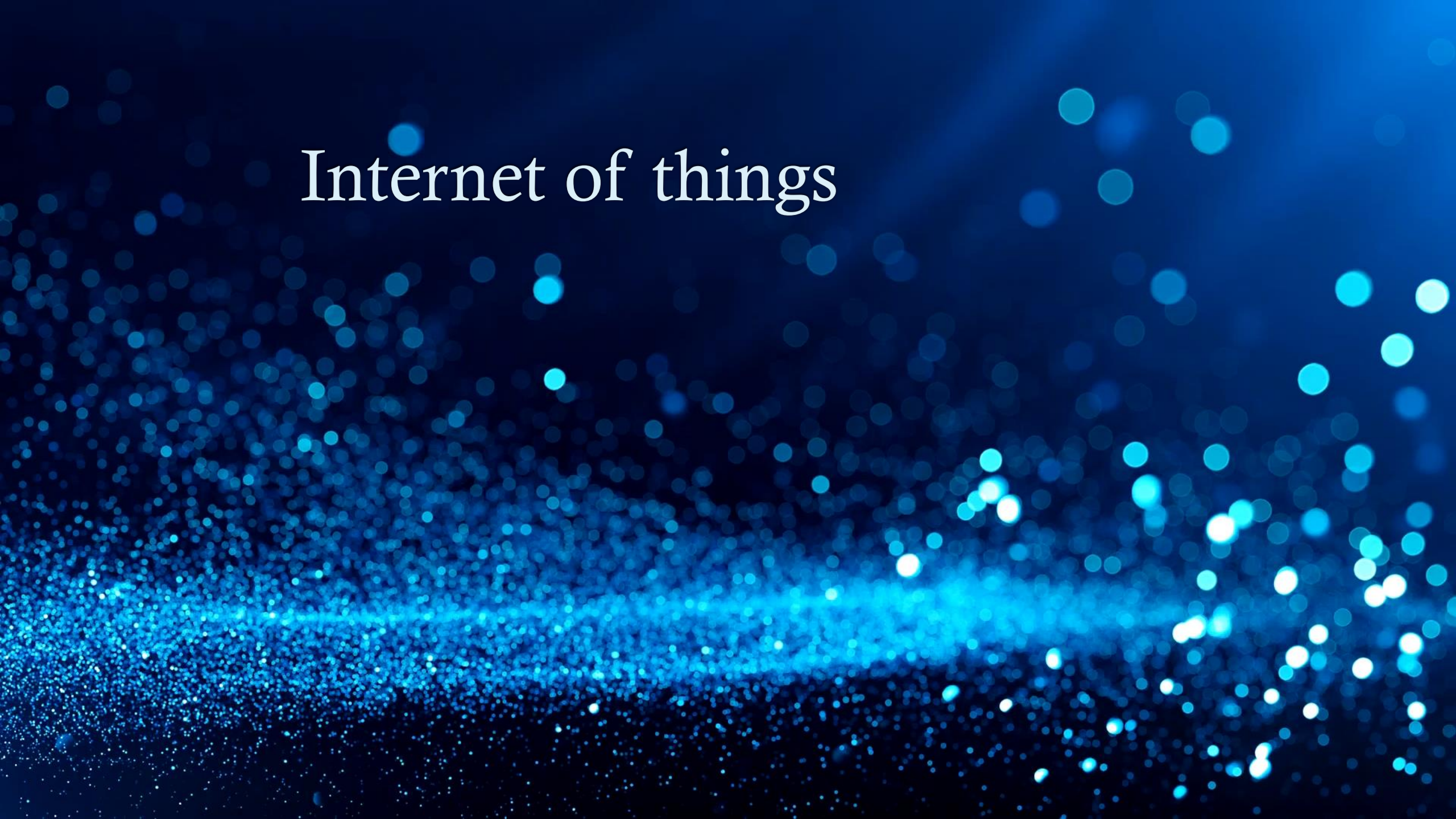
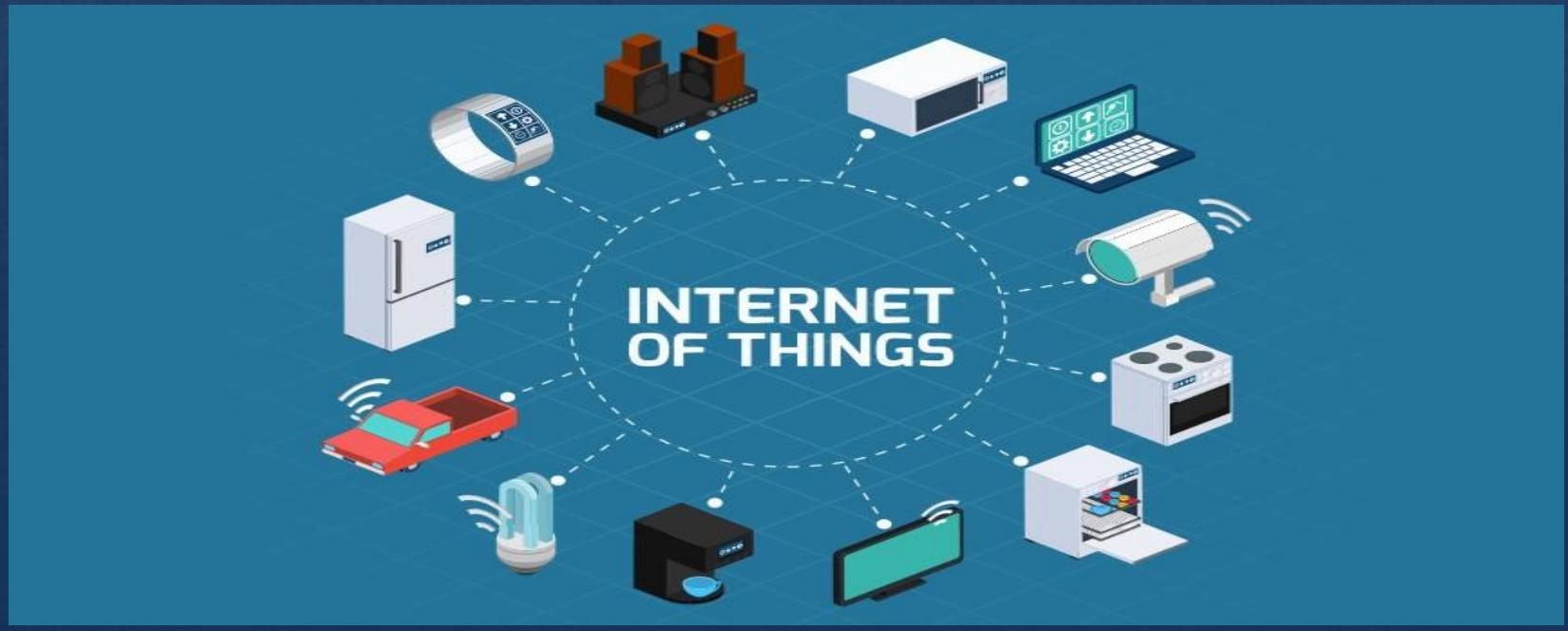
# Internet of things



# INTRODUCTION

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.



# How does IoT work

- An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally
- The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.
- IoT can also make use of artificial intelligence (AI) and machine learning to aid in making data collecting processes easier and more dynamic.

# Applications of IoT

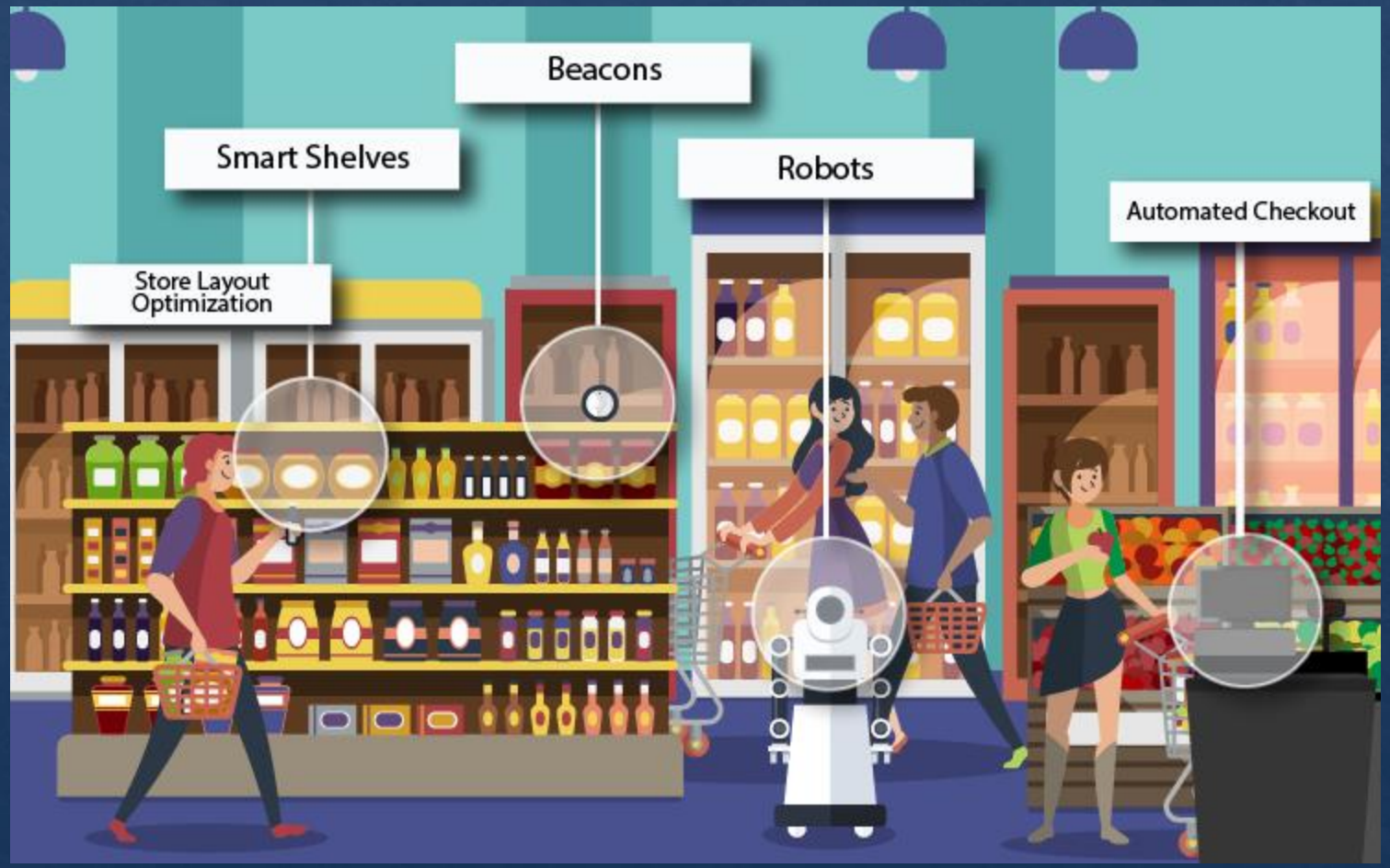
1. Smart Homes
2. Smart City
3. Self-driven Cars
4. IoT Retail Shops
5. Farming
6. Wearables
7. Smart Grids
8. Industrial Internet
9. Telehealth
10. Smart Supply-chain Management,  etc….


