
IOT ARCHITECTURES

- IoT has a wide variety of applications-depending upon different application areas of IoT, it works as per it has been designed.
- It does not have a standard defined architecture of working, the architecture of IoT depends upon its functionality and implementation in different sectors.
- **Building blocks of IoT:**
  - Sensory devices
  - Remote service invocation
  - Communication networks
  - Context aware processing

IOT - LAYERS


Sensing/device
Connectivity
Data Processing
User Interface

IOT ECOSYSTEM



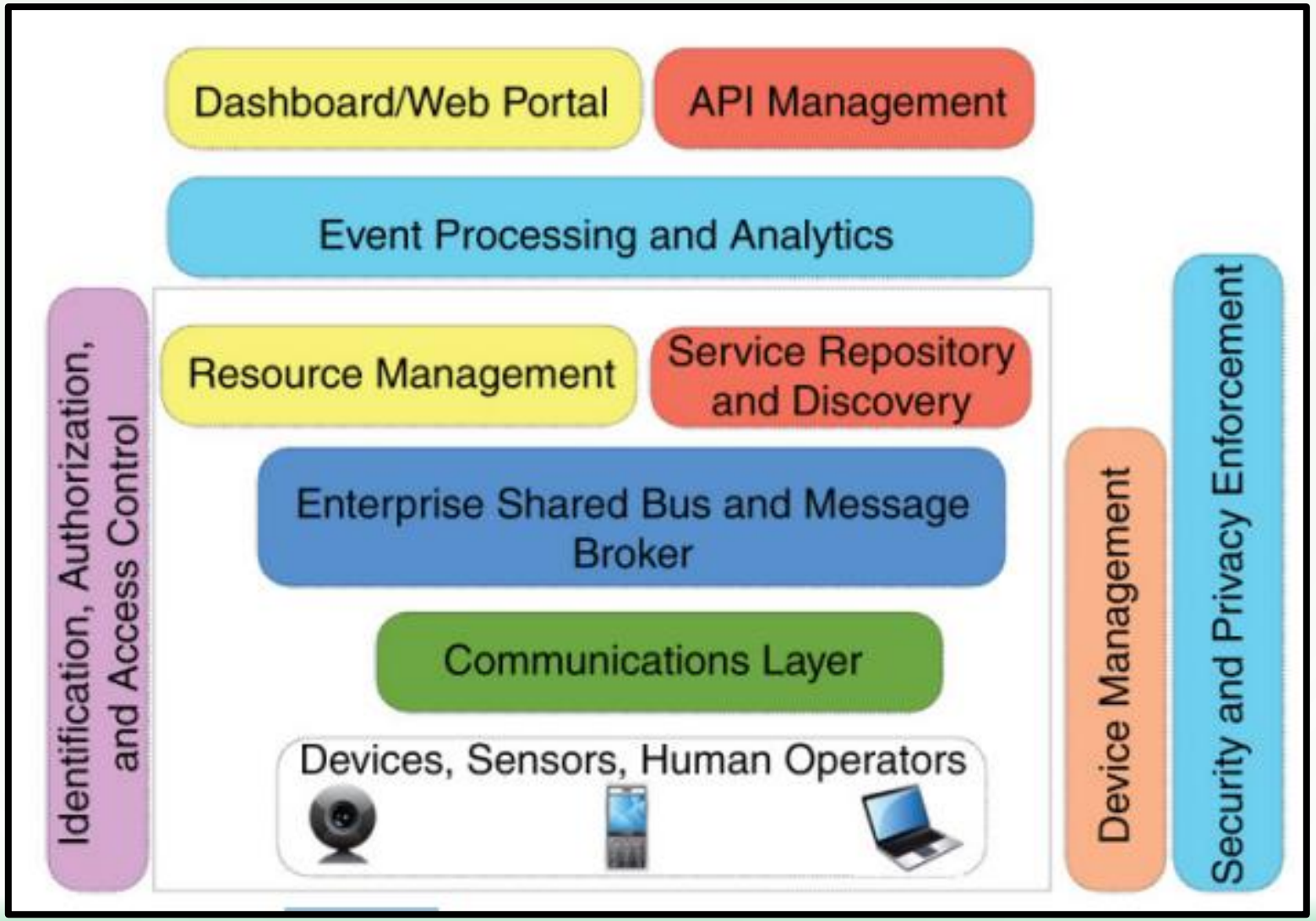
IOT ARCHITECTURE:

- SOA -Based Architecture
- API oriented Architecture
- Resource Management.

IOT ARCHITECTURE

- The building blocks of IoT are sensory devices, remote service invocation, communication networks, and context-aware processing of events;
- Reference architectures and models give a bird's eye view of the whole underlying system, hence their advantage over other architectures relies on providing a better and greater level of abstraction, which consequently hides specific constraints and implementation details.

Reference Architecture for IOT



Reference Architecture for IOT

- Depicts an outline of our extended version of 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

- In IoT, Service-oriented Architecture (SOA) might be imperative for the service providers and users.
- SOA ensures the interoperability among the heterogeneous devices.
- SOA consisting of Four layers:
  1) Sensing Layer is integrated with available hardware objects to sense the status of things
  2) Network Layer is the infrastructure to support over wireless or wired connections among things
  3) Service Layer is to create and manage services required by users or applications
  4) Interfaces Layer consists of the interaction methods with users or applications

SOA-BASED ARCHITECTURE

- In this architecture a complex system is divided into subsystems that are loosely coupled and can be reused later (modular decomposability feature), hence providing an easy way to maintain the whole system by taking care of its individual components.
- This can ensure that in the case of a component failure the rest of the system (components) can still operate normally.
- This is of immense value for effective design of an IoT application architecture, where reliability is the most significant parameter.

SOA-BASED ARCHITECTURE

- intensively used in WSN (Wireless sensor Networks) - due to its appropriate level of abstraction and advantages pertaining to its modular design .
- Bringing these benefits to IoT, SOA has the potential to augment the level of interoperability and scalability among the objects in IoT.
- Moreover, from the user's perspective, all services are abstracted into common sets, removing extra complexity for the user to deal with different layers and protocols.
- Additionally, the ability to build diverse and complex services by composing different functions of the system (ie, modular composability) through service composition suits the heterogeneous nature of IoT, where accomplishing each task requires a series of service calls on all different entities spread across multiple locations

API - ORIENTED ARCHITECTURE

- due to overhead and complexity imposed by the conventional techniques, Web APIs and Representational State Transfer (REST)-based methods were introduced as promising alternative solutions.
- The required resources range from network bandwidth to computational and storage capacity, and are triggered by request–response data conversions happening regularly during service calls.
- Lightweight data-exchange formats like JSON can reduce the aforementioned overhead, especially for smart devices and sensors with a limited amount of resources, by replacing large XML files used to describe services.
- This helps in using the communication channel and processing the power of devices more efficiently

API - ORIENTED ARCHITECTURE

- **Application Programming Interface (API)**
- **Application programming interfaces, or APIs, simplify software development and innovation by enabling applications to exchange data and functionality easily and securely.**
- ***API architecture → refers to the process of developing a software interface that exposes backend data and application functionality for use in new applications.***
- **With an API-first architecture, you can create ecosystems of applications that are modular and reusable — which is ideal for microservices.**

API - ORIENTED ARCHITECTURE

- Building APIs for IoT applications helps the service provider attract more customers while focusing on the functionality of their products rather than on presentation.
- In addition, it is easier to enable multitenancy by the security features of modern Web APIs such as Oauth(Open Authorization), APIs which indeed are capable of boosting an organization's service exposition and commercialization.
- It also provides more efficient service monitoring and pricing tools than previous service-oriented approaches.

API - ORIENTED ARCHITECTURE

- Developers and business managers are advised to focus on developing and sharing APIs from the early stage of their application development lifecycle, so that eventually, by properly exposing data to other developers and end users, an open-data environment is created that facilitates collaborative information gathering, sharing, and updating.

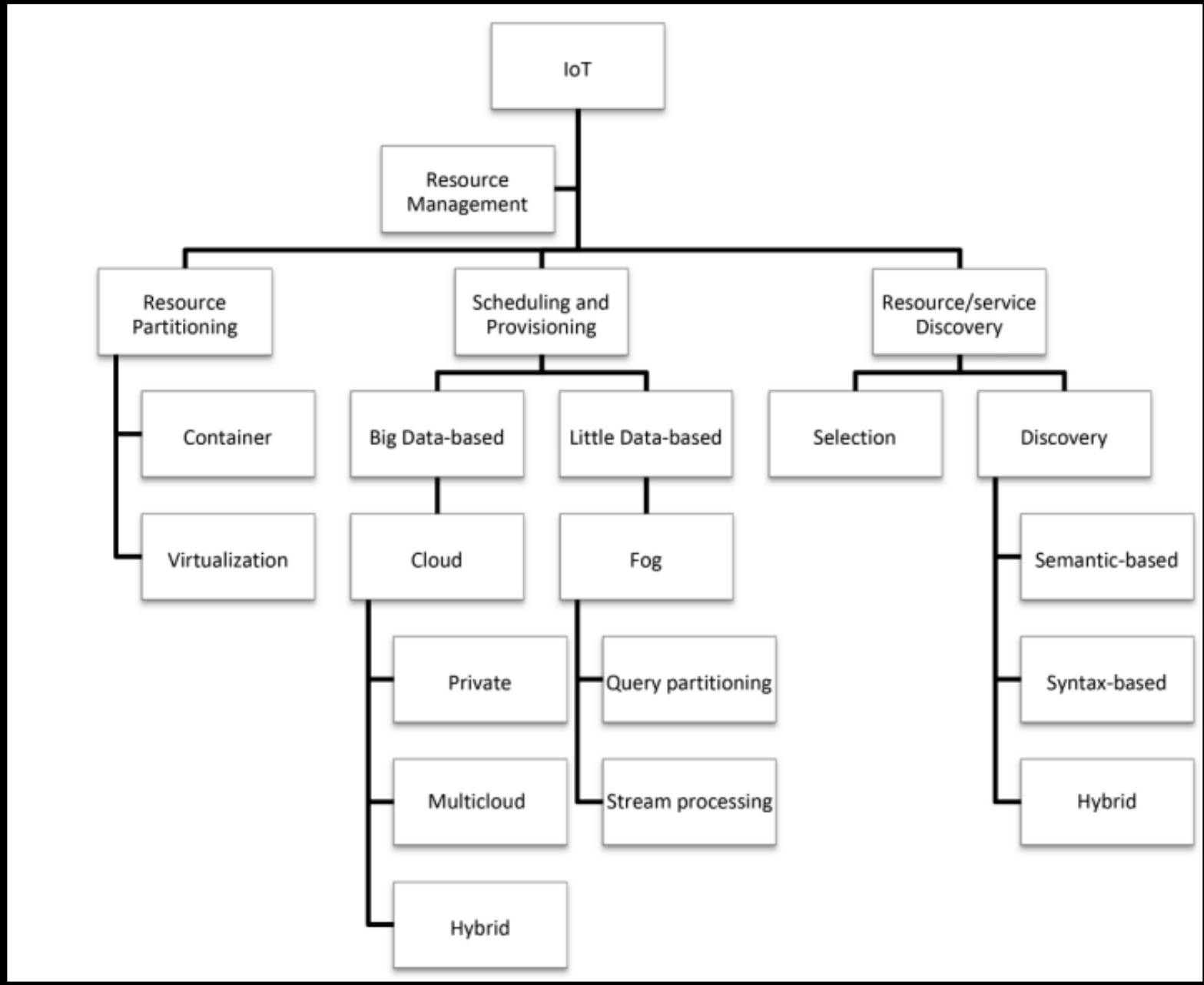
RESOURCE MANAGEMENT:

i. RESOURCE PARTITIONING
ii. COMPUTATION OFFLOADING
iii. IDENTIFICATION AND RESOURCE/SERVICE DISCOVERY

Resource Management

- Resource management is very important in distributed systems
- resource management in IOT is more challenging since it, relies on the heterogeneous and dynamic nature of resources in IoT.
- Considering 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(service-level agreement).
- Resource management involves:
  i. discovering and identifying all available resources,
  ii. partitioning them to maximize a utility function—which can be in terms of cost, energy, performance, etc., and,
  iii. finally, scheduling the tasks on available physical resources.