SMART HOME
iStock
by Getty Images
518174636


Smart government
Mobility/ Wi-Fi
Open data
Smart health
Smart farming/ agriculture
Smart grid/ energy/ utilities
Smart manufacturing
Smart/ digital citizens
Smart buildings
Smart transportation
SMART CITY COMPONENTS

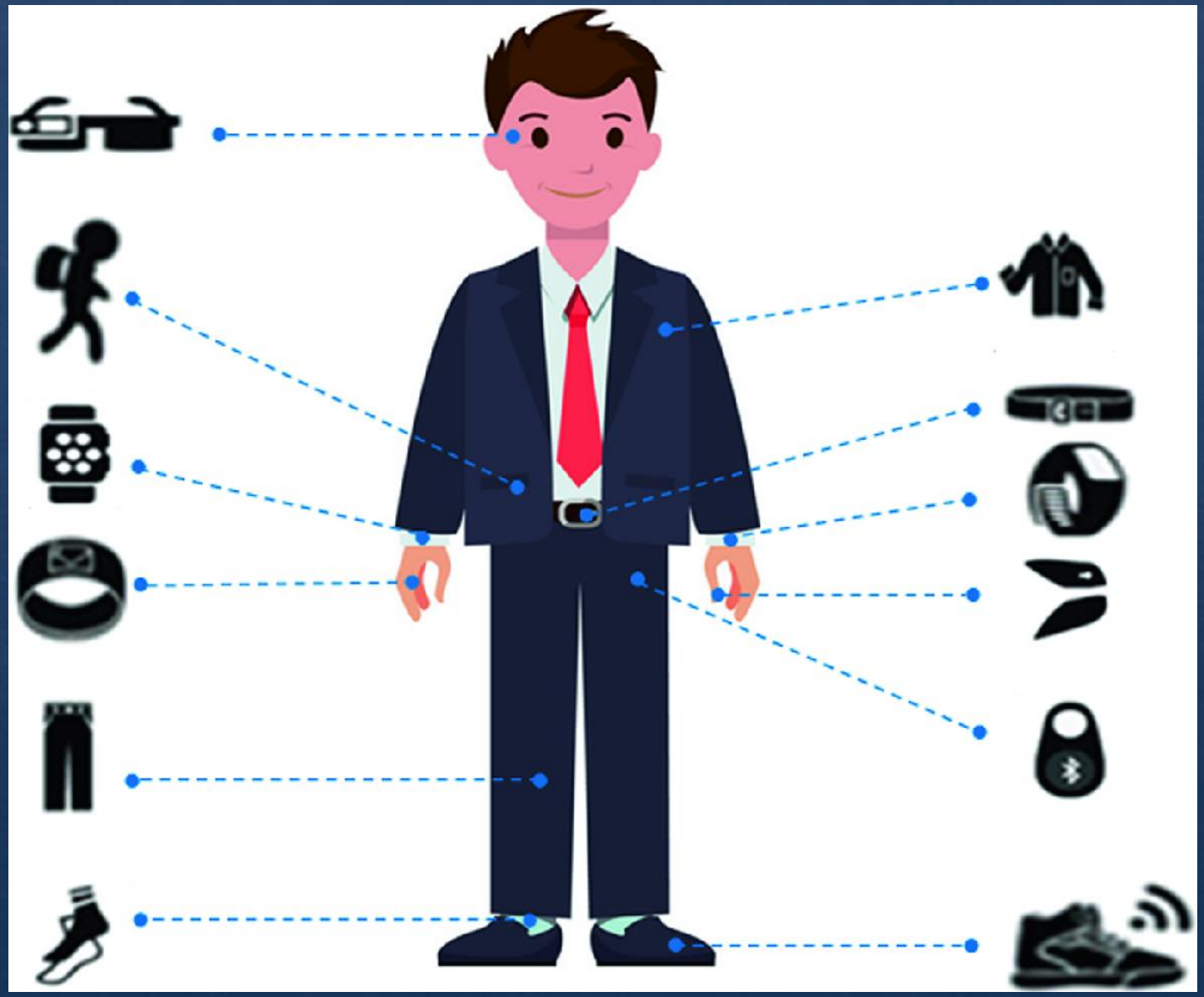
# SELF DRIVEN CARS

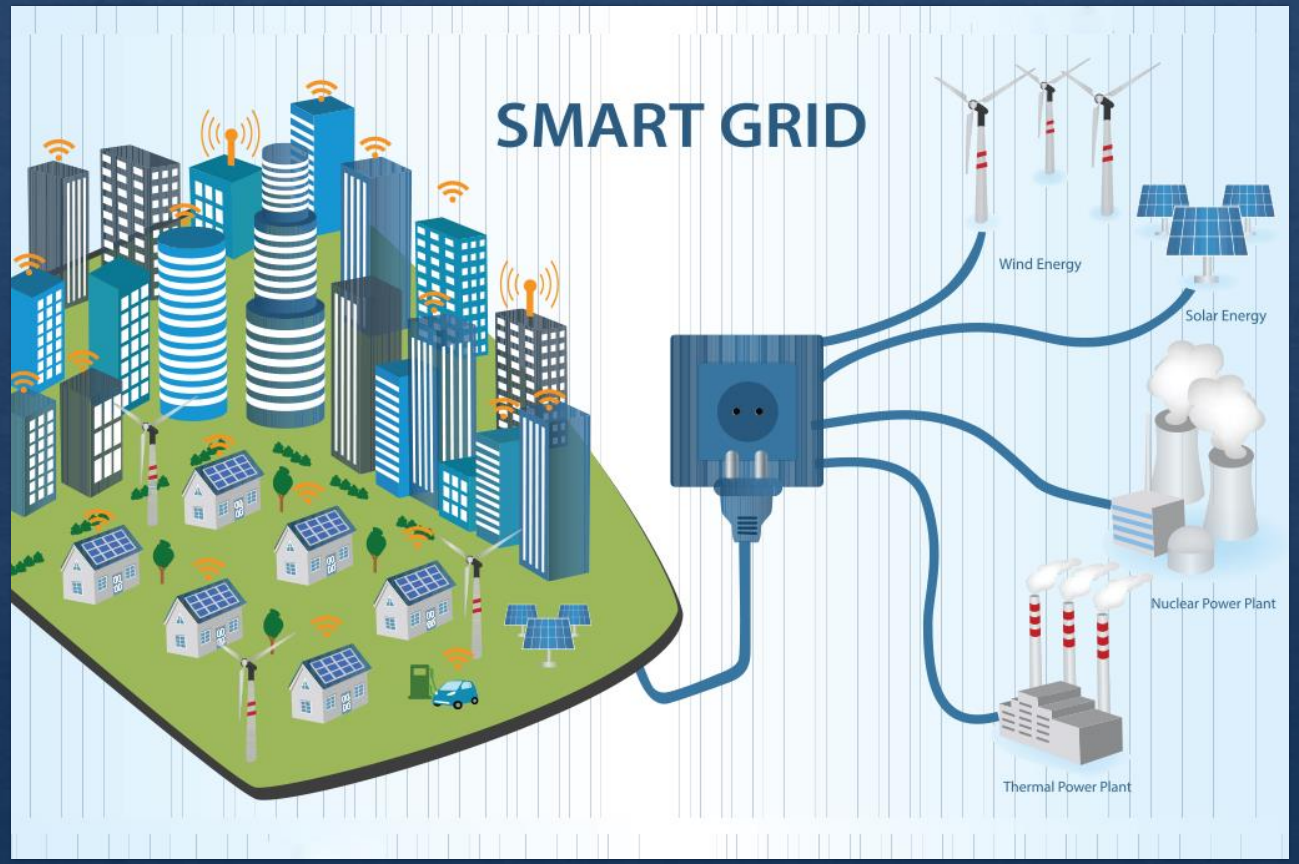
# IOT RETAIL SHOPS




IOT in Agriculture
DreamzTech Solutions
Diagnosis of Diseases
Variable rate of Fertility
Auto Spreading
Water Stress
Crop yield Analysis
Smart Data
Soil Erosion
Field Monitoring
SAMSUNG
Wednesday:
Hi: 90°F
Lo: 72°F
Thursday:
83°F
So far today:
Hi: 88°F Lo: 71°F
Rain: 0"
Gust: 6mph S
Heat Index: 89°F
Humidity: 72%
Dew Point: 73°F