TAXONOMY OF RESOURCE MANAGEMENT IN IoT



i. Resource Partitioning

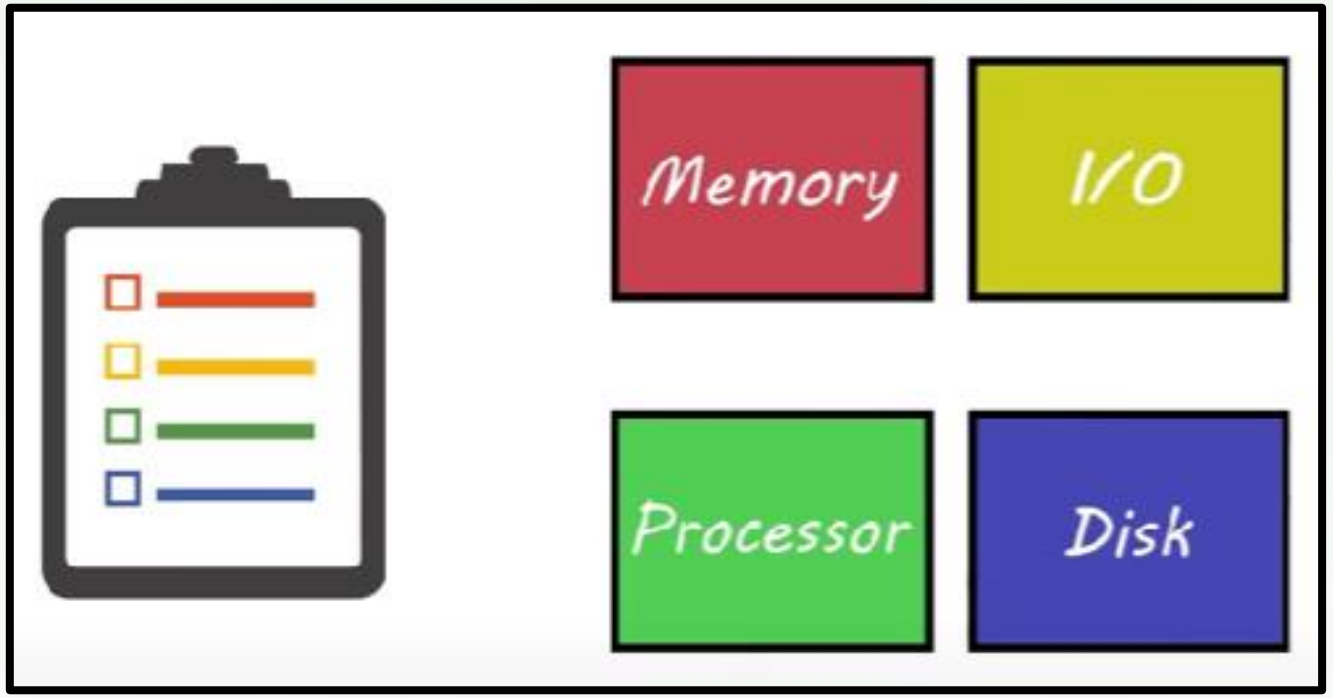
- The first step for satisfying resource provisioning requirements in IoT is:
  - To efficiently partition the resources and gain a higher utilization rate.
- The concept of Containers has emerged as a new form of virtualization technology that can match the demand of devices with limited resources.
- Docker and Rocket are the two most famous container solutions.
  - (https://www.docker.com/)
  - (https://github.com/coreos/rkt)

i. Resource Partitioning

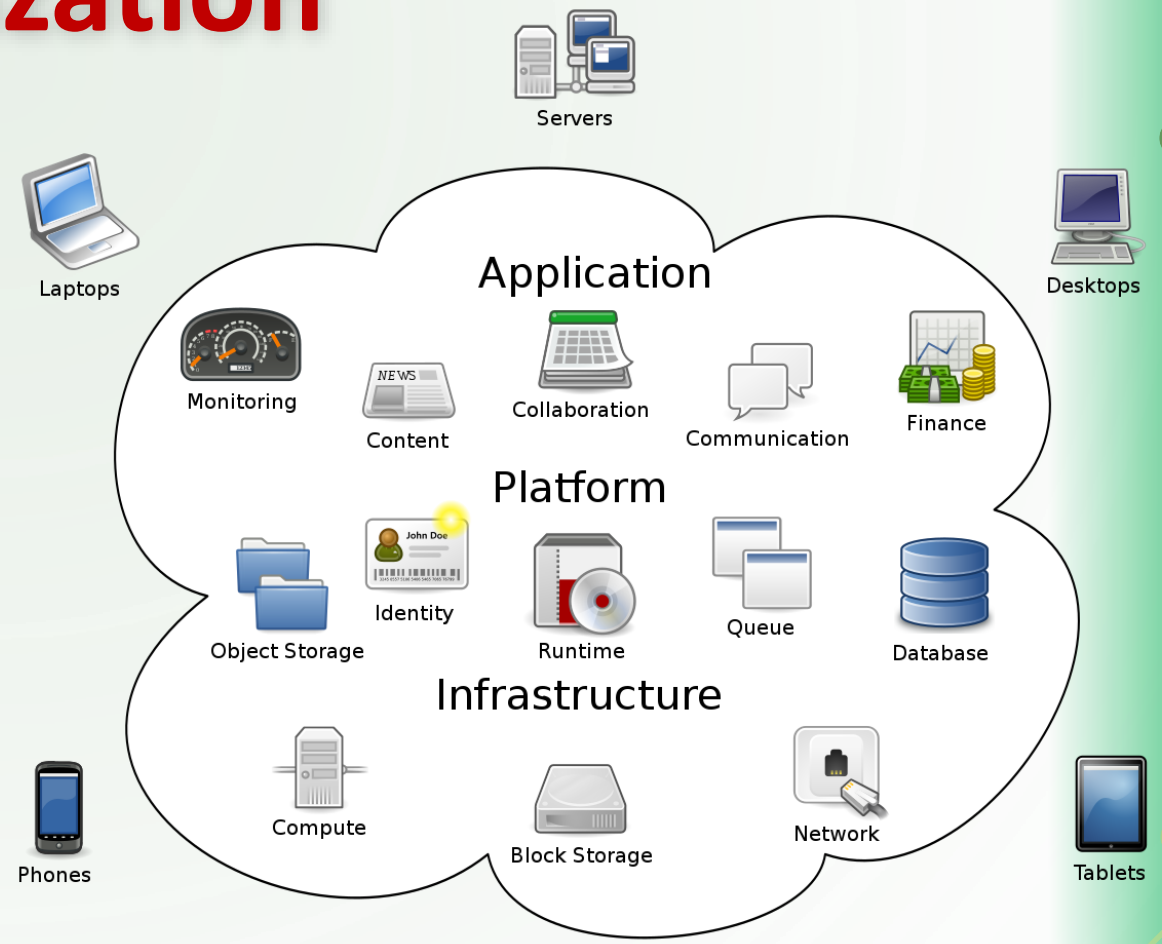
Containers:

- 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.
- 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.
- Require considerably less spin-up time.
- They are ideal for distributed applications in IoT.

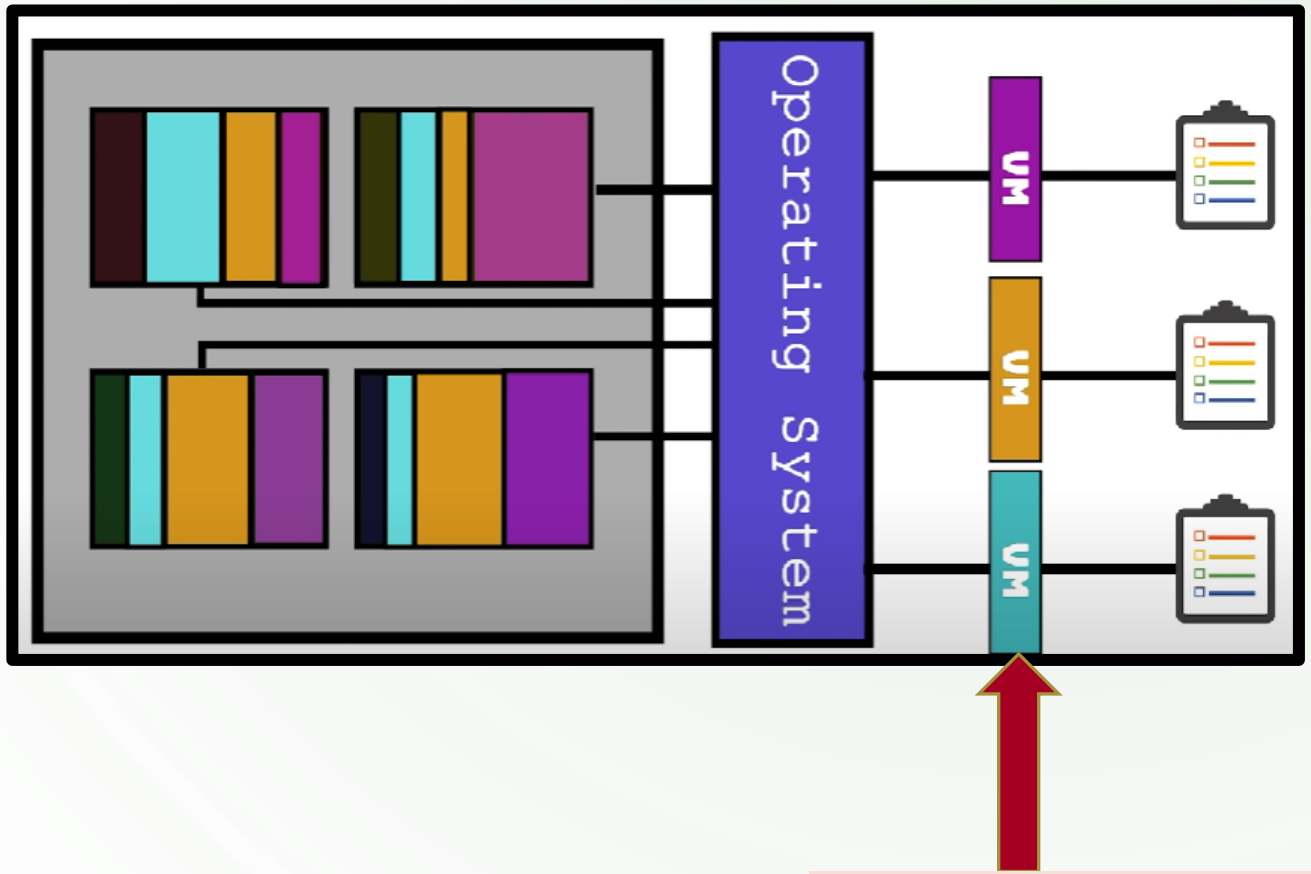
Resources:



Virtualization

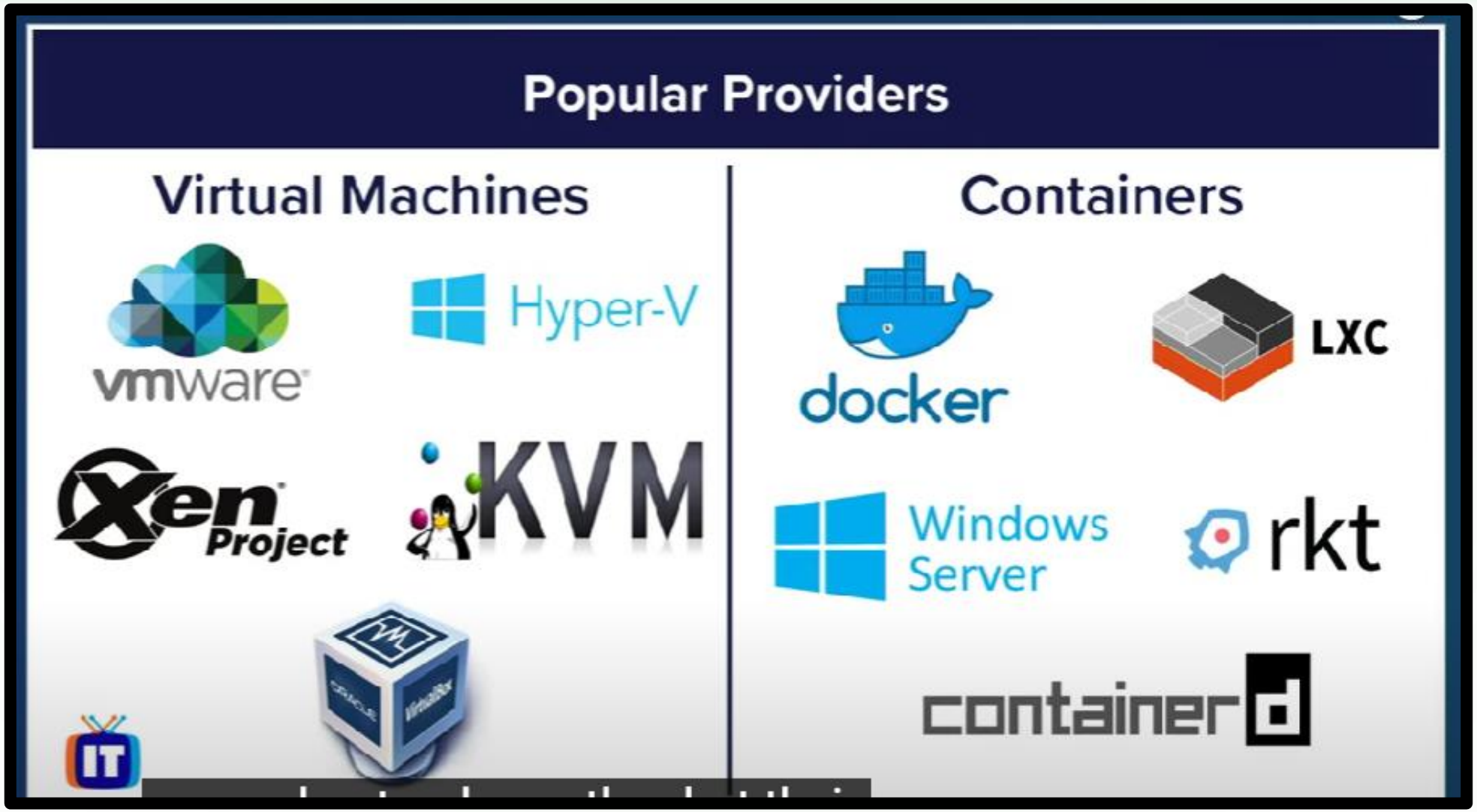


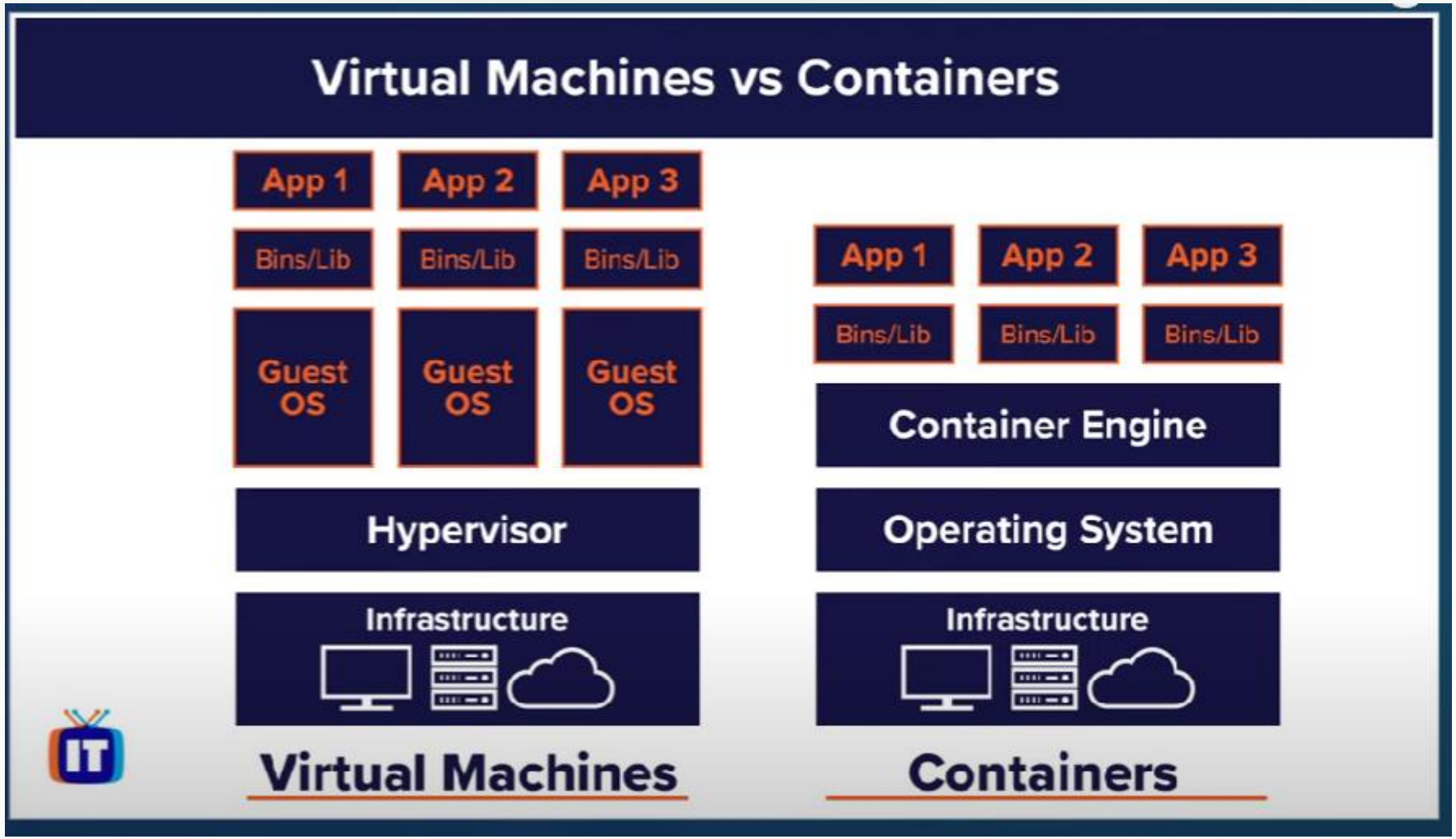
Cloud Computing



Virtualization→Cloud Computing

VM → Virtual Machine


Popular Providers
Virtual Machines
Containers
vmware
Hyper-V
docker
LXC
Xen Project
KVM
Windows Server
rkt
containerd


Virtual Machines vs Containers
App 1
App 2
App 3
Bins/Lib
Bins/Lib
Bins/Lib
Guest OS
Guest OS
Guest OS
Hypervisor
Infrastructure
Virtual Machines
App 1
App 2
App 3
Bins/Lib
Bins/Lib
Bins/Lib
Container Engine
Operating System
Infrastructure
Containers

- container-based virtualization, can bring advantages in terms of performance and security by sandboxing applications on top of a shared OS layer.
- Examples of using OS virtualization in an embedded systems domain are:
  - Linux VServer,
  - Linux Containers LXC, and
  - OpenVZ

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.
- *Different approaches of code offloading focus on efficient code segmentation and cloud computing.*
- The proposed combination of VMs and mobile clouds can create a powerful environment for sharing, synchronizing, and executing codes in different platforms.

Computation Offloading

- Majority of code offloading techniques require the developers to manually annotate the functions required to execute on another device.
- Using static code analyzers and dynamic code parsers is an alternative approach that results in better adaptivity in case of network fluctuations and increased latency.
- Instead of using physical instances, ThinkAir and COMET leverage virtual machines offered by IaaS cloud providers as offloading targets to boost both scalability and elasticity.
- 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 two fold.

i. To identify and locate the actual device, which can be achieved by storing and indexing metadata information about each object.

ii. 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.
- The Semantic Web of Things (SWoT) envisions advanced resource management and service discovery for IoT by extending Semantic Web notation and blending it with IoT and Web of Things.

IoT Data Management and Analytics.

- **Special considerations** are required to process **huge amounts** of data originating from, and circulating in, 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 IoT.
- **Big Data** is characterized by **3Vs:**
  - **Velocity,**
  - **Volume, and**
  - **Variety**

IoT Data Management and Analytics.

- Focusing on either an individual or a combination of these three Big Data dimensions has led to the introduction of different data-processing approaches:
- Two methods used for data analysis are:
  - Batch Processing
  - Stream Processing
- Applications utilize pattern detection and data-mining techniques to extract knowledge and make smarter decisions - but the centralized nature of these algorithms, affects their performance and makes them unsuitable for IoT environments that are meant to be geographically distributed and heterogeneous.

IoT Data Management and Analytics (cntd.).

- **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.

i. "**Batch Layer**" → Hosts the master dataset and batch views where precomputed queries are stored.
ii. "**Serving Layer**," → Adds dynamic query creation and execution to the batch views by indexing and storing them.
iii. "**Speed Layer**" → Captures and processes recent data for delay-sensitive queries.

IOT AND THE CLOUD

- Cloud computing 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 auto scaling 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.

IOT AND THE CLOUD

- Most of the current design approaches for integrating cloud with IoT are based on a three-tier architecture:
  - Bottom layer consists of IoT devices,
  - Middle layer is the cloud provider
  - 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 . Eg: Online game streaming)

REAL-TIME ANALYTICS IN IOT AND FOG COMPUTING

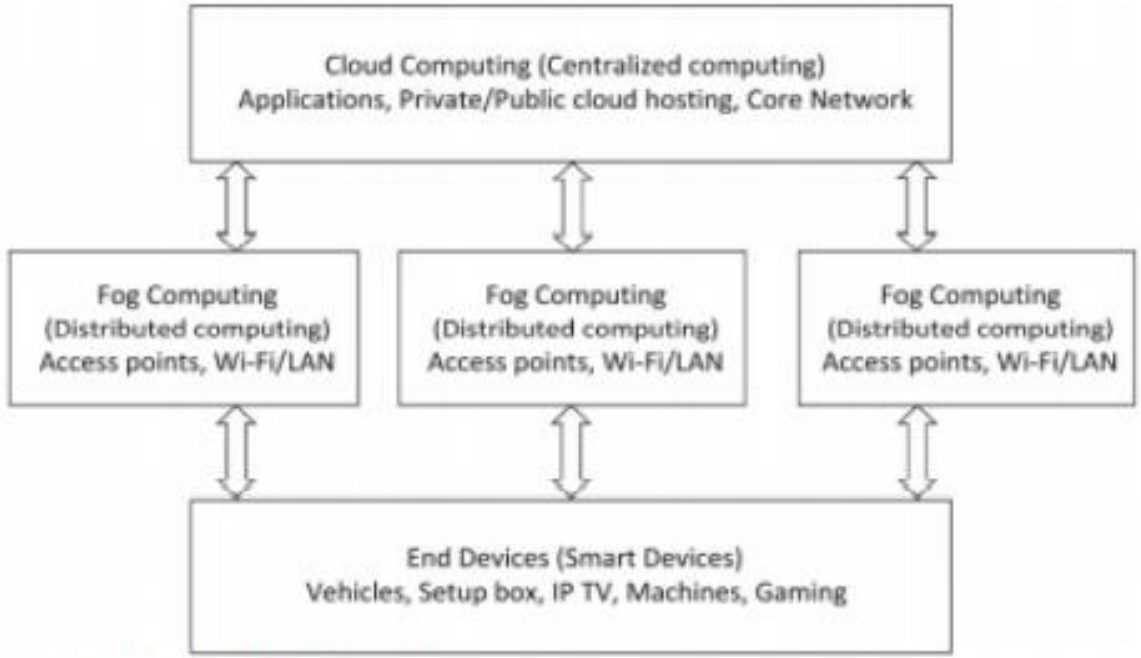
- Current data-analytics approaches mainly focus on dealing with Big Data, however, processing data generated from millions of sensors and devices in real time is more challenging.
- Proposed solutions that only utilize cloud computing as a processing or storage backbone are not scalable and cannot address the latency constraints of realtime 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 smart phones, smart TVs, network routers etc.