# IOT WEARABLES




SMART GRID
Wind Energy
Solar Energy
Nuclear Power Plant
Thermal Power Plant

# INDUSTRY 4.0

# TELE HEALTH

# SMART SUPPLY CHAIN

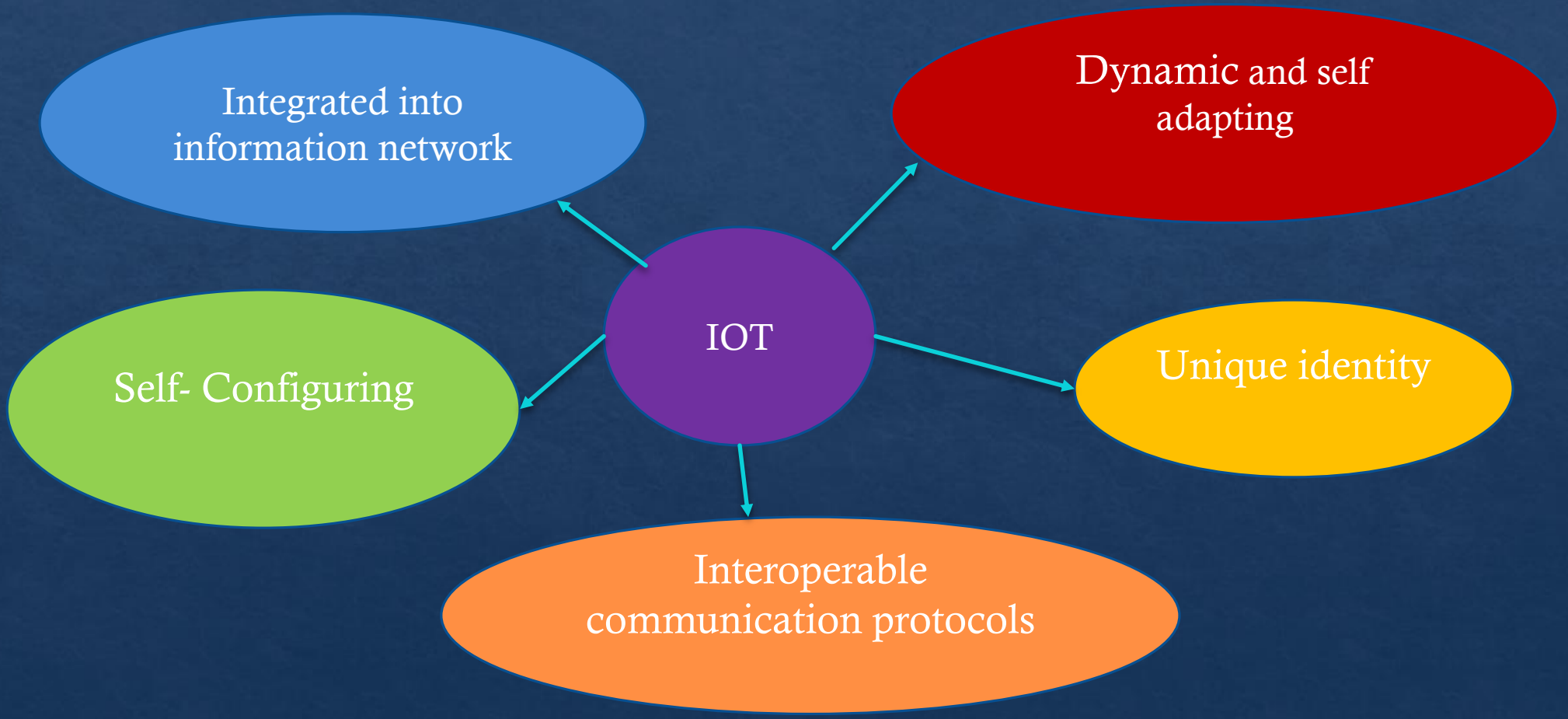
# Advantages and disadvantages

| Advantages | Disadvantages |
| --- | --- |
| Minimizes the human work and effort | Increased privacy concerns |
| Saves time and effort | Increased unemployment rates |
| Good for personal safety and security | Highly dependent on the internet |
| Useful in traffic and other tracking or monitoring systems | Lack of mental and physical activity by humans leading to health issues. |
| Beneficial for the healthcare industry | Complex system for maintenance |
| Improved security in homes and offices | Lack of security |
| Reduced use of many electronic devices as one device does the job of a lot of other devices | Absence of international standards for better communication |

# Things in IoT

- The things in IoT are the devices which have unique identities and can perform remote sensing actuating and monitoring capabilities .
- IoT devices can be of varied types. Almost all devices generate data in some form or other which then processed by a data analytics systems leads to useful information to guide further actions locally or remotely.

# Characteristics of IoT devices



# Unique identity

- Each IoT device has a unique identity and a unique identifier ,such as an IP address or URI.

# Integrated into network

- IoT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems.
- IoT devices can be dynamically discovered in the network, by other devices or/and the network and have the capability to describe themselves to other devices or user applications

# Dynamic and self adapting

- IoT devices and systems may have the ability to dynamically adapt with the changing context and take actions based on their operating conditions or sensed environment.

# Self-configuring

- IoT devices may have self configuring capability allowing a large number of devices to work together to provide certain functionality .
- They can configure themselves ,setup the net working, and fetch latest software upgrades with minimal manual or user interventions.

# Interoperable communication protocols

- IoT can support a number of communication protocols and can communicate with other devices and also with the infrastructure.

# Internet of Things

# Components in IoT

- Sensors and actuators
- Device  -(thing +  sensors)
- Gateway(Device + External gateway(+Internet)
- Protocols
- Cloud

# Sensor

- Sensors are devices that detect the feature quantity of a measurement object and convert this quantity(physical) into a readable signal(electrical).
- Temperature Sensors.
- Humidity Sensors.
- Pressure Sensors.
- Proximity Sensors.
- Level Sensors.
- Accelerometers.
- Gyroscope.
- Gas Sensors.

# Actuators

- An actuator is a machine component or system that moves or controls the mechanism or the system. It takes an electrical input and turns it into physical action.
- Hydraulic Actuators.
- Pneumatic Actuators. .
- Electrical Actuators.
- Thermal Actuators.
- Magnetic Actuators.
- Relay Actuators.

# IoT Device

- An IoT device is made up of a Physical object (“thing”) + Controller (“brain”) + Sensors + Actuators + Networks (Internet)
- IoT devices are **pieces of hardware**, such as sensors, actuators, gadgets, appliances, or machines, that are programmed for certain applications and can transmit data over the internet or other networks. Examples
- Arduino with Arduino Ethernet connection
- Raspberry Pi connected via Ethernet or Wi-Fi
- Intel Galileo connected via Ethernet or Wi-Fi Examples of indirectly connected device include

# Gateway

- An IoT gateway is a physical device or virtual platform that connects sensors, IoT modules, and smart devices to the cloud. Gateways serve as a wireless access portal to give IoT devices access to the Internet.

# Types of communication in IoT

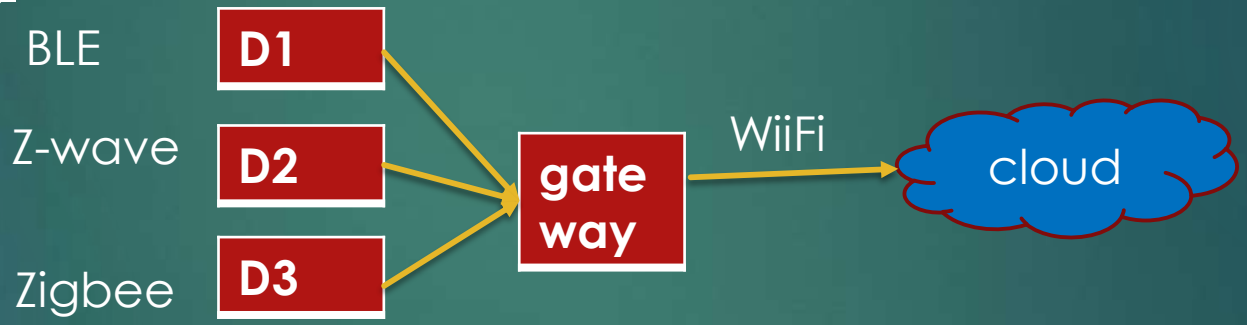
- Device to device


BLE, Z-wave, Zigbee

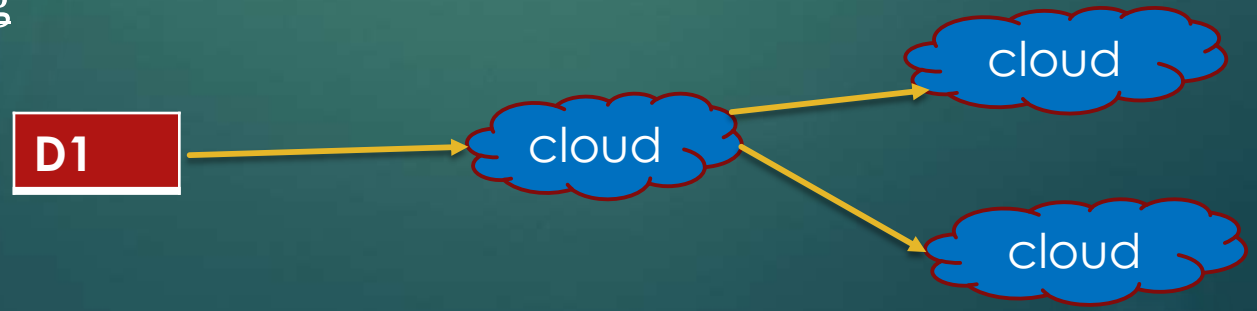
- Device to cloud


WiFi , Ethernet

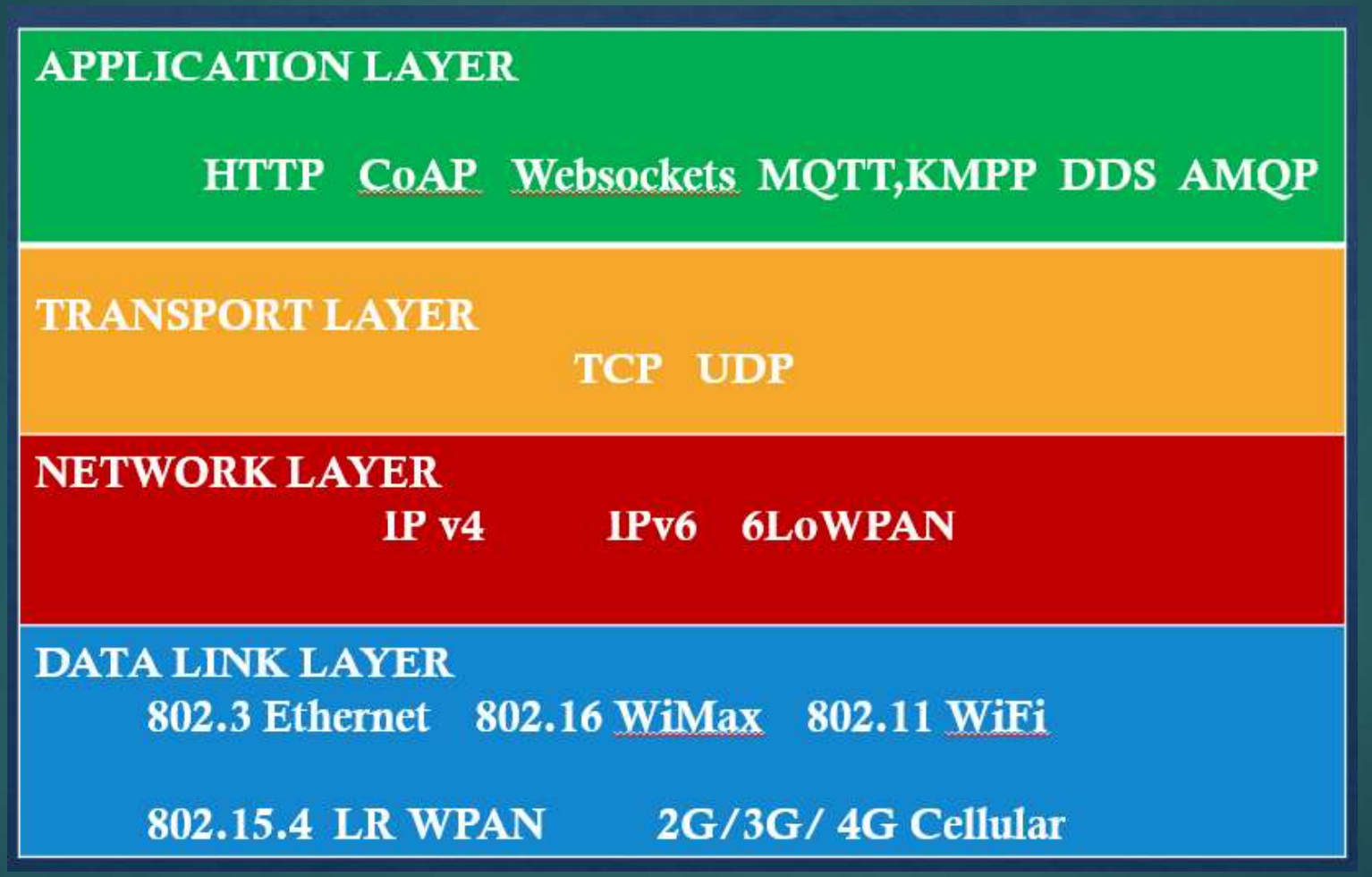
- Device to gateway




- Backend data sharing



# Communication protocols



# Link layer protocols

- Link layer : protocols determine how the data is physically sent over the networks physical medium

**802.3 – Ethernet**: Medium can be coaxial cable,twisted pair or optical fiber. Data rates can be 10 Mb/s to 40 Gb/s

**802.11- Wi Fi:** Collection of wireless local area network protocols. Data rates can be 1 Mb/s to 6.75 Gb/s.

**802.16 – Wi Max** : Collection of wireless broadband standards. Data rate 1.5 Mb/s to 1G/s

**802.15.4 – LR –WPAN**: collection of standards for low rate wireless personal area networks. Data rates 40 kb/s to 250 kb/s.Used for low cost ,low speed communication for power constrained devices.

**2G/ 3G/ 4G/5G –Mobile communication**: Communicates over cellular networks. Data rates 9.6kb/s to 100Mb/s

# Network/Internet Layer

- The network layer is responsible for sending IP datagrams from the source network to the destination network. It performs the host addressing and packet routing. Host addressing is done using the following IP addressing scheme.
- IPv4: 32bit addressing scheme
- IPv6: 128 bit addressing scheme
- 6LoWPAN: IPv6 over Low-Power Wireless Personal Area Networks, brings IP protocol to the low power devices which have limited processing capability.

# Transport layer

- The transport layer protocols provide end to end message transfer and functions such as error control segmentation, flow control and congestion control.
- TCP : Transmission Control Protocol connection oriented and stateful protocols. Ensures reliable transmission of packets in order. Used by web browsers, email programs and file transfer. Light weight implementation of TCP is used in IoT.
- UDP: User datagram protocol Connectionless protocol. Transaction oriented and stateless protocol. Used for time sensitive applications.

# Application layer

- Application layer protocols define how the applications interface with the lower layer protocols to send the data over the network. Port numbers are used for application addressing.
- HTTP: foundation of WWW. The protocol follows a request-response model where a client(browser or application) sends a request to server using HTTP commands. It is a stateless protocol. Protocol uses URL to identify resources. Runs over TCP.
- CoAP: Constrained application protocol: Used for machine to machine applications, meant for constrained environments with constrained devices and constrained networks. It uses a client server architecture, request-response model where clients communicate with servers using connectionless datagrams. It runs over UDP.
- WebSocket: allows full duplex communication over a single socket connection for sending messages between client and server while keeping the TCP connection open.

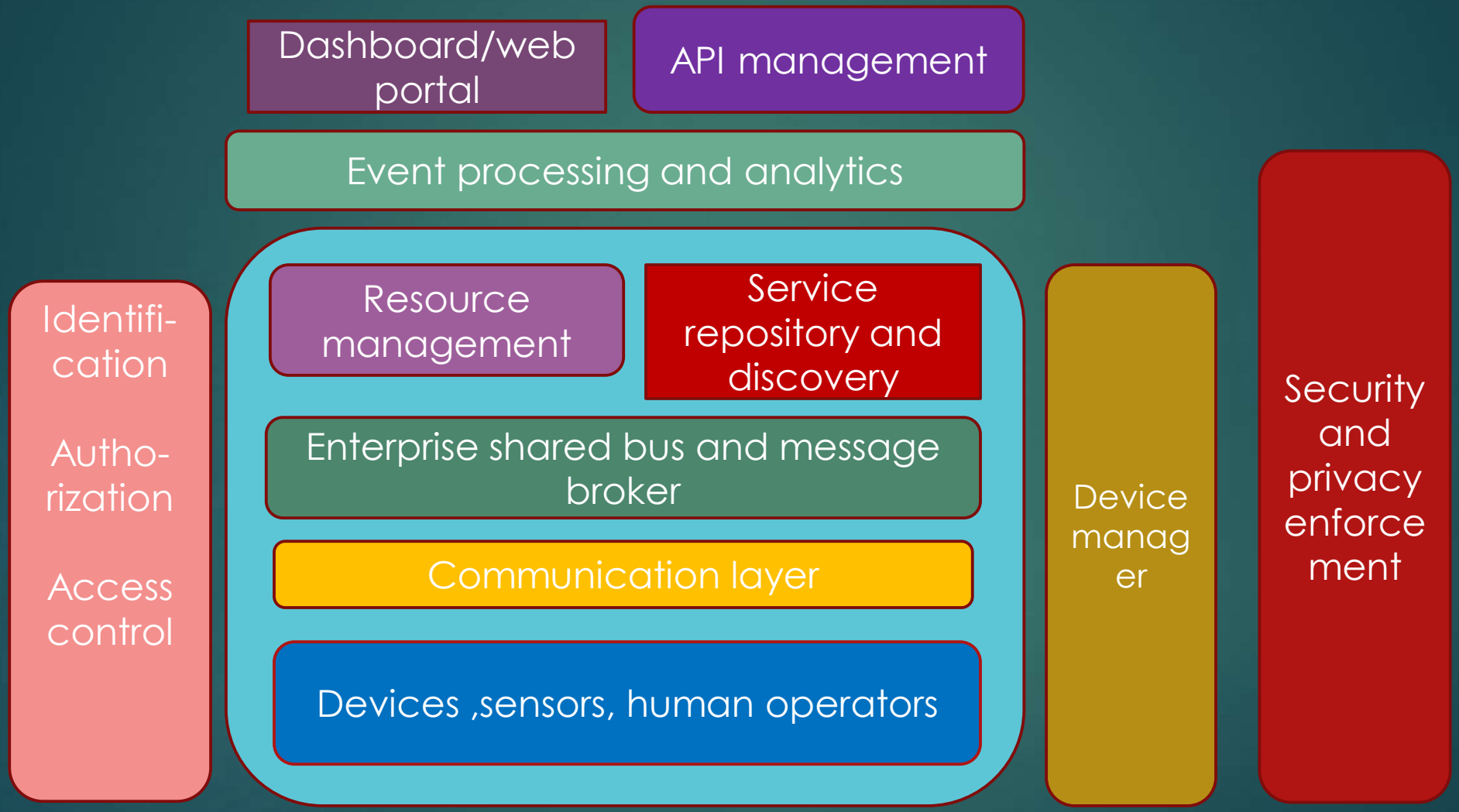
- MQTT: Message Queue Telemetry Transport is a light weight messaging protocol based on publish subscribe model. Uses a client server architecture where the client connects to the server and publishes messages to topics on the server. Well suited for constrained environments where the devices have limited processing and memory resources and the bandwidth is low.
- XMPP: Extensible Messaging and Presence Protocol is used for real time communication. It is a decentralized protocol and supports both client - server and server –server architecture. It has wide range of applications including messaging, gaming, multi-party chat and voice/video calls.
- DDS: Data Distribution Services is a data-centric middleware standard for device - device or machine-machine communication. It uses publish-subscribe model. DDS provides real time, scalable, dependable high performance, inter-operable data communication between publisher and subscriber.
- AMQP: Advanced Message Queuing Protocol is an open application protocol for business messaging. It supports both point-to-point and publisher/subscriber models.

# IoT architecture

IoT architecture should guarantee reliability in the operations, good failure recovery, scalability and adaptability to the mobility and dynamic nature of IoT ecosystem .The requirements for a reference architecture are

- Connectivity and communications
- Device management
- Data collection, analysis, and actuation
- Scalability
- Security
- Integration

# IoT Reference Architecture



# Device layer

- The bottom layer of the architecture is the device layer. Devices can be of various types, but in order to be considered as IoT devices, they must have some communications that either indirectly or directly attaches to the Internet. Examples
- ZigBee devices connected via a ZigBee gateway
- Bluetooth or Bluetooth Low Energy devices connecting via a mobile phone
- **Each device typically needs an identity. The identity may be one of the following:**

- A unique identifier (UUID) burnt into the device (typically part of the System-on-Chip, or provided by a secondary chip)
- A UUID provided by the radio subsystem (e.g. Bluetooth identifier, Wi-Fi MAC address)
- An OAuth2 Refresh/Bearer Token (this may be in addition to one of the above)
- An identifier stored in nonvolatile memory such as EEPROM

# Communication layer

- **The communication layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud. The most well known three potential protocols are**
- HTTP/HTTPS (and RESTful approaches on those)
- MQTT 3.1/3.1.1
- Constrained application protocol (CoAP)

# Aggregation/bus layer

- **An important layer of the architecture is the layer that aggregates and brokers communications. This is an important layer for three reasons:**

1. The ability to support an HTTP server and/or an MQTT broker to talk to the devices;
2. The ability to aggregate and combine communications from different devices and to route communications to a specific device (possibly via a gateway)
3. The ability to bridge and transform between different protocols, e.g. to offer HTTP based APIs that are mediated into an MQTT message going to the device.

# Event processing and analytics

- This layer takes the events from the bus and provides the ability to process and act upon these events. A core capability here is the requirement to store the data into a database. This layer is used for
- Highly scalable, column-based data storage for storing events
- Map-reduce for long-running batch-oriented processing of data
- Complex event processing for fast in-memory processing and near real-time reaction and autonomic actions based on the data and activity of devices and other systems
- In addition, this layer may support traditional application processing platforms.

# Client/external communication layer

- The reference architecture needs to provide a way for these devices to communicate outside of the device-oriented system. This includes three main approaches.
- create web-based front-ends and portals that interact with devices and with the event-processing layer.
- create dashboards that offer views into analytics and event processing.
- interact with systems outside this network using machine-to-machine communications (APIs). These APIs need to be managed and controlled and this happens in an API management system.