REAL-TIME ANALYTICS IN IOT AND FOG COMPUTING

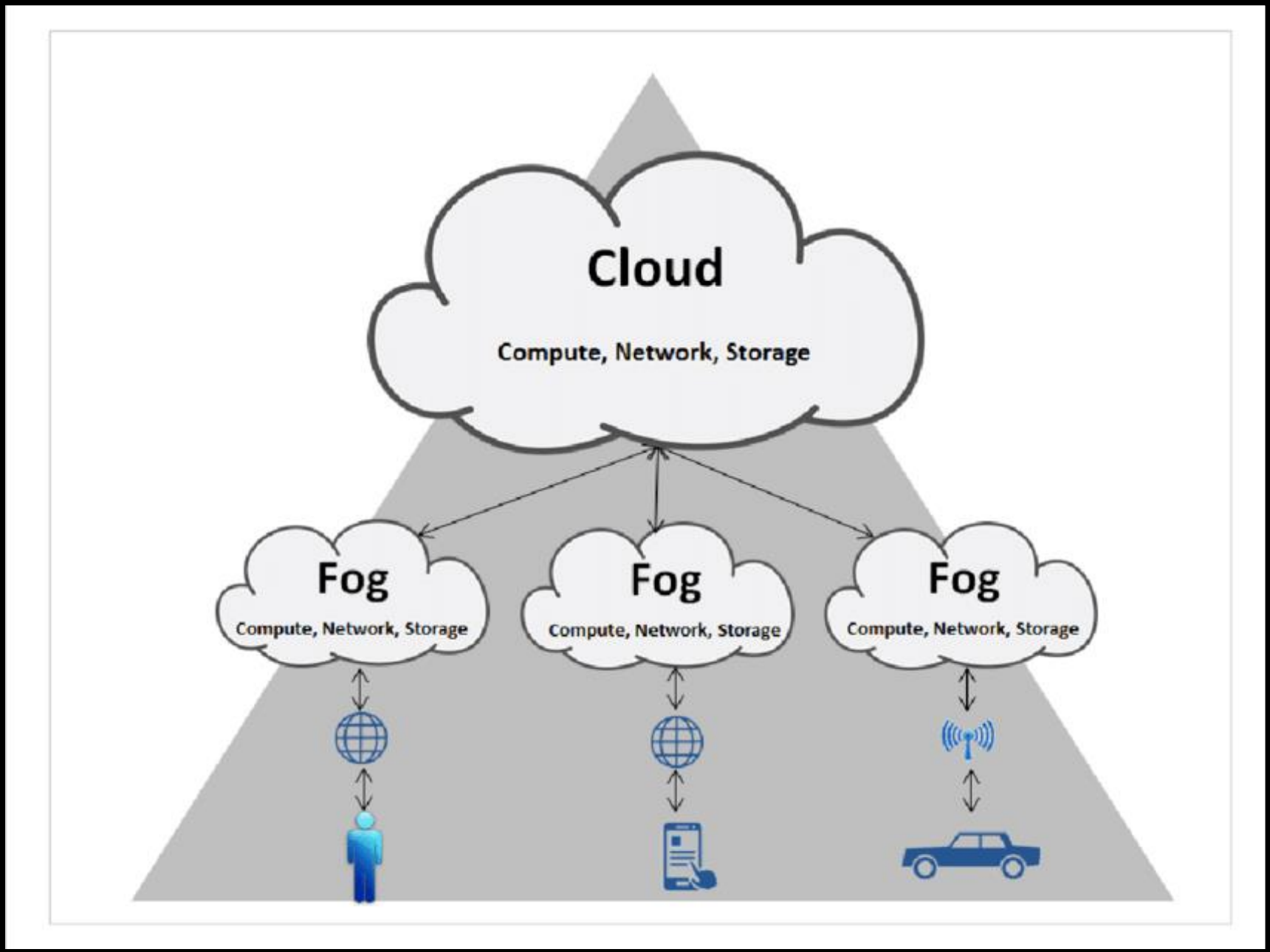
- Fog Computing is an extension of cloud computing that aims to keep the same features of Cloud, such as networking, computation, virtualization, and storage, but also meets the requirements of applications that demand low latency, specific QoS requirements, Service Level Agreement (SLA) considerations, or any combination of these .
- 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.

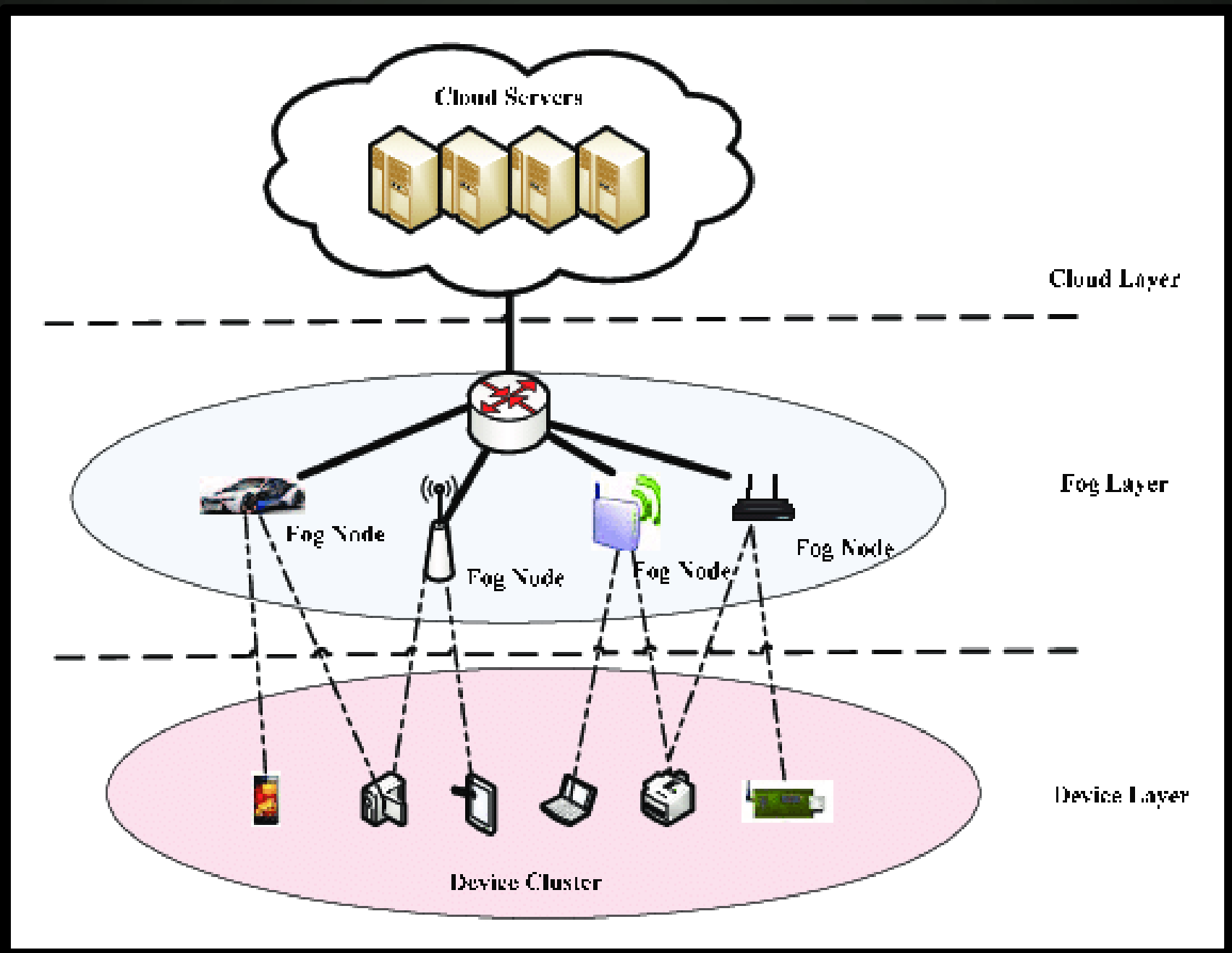
Cloud Versus Fog

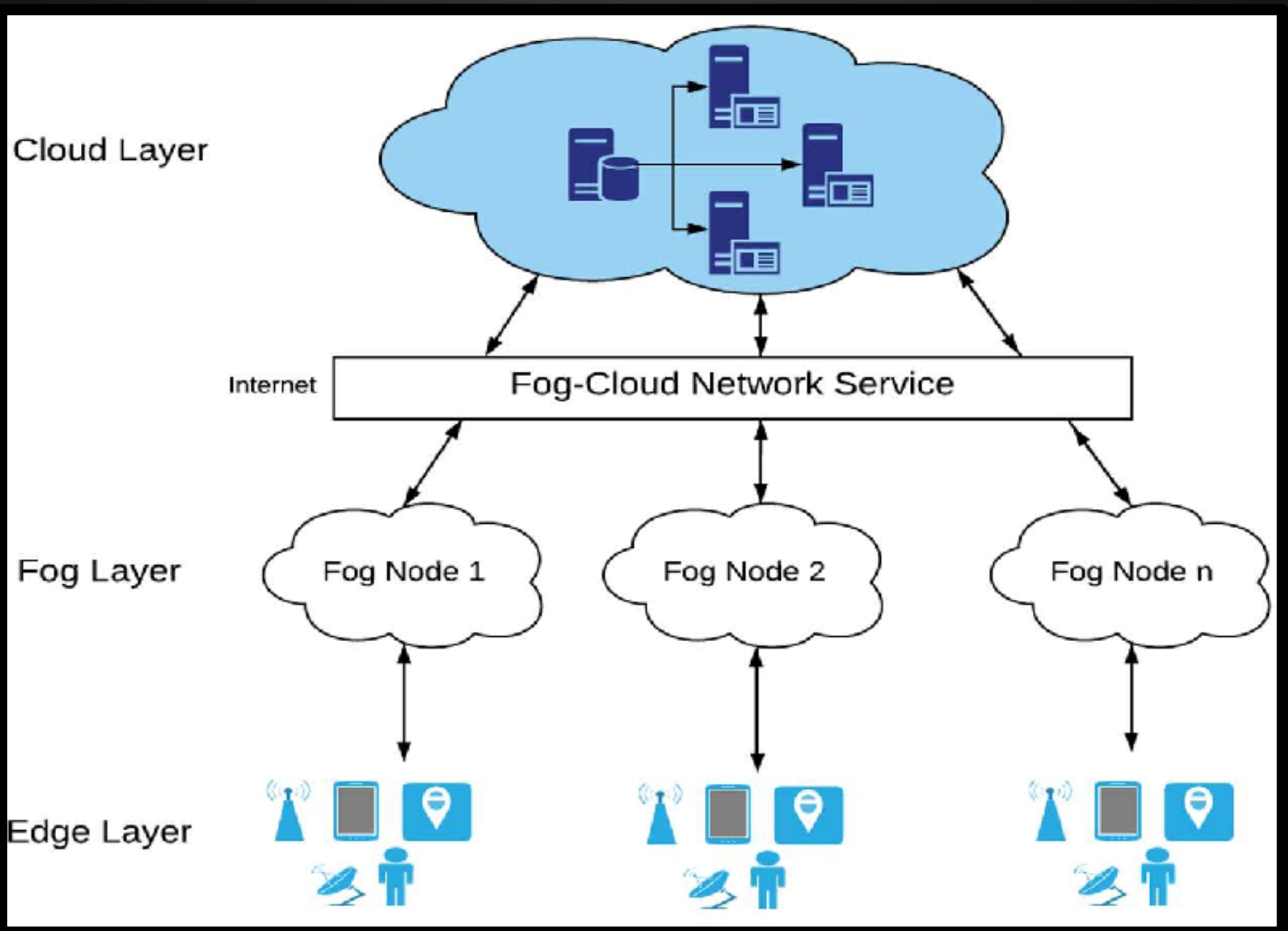
	Fog	Cloud
Response time	Low	High
Availability	Low	High
Security level	Medium to hard	Easy to medium
Service focus	Edge devices	Network/enterprise core services
Cost for each device	Low	High
Dominant architecture	Distributed	Central/distributed
Main content generator—consumer	Smart devices—humans and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