# Device manager

- Device management (DM) is handled by two components.
- A server-side system (the device manager) communicates with devices via various protocols and provides both individual and bulk control of devices. It also remotely manages software and applications deployed on the device. It can lock and/or wipe the device if necessary
- The device manager also needs to maintain the list of device identities and map these into owners. It must also work with the identity and access management layer to manage access controls over devices

# Identity and access management layer

- OAuth2 token issuing and validation
- Other identity services for identifying inbound requests from the Web layer
- Directory of users
- Policy management for access control (policy control point)

# Service Oriented Architecture(SOA)

- **Sensing layer**

  Sensing layer is integrated with hardware objects to sense the status of things

- **Network layer**

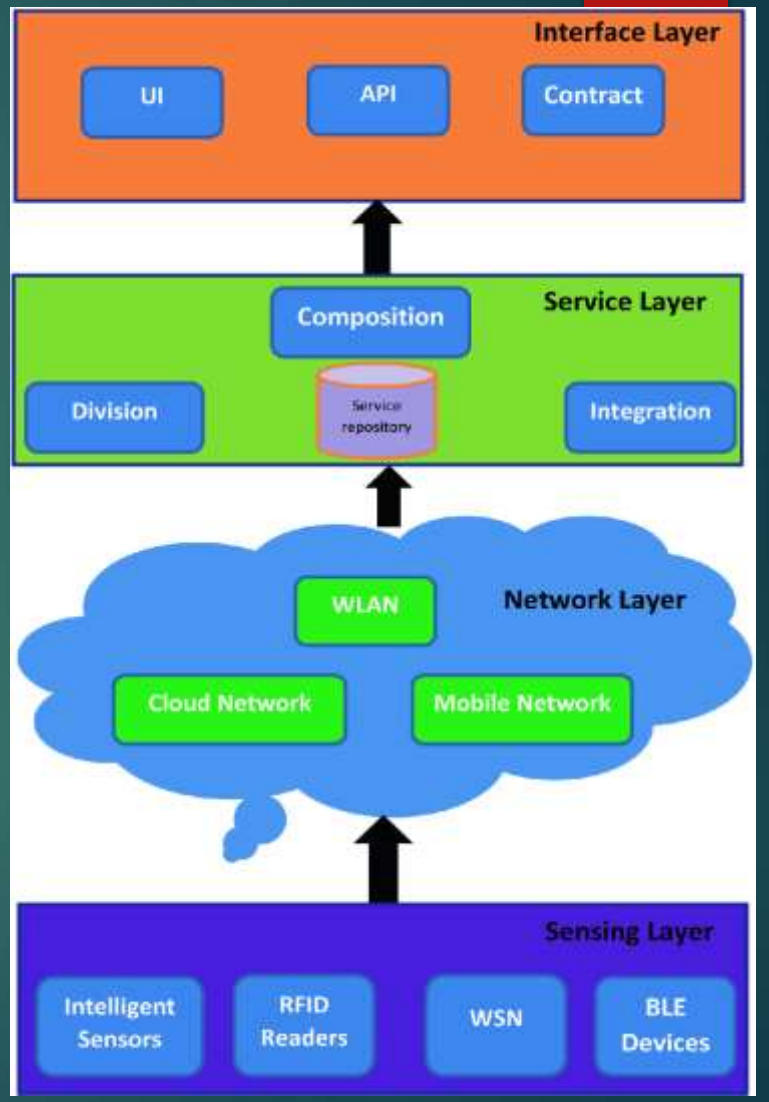
  This layer is the infrastructure over wired or wireless Connections among things

- **Service layer**

  This layer is used to create and manage services Required by users or applications

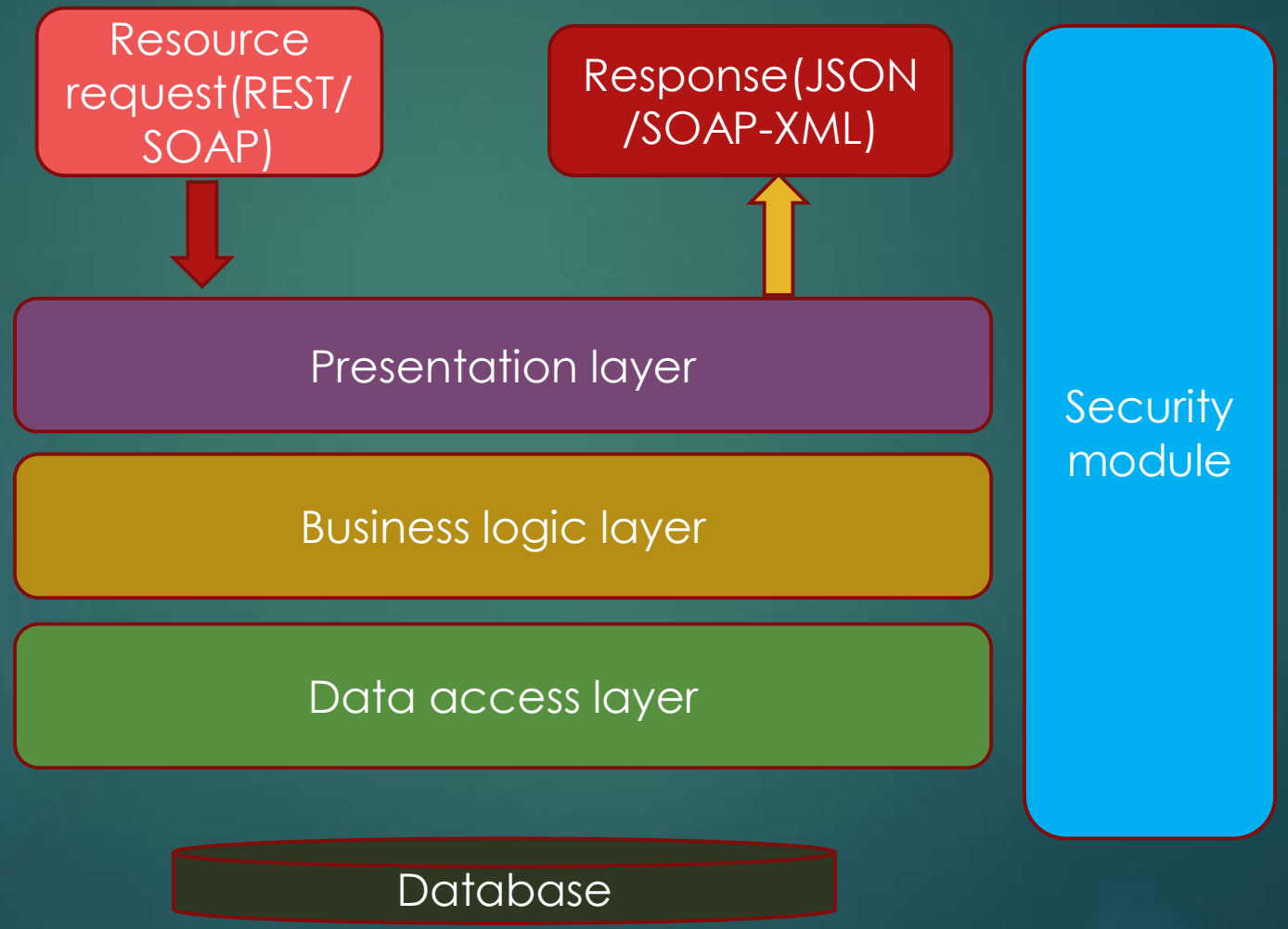
- **Interface layer**

  Interface layer consists of interaction methods with users or applications



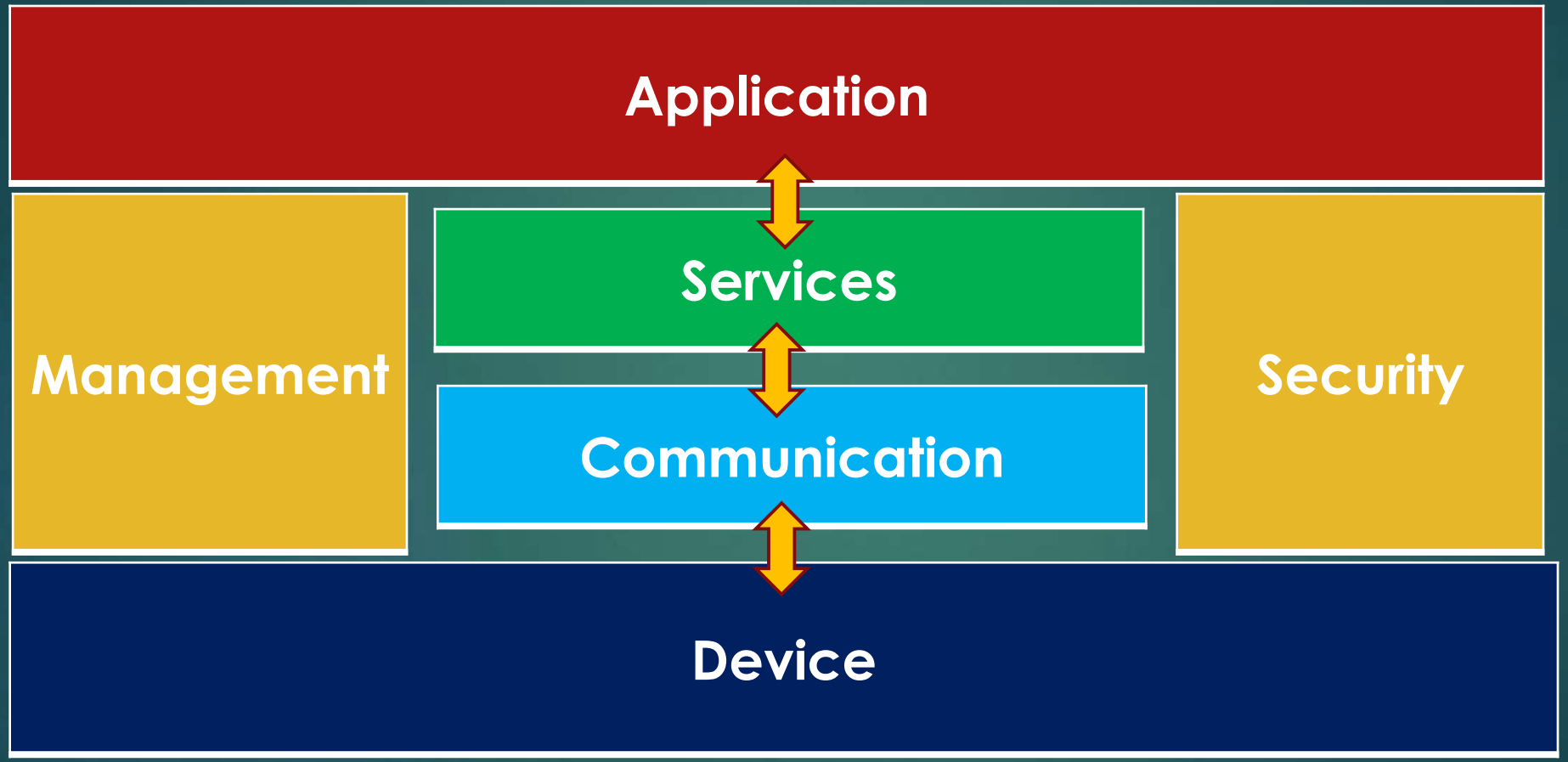
- In SOA architecture, the complex system is divided into smaller systems which are loosely coupled so that maintainability is high.
- In case of component failure, system can operate normally. Reliability is high.
- SOA has appropriate level of abstraction, interoperability and scalability.
- SOA has the ability to build diverse and complex services by composing different functions of the system.

# API oriented architecture



- Architecture is structured into a secure API, a backbone, and separate device networks with standard interface to the backbone.
- In conventional method, SOAP(Simple Object Access Protocol) or RMI(remote procedure invocation) are used to describe, discover and call services. They have huge overhead and complexity.
- Web API and REST are the alternative solutions.
- They use light weight data exchange formats like JSON which replace XML files to describe services. They have less overhead and uses the communication channel and processing ability of the devices efficiently.
- The API decouples innovation of services and service logic from protocols and network elements. It also enables service portability between systems, i.e. a service may be allocated to end-systems or servers, with possible relocation and replication throughout its lifecycle.
- Building APIs for IoT applications helps the service provider to focus on functionality, efficient service monitoring and pricing tools.

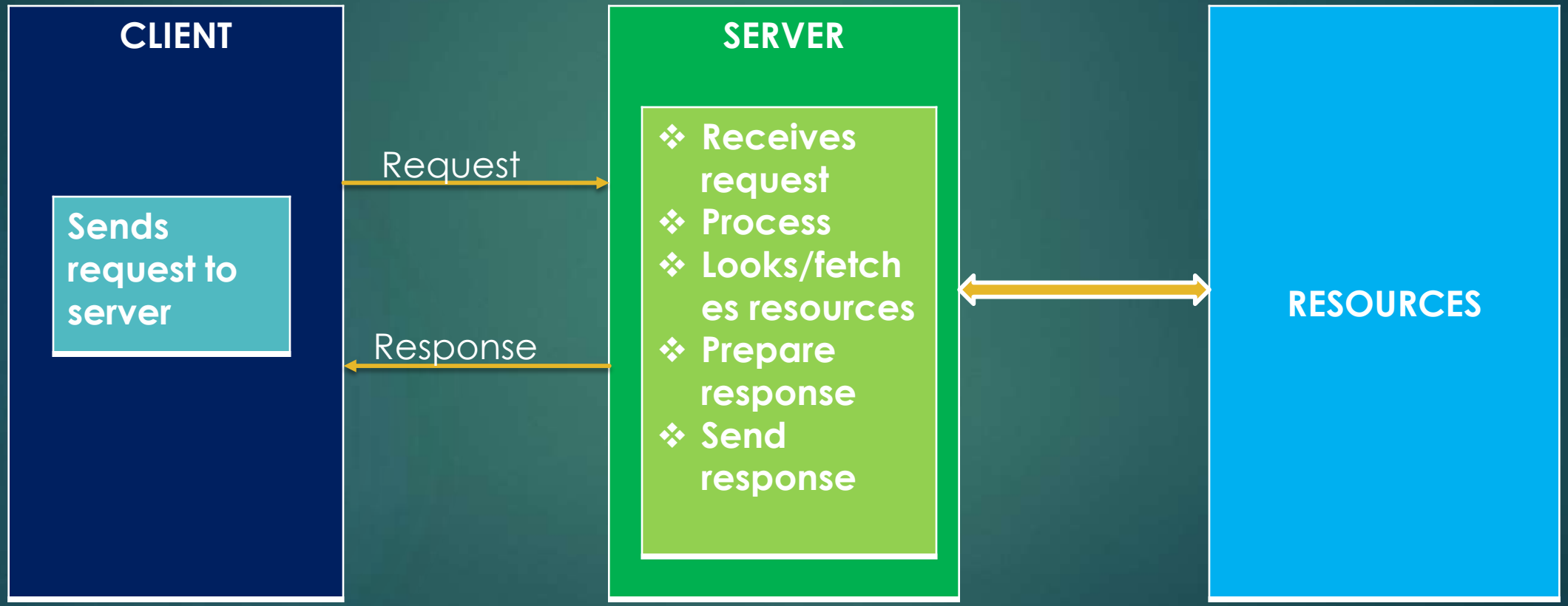
# Functional Blocks of IoT



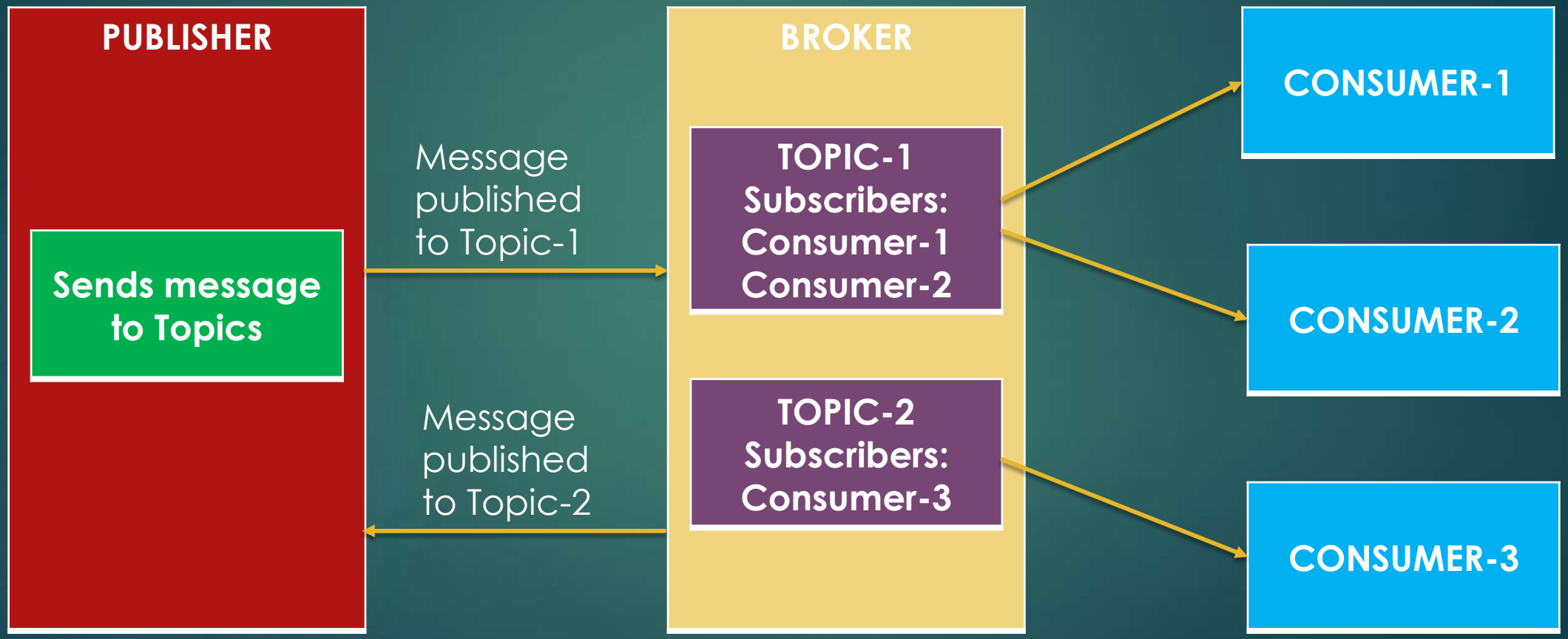
# IoT communication models

- Request-Response
- Publish-Subscribe
- Push-Pull
- Exclusive Pair

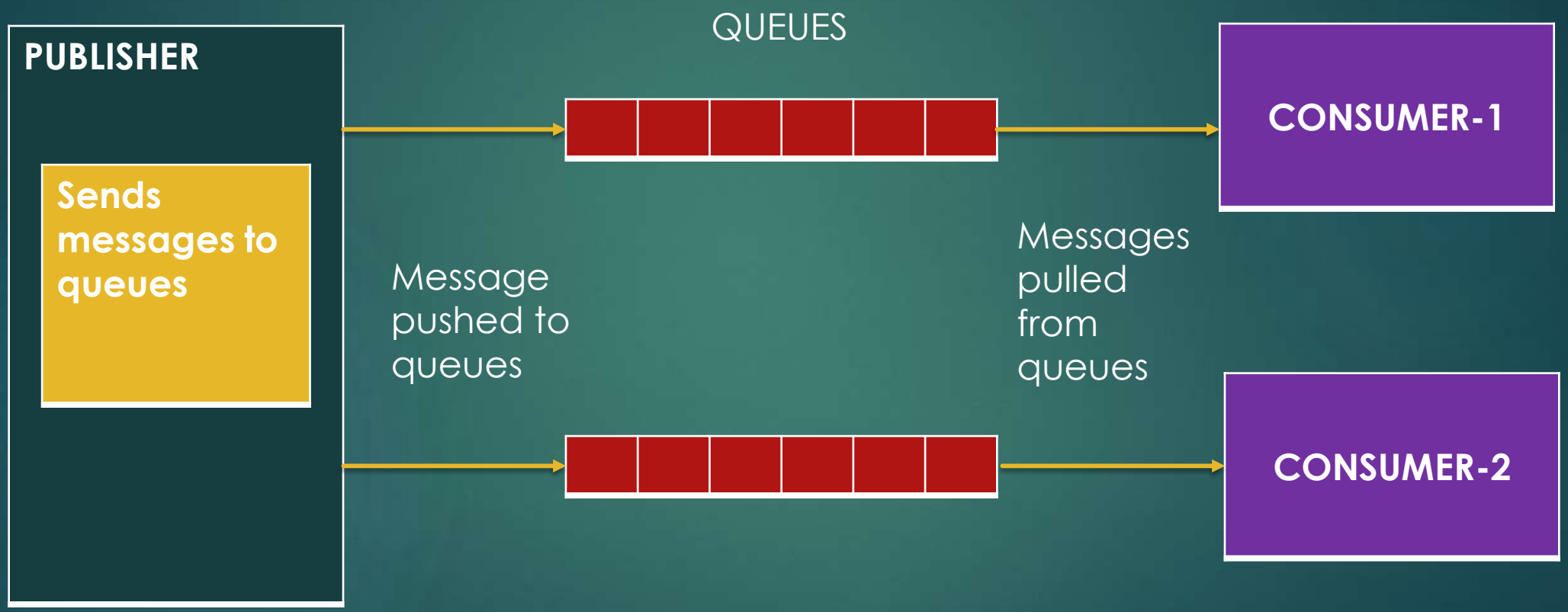
# Request-Response model



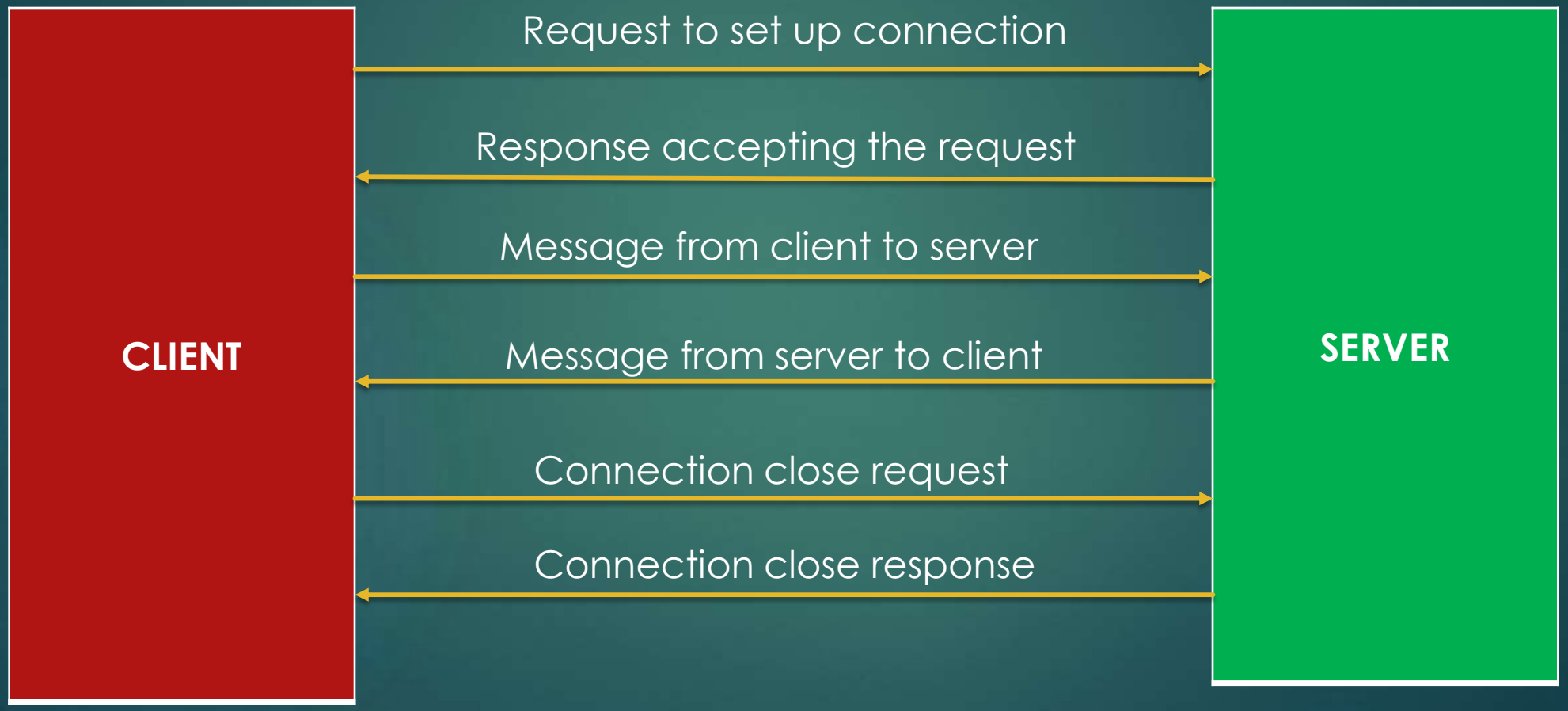
# Publish-Subscribe Communication model



# Push-Pull Communication model



# Exclusive pair communication model



# IoT communication APIs

- REST based communication  APIs
- Web Socket –based Communication APIs