Typical Fog Computing Architecture


Cloud
Compute, Network, Storage
Fog
Compute, Network, Storage
Fog
Compute, Network, Storage
Fog
Compute, Network, Storage


Cloud Servers
Cloud Layer
Fog Node
Fog Node
Fog Node
Fog Node
Fog Layer
Device Cluster
Device Layer

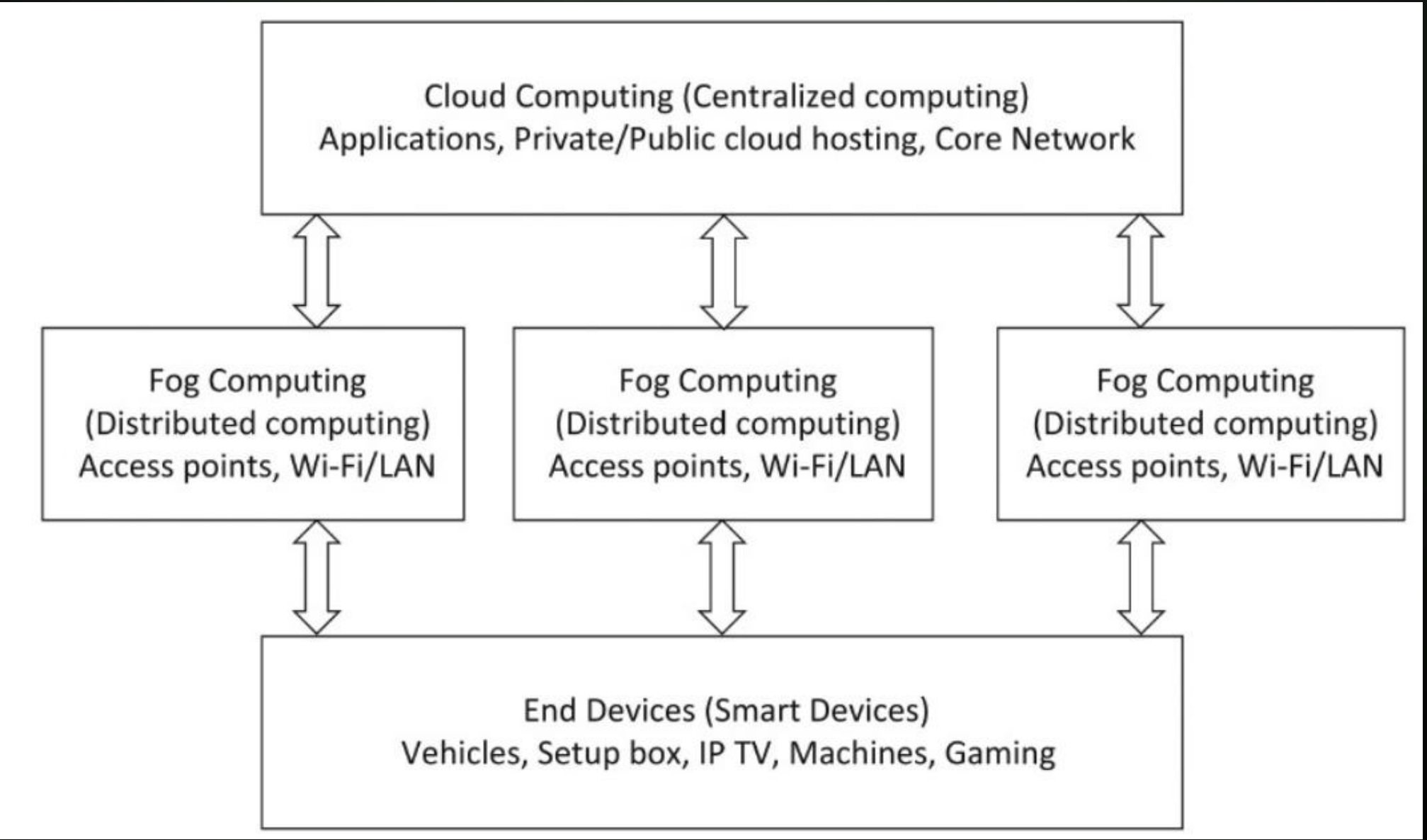

Cloud Layer
Internet
Fog-Cloud Network Service
Fog Layer
Fog Node 1
Fog Node 2
Fog Node n
Edge Layer

CLOUD Vs FOG COMPUTING

Table 1.1 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 in IoT.

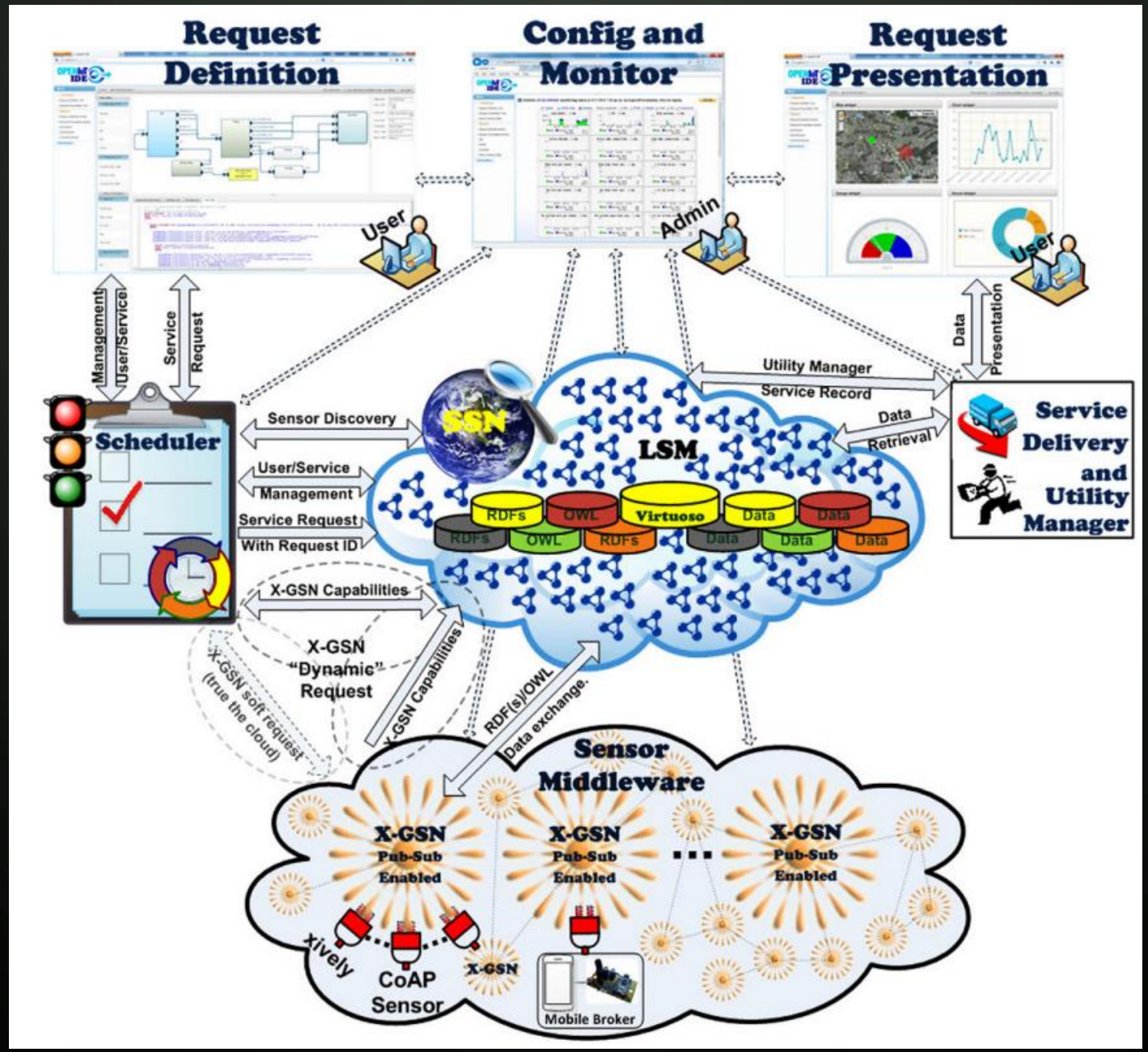
(p 15)

INTERNET OF THINGS APPLICATIONS

- Applications usually fall into 3-categories:

1) **Monitoring And Actuating,**
2) **Business Process And Data Analysis, And**
3) **Information Gathering And Collaborative Consumption.**

Open Iot Architecture For IoT/Cloud Convergence



Open Iot Architecture For IoT/Cloud Convergence

- The figure illustrates the main elements of the OpenIoT software architecture along with their interactions and functionalities.
- **Main Elements:**
  - The Sensor Middleware
  - The Cloud Computing Infrastructure,
  - The Directory Service
  - The Global Scheduler
  - The Local Scheduler component
  - The Service Delivery and Utility Manager
  - The Request Definition tool
  - The Request Presentation component
  - The Configuration and Monitoring component

Open Iot Architecture For IoT/Cloud Convergence

The Sensor Middleware

- Collects, filters, and combines data streams stemming from virtual sensors (eg: signal-processing algorithms, information- fusion algorithms, and social-media data streams) or physical-sensing devices (such as temperature sensors, humidity sensors, and weather stations).
- This middleware acts as a hub between the Open IoT platform and the physical world, as it enables the access to information stemming from the real world.
- It facilitates the interface to a variety of physical and virtual sensors, data streams from other IoT platforms and social networks.

Open Iot Architecture For IoT/Cloud Convergence

The Sensor Middleware

- Main characteristics of the sensor middleware is:
  - Its ability to stream sensor data in the cloud, according to semantic format (ie, ontology).
  - The Sensor Middleware is deployed on the basis of one or more distributed instances (nodes), which may belong to different administrative entities.
  - The prototype implementation of the OpenIoT platform uses the GSN middleware.

    GSN→ Global Sensor Networks

Open Iot Architecture For IoT/Cloud Convergence

The Cloud Computing Infrastructure

- Enables the storage of data streams stemming from the sensor middleware, thereby acting as a cloud database.
- The cloud infrastructure also stores metadata for the various services, as part of the scheduling process.
- In addition to data streams and metadata, computational (software) components of the platform could also be deployed in the cloud in order to benefit from its elasticity, scalability, and performance characteristics.
- The cloud infrastructure could be either a public infrastructure (Amazon Elastic Compute Cloud (EC2)) or a private infrastructure (a private cloud deployed, based on Open Stack).
- The cloud infrastructure can be characterized as a sensor cloud, given that it primarily supports storage and management of sensor data-streams (and of their metadata).

- **Private cloud** is cloud computing →dedicated solely to one organization.

Open Iot Architecture For IoT/Cloud Convergence

The Directory Service

- Stores information about all the sensors that are available in the Open IoT platform.
- It also provides the means for registering sensors with the directory,
- Also for the discovery of sensors.
- The IoT/cloud architecture specifies the use of semantically annotated descriptions of sensors as part of its directory service.
- The Directory Service is deployed within the cloud infrastructure, thereby providing the means for accessing sensor data and metadata residing in the cloud.