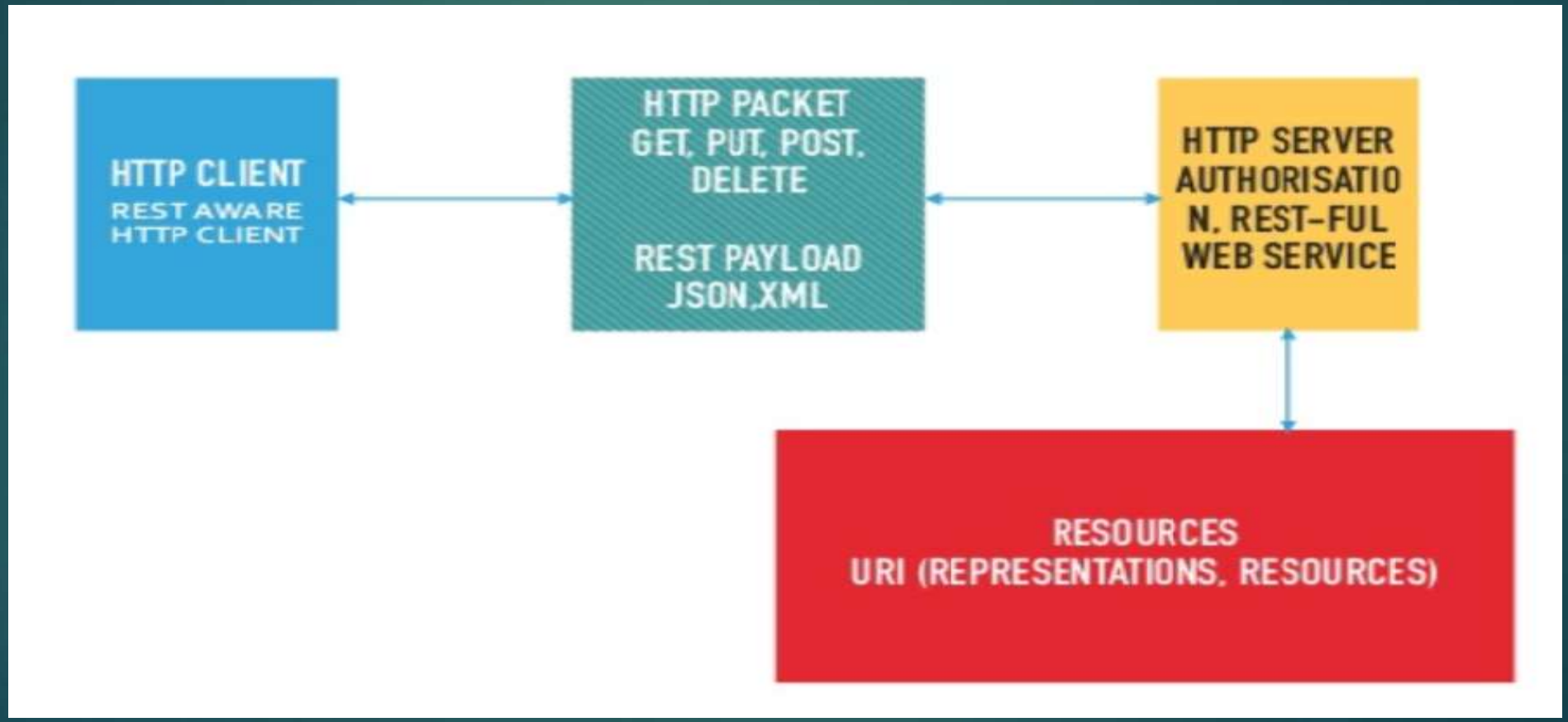
# REST based communication  APIs

- A RESTful API is an application program interface (API)
- based on REpresentational STate transfer (**REST**), an architectural style and approach to communications often used in **web services** development
- REST is a logical choice for building APIs that allow users to connect to, manage and interact with cloud services
- A RESTful API uses existing HTTP methodologies. They use GET to retrieve a resource; PUT to change the state of or update a resource, which can be an object, file or block; POST to create that resource; and DELETE to remove it.
- With REST, networked components are a resource the user requests access to.

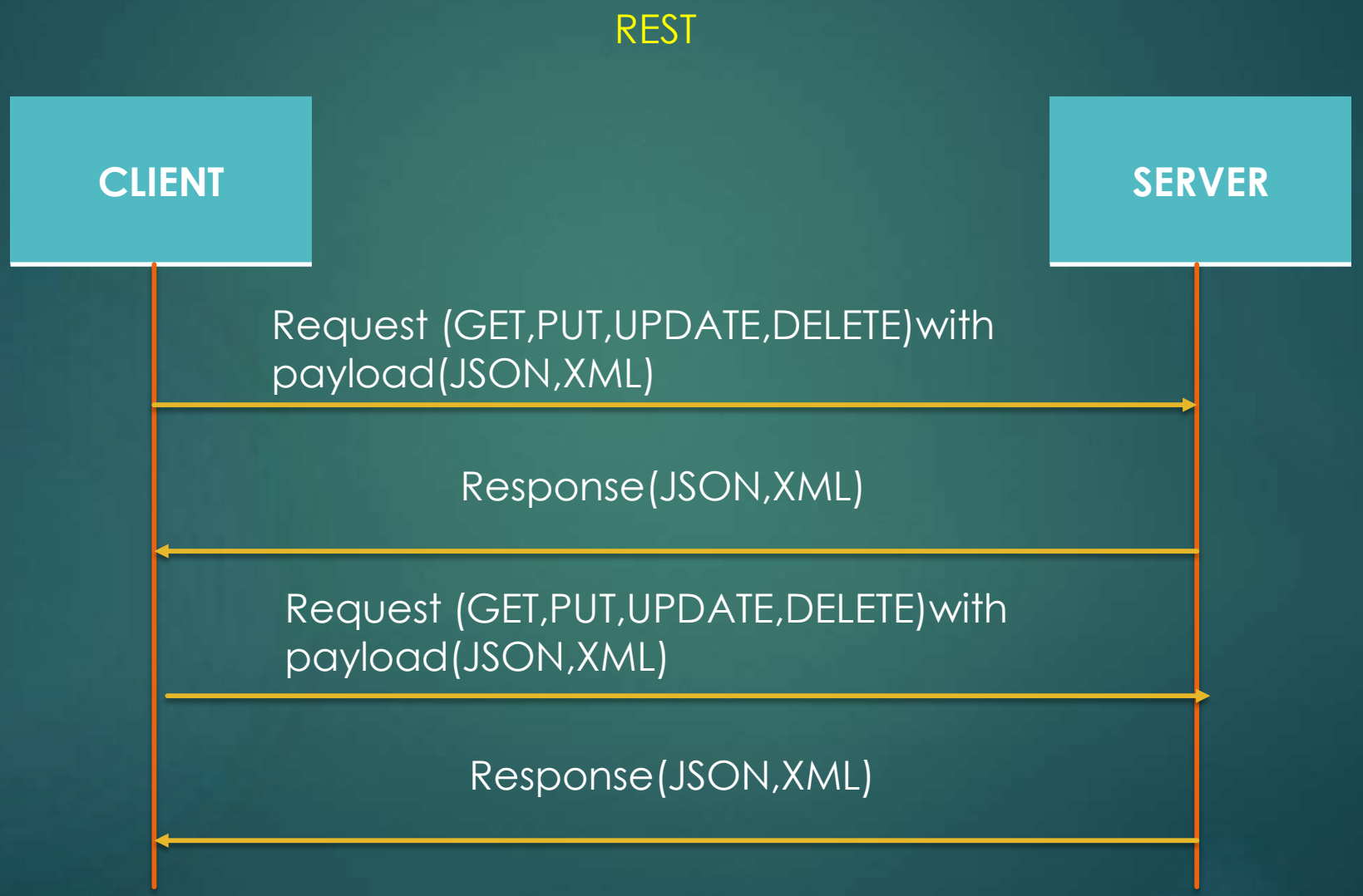
# REST based APIs Constraints

- Client server: UI and request-gathering concerns are the client's domain. Data access, workload management and security are the server's domain
- State less :All client-server operations should be stateless, and any state management that is required should take place on the client, not the server.
- Cacheable : All resources should allow caching unless explicitly indicated that caching is not possible.
- Layered system: REST allows for an architecture composed of multiple layers of servers.
- Uniform interface : Resources should be uniquely identifiable through a single URL,
- Code on demand : Most of the time, a server will send back static representations of resources in the form of XML or JSON. However, when necessary, servers can send executable code to the client.