Open Iot Architecture For IoT/Cloud Convergence

The Global Scheduler

- Processes all the requests for on-demand deployment of services, and ensures their proper access to the resources (eg, data streams) that they require.
- This component undertakes the task of parsing the service request, and discovering the sensors that can contribute to its fulfilment.
- It also selects the resources(sensors) that will support the service deployment, while also performing the relevant reservations of resources.
- This component enables the scheduling of all IoT services.

Open Iot Architecture For IoT/Cloud Convergence

The Local Scheduler component

- It is executed at the level of the Sensor Middleware, and ensures the optimized access to the resources managed by sensor middleware instances, whereas the 'Global Scheduler' regulates the access to the resources of the 'Open IoT platform' ;
- it regulates the access and use of the data streams at the lower level of the Sensor Middleware.

The Service Delivery and Utility Manager

- It performs a dual role

i. It combines the data streams as indicated by service workflows within the Open IoT system, in order to deliver the requested service. To this end, this component makes use of the service description and the resources identified and reserved by the scheduler component.

ii. It acts as a service-metering facility, which keeps track of utility metrics for each individual service. This metering functionality is accordingly used to drive functionalities such as accounting, billing, and utility-driven resource optimization. Such functionalities are essential in the scope of "pay-as-you-go" computing paradigm .

Open Iot Architecture For IoT/Cloud Convergence

The Request Definition tool

- It enables the specification of service requests to the Open IoT platform.
- It comprises a set of services for specifying and formulating such requests, while also submitting them to the Global Scheduler.
- This tool features a Graphical User Interface (GUI).

Open Iot Architecture For IoT/Cloud Convergence

The Request Presentation component

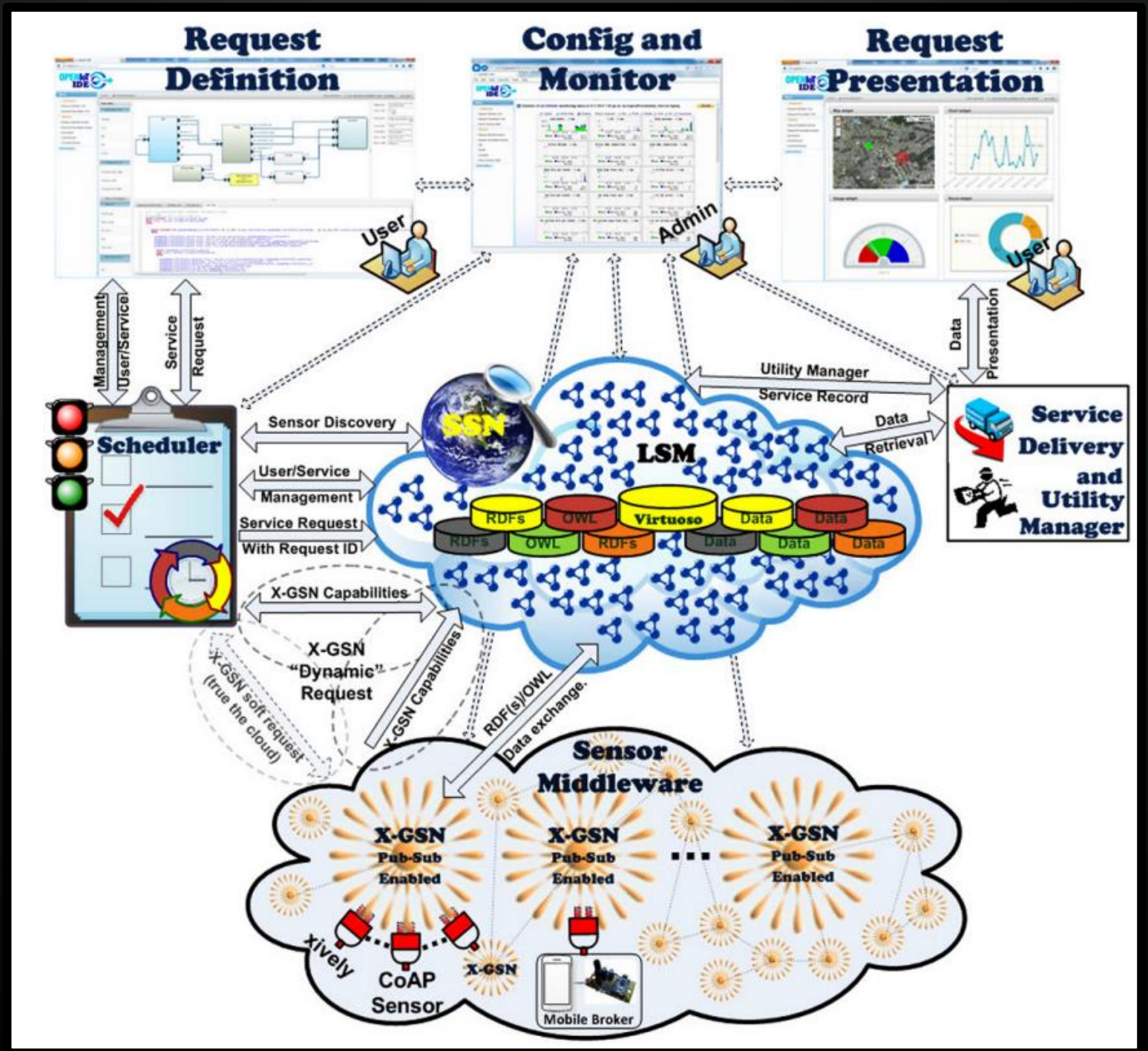
- It is in charge of the visualization of the outputs of an IoT service.
- This component takes a few combinations of services and libraries in order to facilitate service presentation.

Open Iot Architecture For IoT/Cloud Convergence

The Configuration and Monitoring component:

- Enables management and configuration functionalities over the sensors, and the IoT services that are deployed within the platform.
- This component is also supported by a GUI.

Open Iot Architecture For IoT/Cloud Convergence



Open Iot Architecture For IoT/Cloud Convergence

Typical Workflow Associated with the Use Of Open IoT Platform:

i. The **formulation of a request for an IoT service** using the Request Definition tool, and its submission to the (Global) Scheduler component.

ii. The **Parsing of the IoT service request by the scheduler**, and the subsequent discovery of the sensors/ICOs to be used in order to deliver the IoT service. Toward discovering the required sensors, the Directory Service is queried and accessed.

iii. The **Formulation of the service** and its persistence in the cloud, along with other metadata about the service.

iv. The **Execution of the service by end users** and the **visualization** of the results.

SCHEDULING PROCESS AND IOT SERVICES LIFECYCLE

- The Global Scheduler Component is the main entry point for service requests submitted to the cloud platform.
- It parses each service request and accordingly performs two main functions toward the delivery of the service (selection of the sensors, the reservation of the needed resources)
- The scheduler manages all the metadata of the IoT services including:

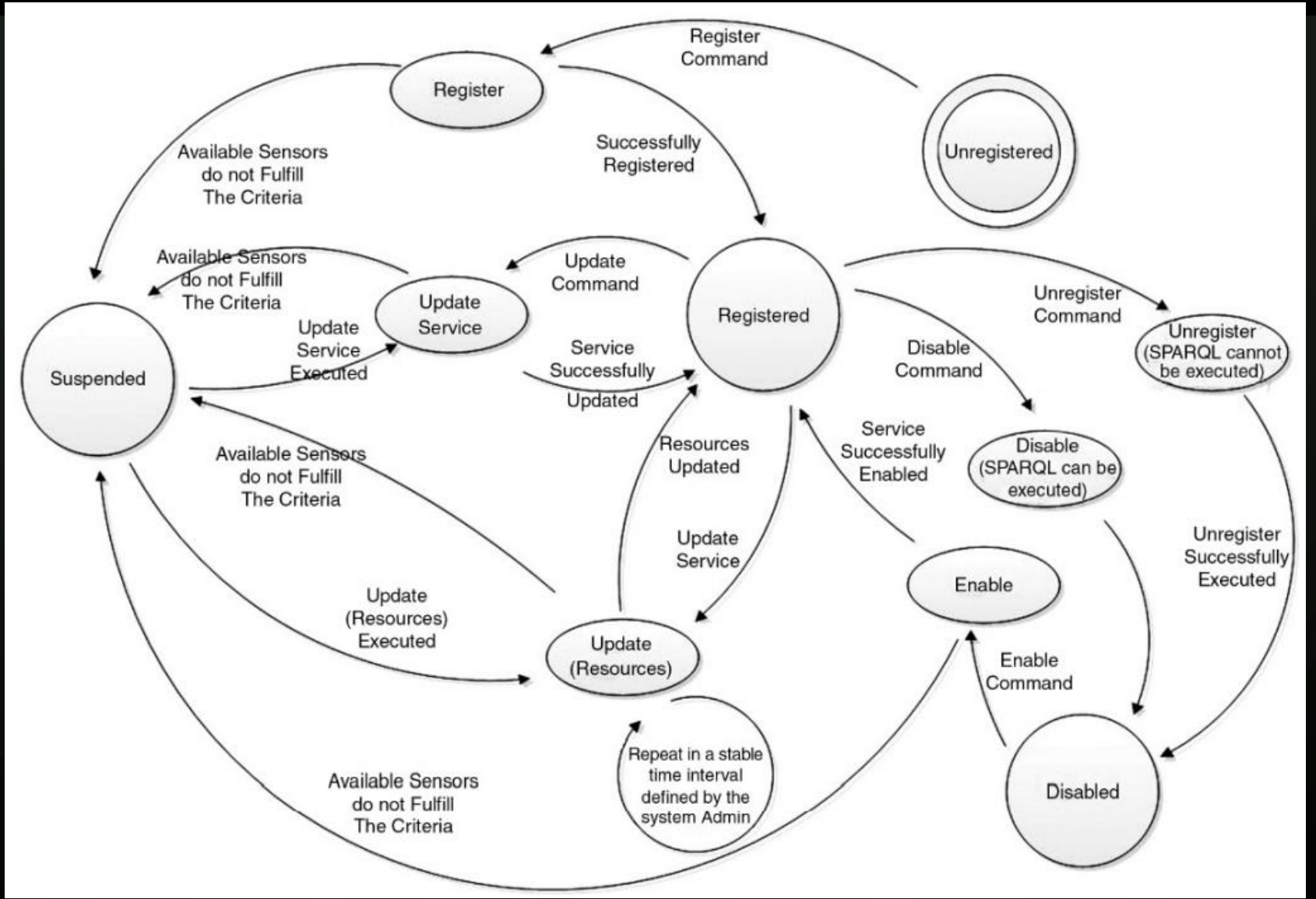
➢ The signature of the service (ie , its input and output parameters),

➢ The sensors used to deliver the service

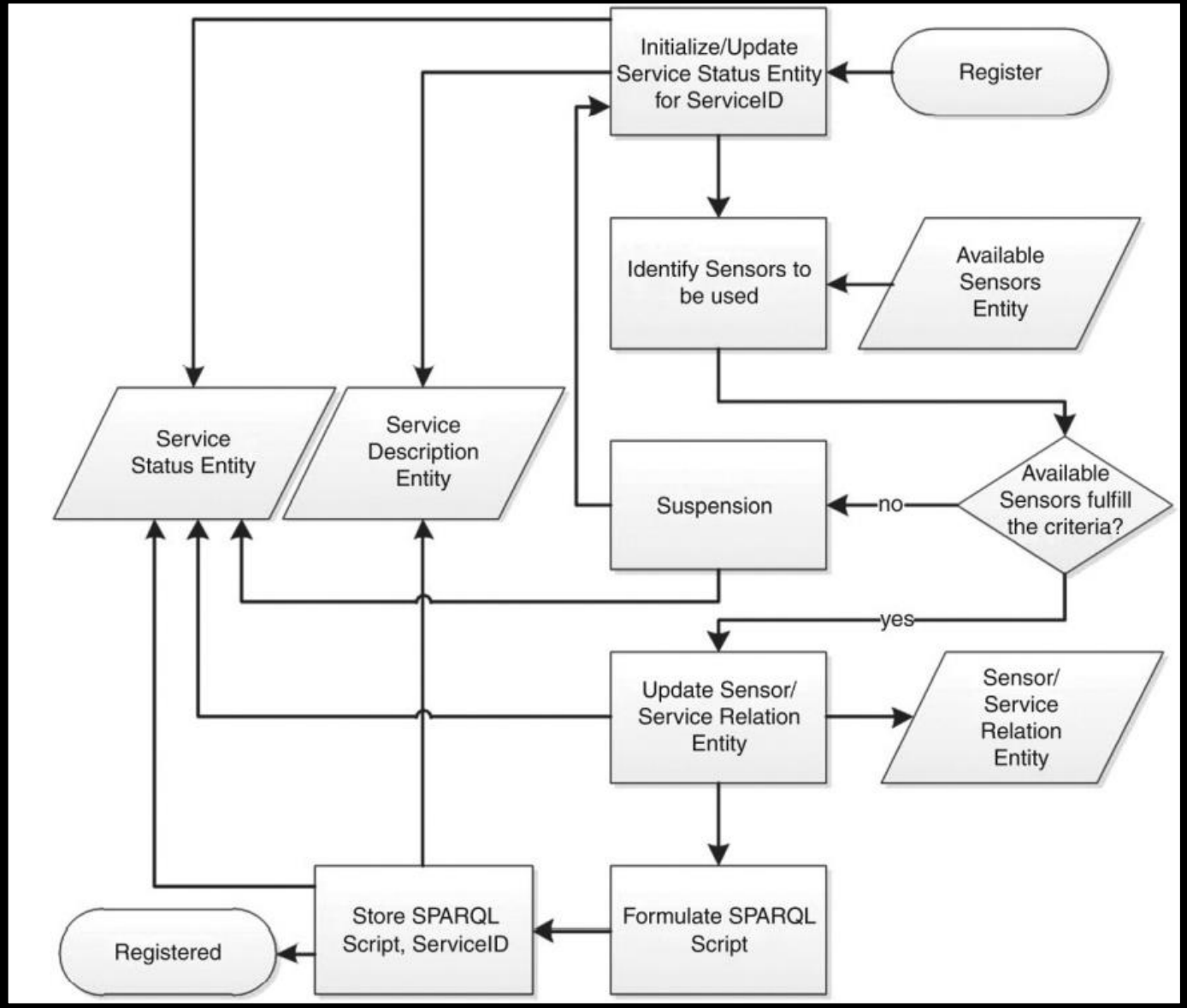
➢ Execution parameters associated with the services(such as the intervals in which the service shall be repeated, the types of visualization (in the request presentation), and other resources used by the service.)

State Diagram Of The OpenIoT Services Lifecycle Within The Scheduler Module

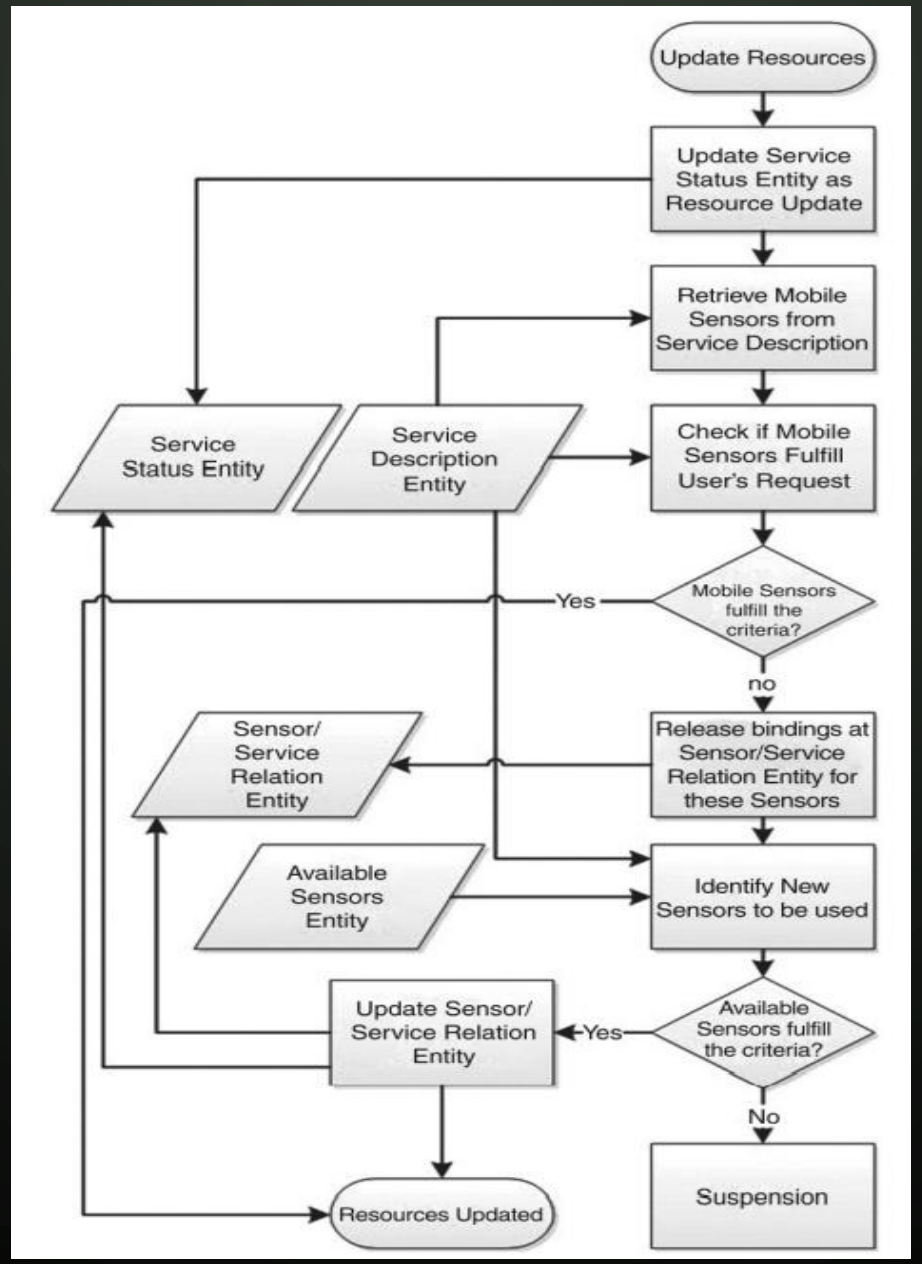
Fig: lifecycle management services are supported by the scheduler:



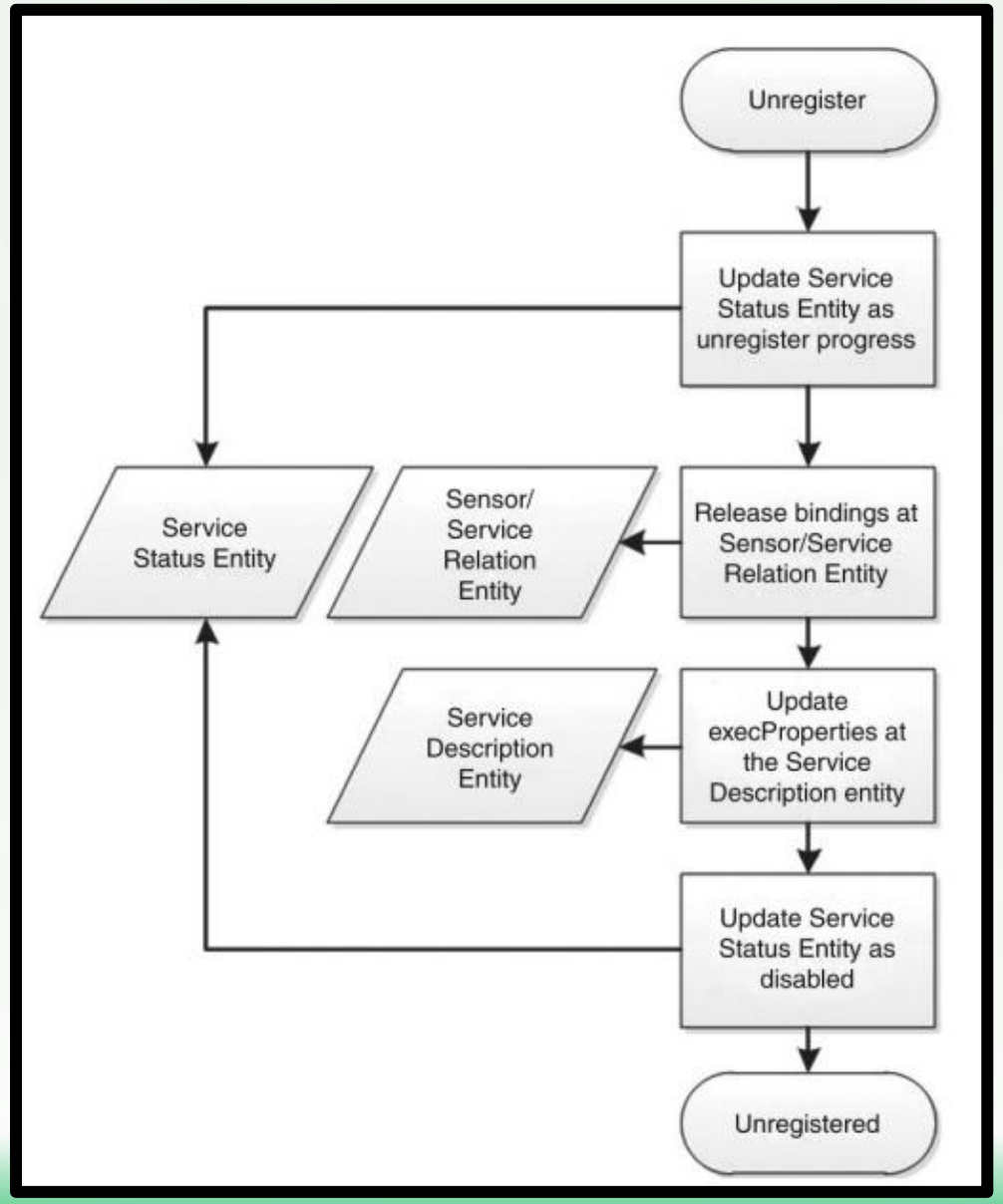
“REGISTER SERVICE” PROCESS FLOWCHART



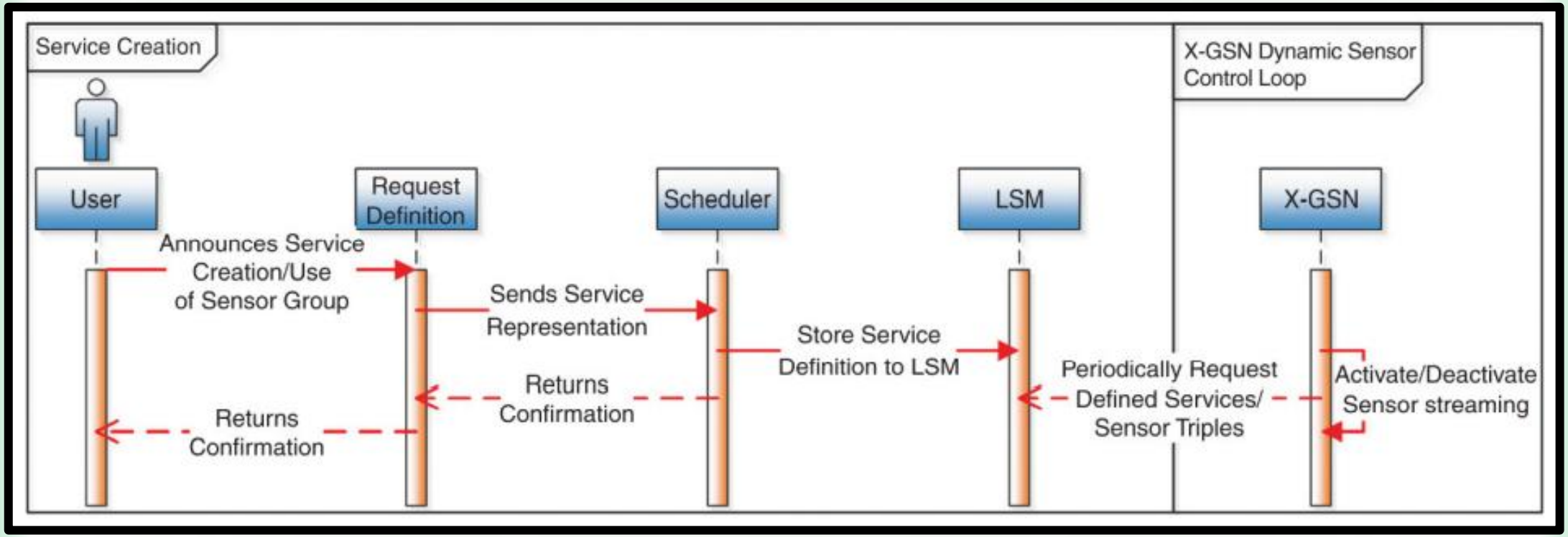
“UPDATE RESOURCES” SERVICE FLOWCHART



Unregister Service” Service Flowchart



“SERVICE CREATION” SERVICE FLOWCHART



- SSN → Sematic Sensor Network

- ****

Context-aware Proactive Suggestion

- An application that makes context-aware recommendations.
- Based on the personal data collected on each mobile device, There is devised Proactive Suggestion(PS).
-

Fig.: Individual components of the PS

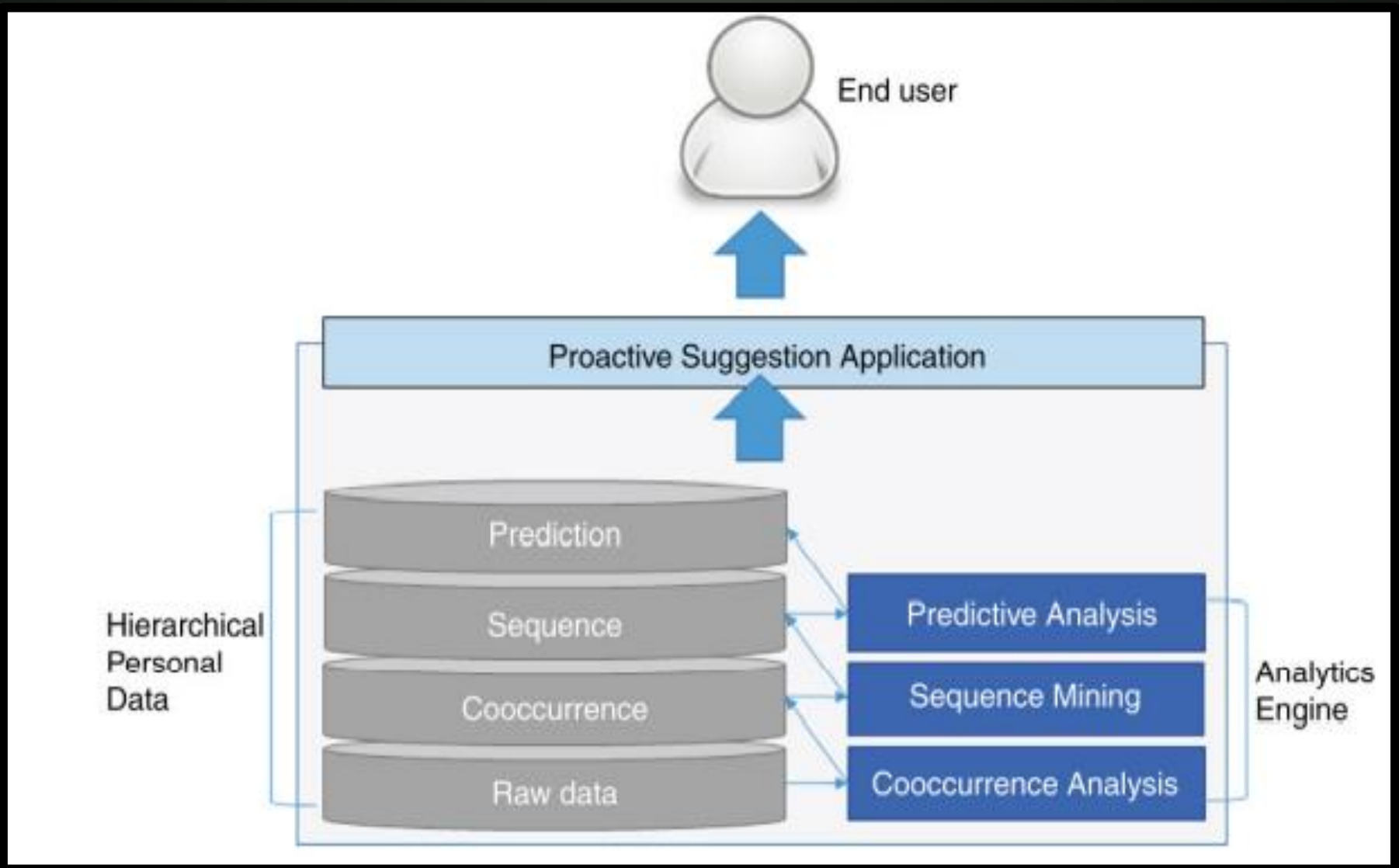
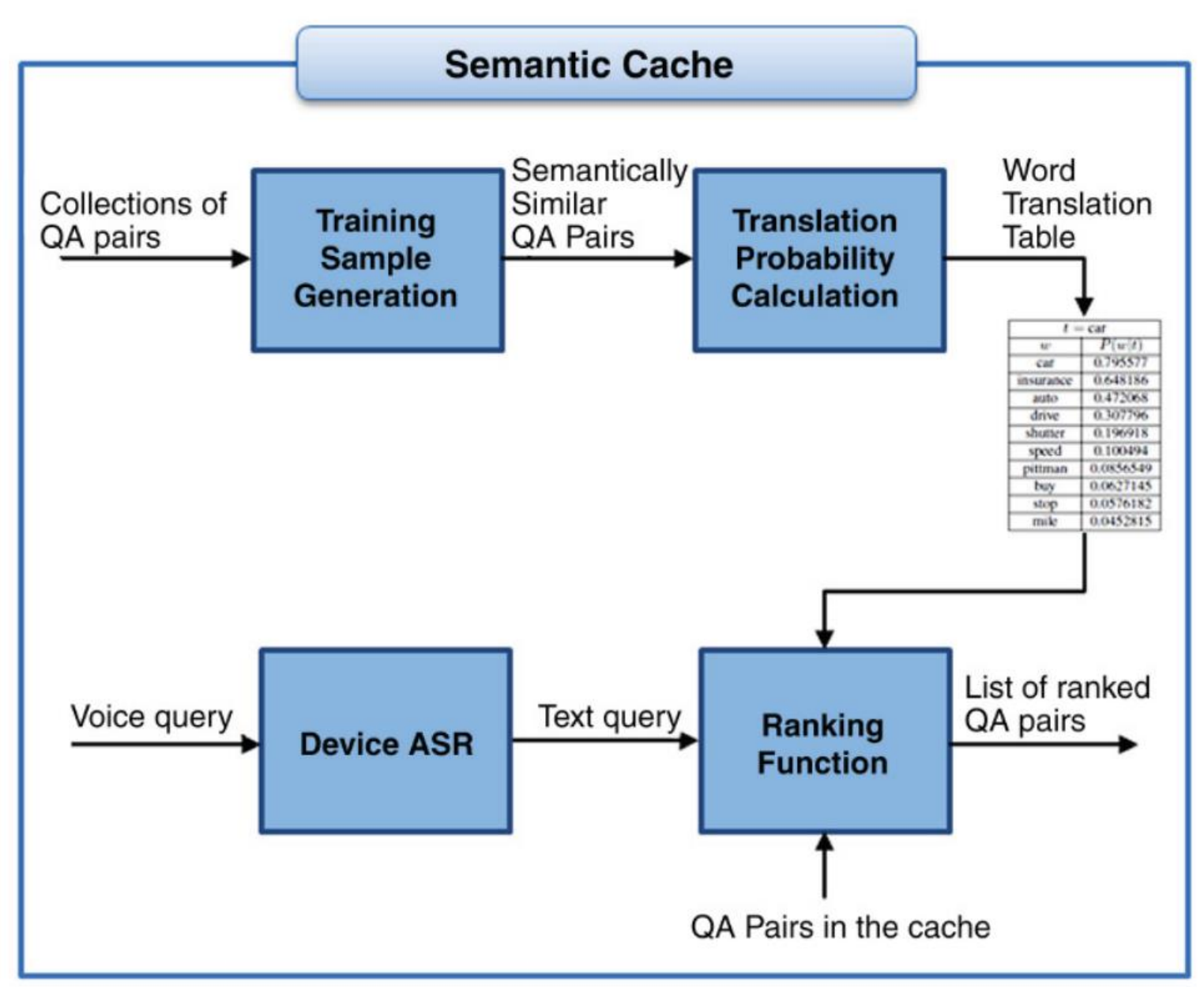
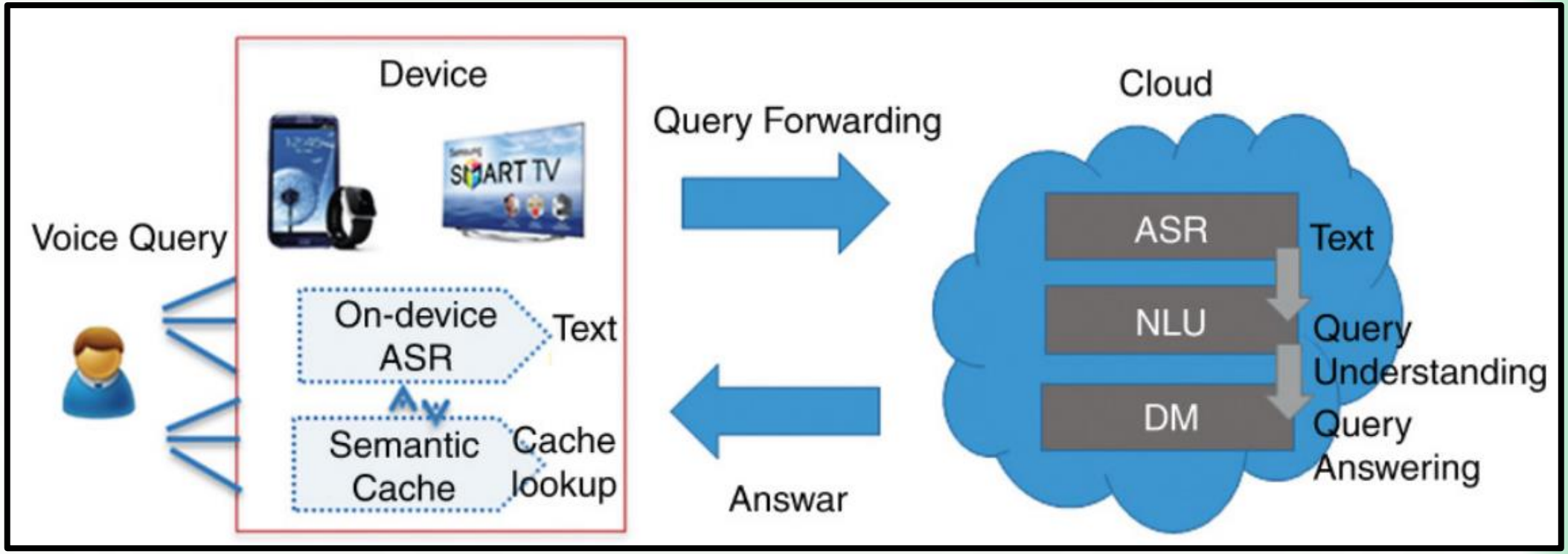


Fig. Illustrates how the Semantic QA cache is managed.



Semantic QA Cache Implementing The Device/Cloud Collaboration Framework



ASR → (Automatic Speech Recognition)

NLU → a Natural Language Understanding module

DM (Dialog Manager)