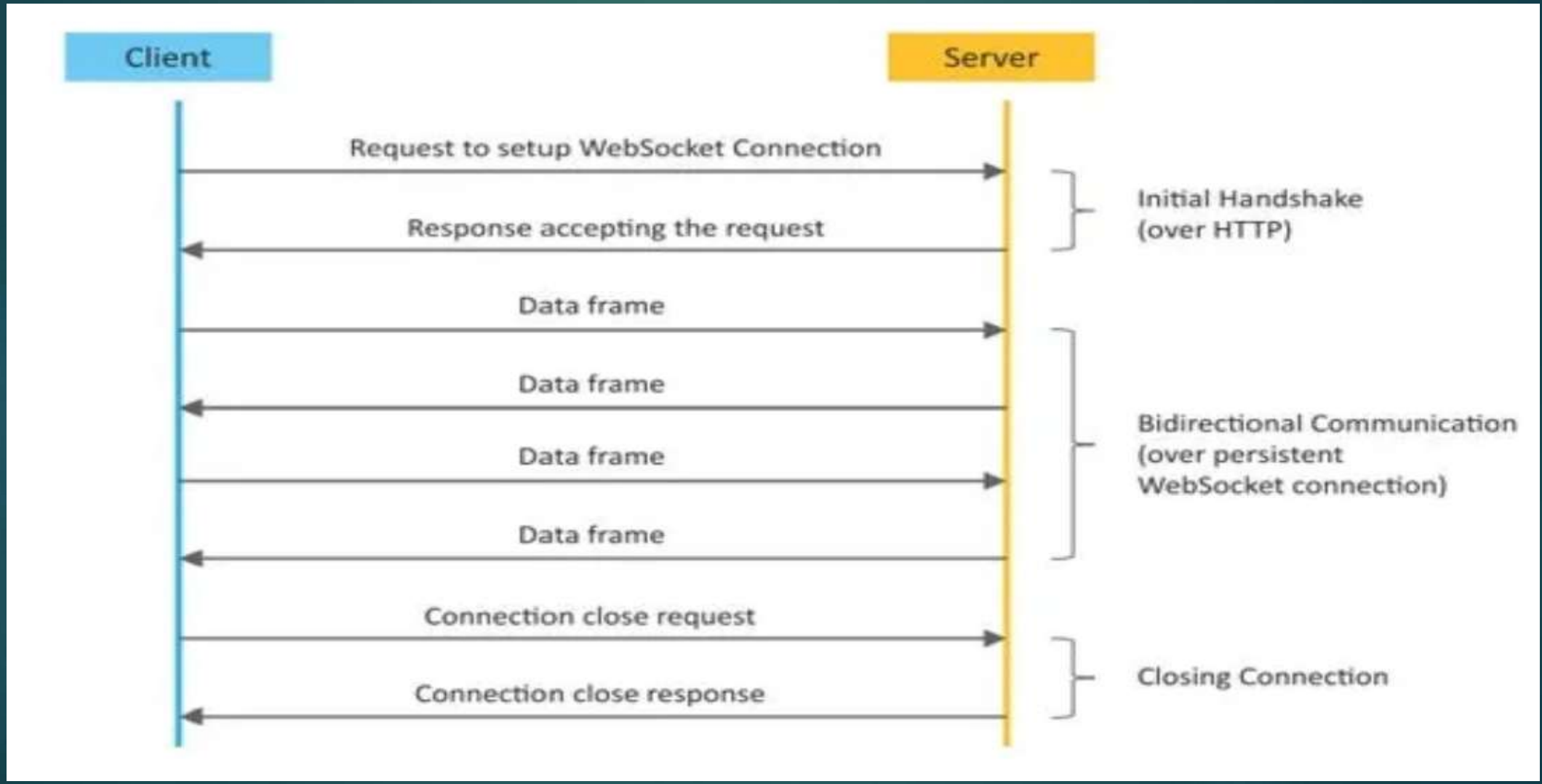
# Communication with REST APIs



# Request-Response model used by REST



# Web socket based  communication APIs



# MOD1-PART3

# RESOURCE MANAGEMENT

- IoT architecture has various resources connected with the network.
- The IoT networked resources consists of computing elements, storage and energy.
- Efficient resource management helps IoT networked devices to utilize these resources in an efficient and cost-effective way to improve system performance and productivity.
- Resource allocation for IoT devices has many challenges due to heterogeneity dynamic nature and distance between the devices.

.

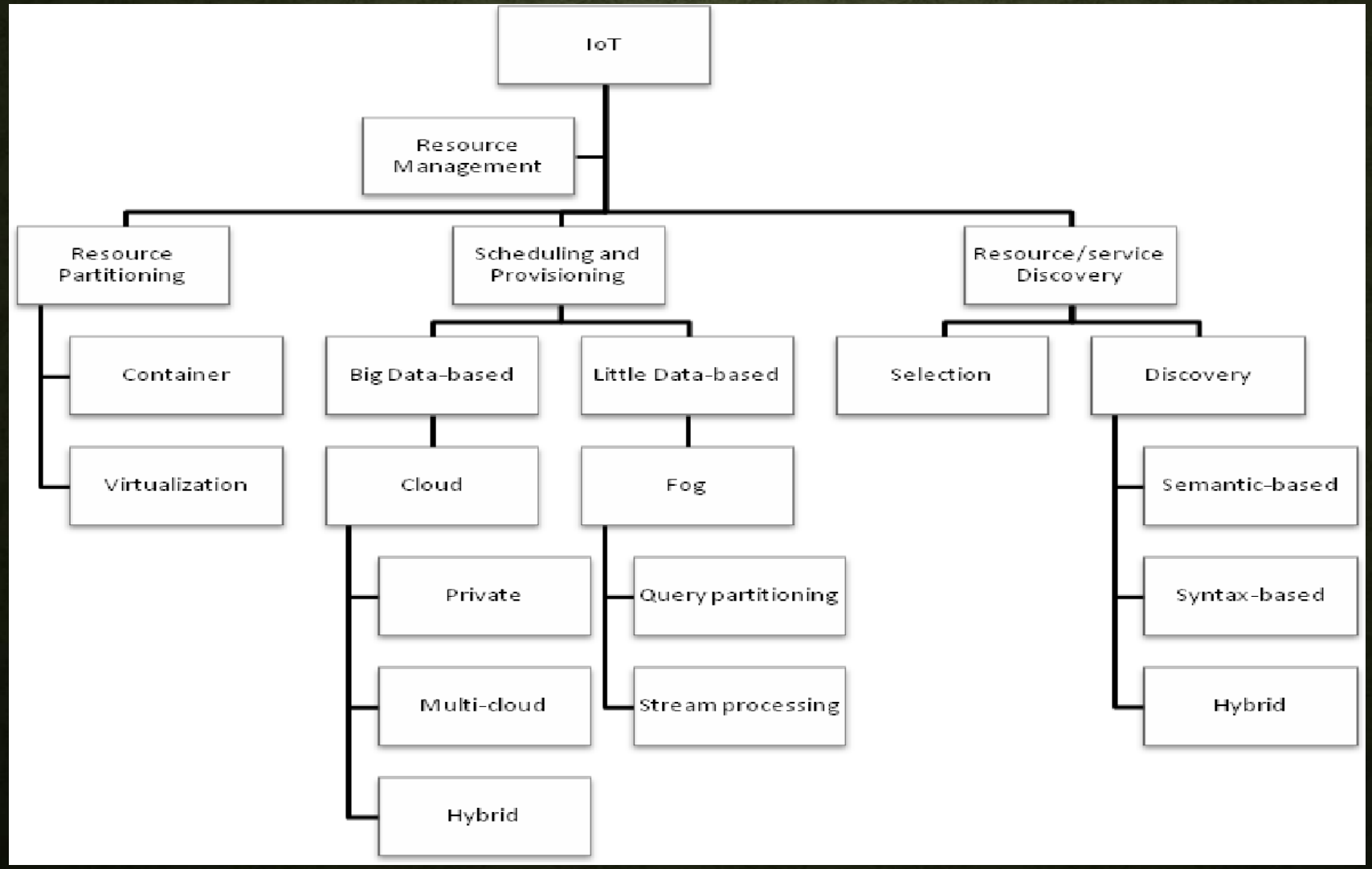
# REQUIREMENTS OF A RESOURCE MANAGEMENT SYSTEM

- An effective Resource management system supports
- standard Quality of Service (QoS),
- cost minimization,
- energy consumption reduction,
- increase resource utilization
- robustness
- scalability
- guaranteed the Service level agreement between the Cloud based IoT system application providers and costumers where user's requirements should be matched in an effective way.

# STEPS

- Resource management involves
- resource discovery and identification,
- resource partitioning,
- resource provisioning and scheduling tasks

# TAXONOMY OF RESOURCE MANAGEMENT IN IOT



# 1.RESOURCE PARTITIONING

- Resource partitioning is done to gain high utilization rate
- In cloud computing virtualization techniques and commodity infrastructures are used for resource partition.
- This method is not suitable for IoT as large memory capacity and processing power is required for virtualization techniques.
- This leads to ,a new light weight virtualization solution ,containers .
- This new virtualization technique is suitable for demand of devices with limited resources.
- Docker and Rocket are the two most famous container solutions.

# CONTAINERS

- Containers are able to provide portable and platform independent platform for hosting applications and all their dependencies, configurations and input/output settings
- This reduces the burden of handling different platform-specific requirements for designing and developing different applications
- This introduce a convenient level of transparency between applications, architects and developers
- Containers are light weight solutions that enable infrastructure providers to efficiently utilize their hardware resources, by eliminating need for expensive hardware and virtualization software packages.
- Containers requires less spin up time ,thus suitable for distributed applications in IoT that scale up in short amount of time.
- Container based virtualization can bring advantages in terms of performance and security

# 2.COMPUTATION/CODE OFFLOADING

- Code offloading is a solution for addressing limited resource availability in mobile and smart devices .
- Computation offloading is the transfer of resource intensive computational tasks to a separate processor
- Developers can manually annotate functions required to execute on another device.
- But in case of network fluctuations or increased latency, static code analysers and dynamic code parsers can efficiently annotate functions.
- The code offloading techniques leads to more efficient power management, fewer storage requirement and higher application performance

# 3.IDENTIFICATION AND RESOURCE/SERVICE DISCOVERY

- IoT environments require an efficient and standard way for service discovery, composition and their integration.
- Discovery module in IoT has two steps

  identify and allocate actual device using the metadata of objects.

  discover the target service that needs to be invoked.

Discovery algorithms are used in IoT discovery module.

Discovery algorithms should be efficient to avoid execution delays and runtime failures.

# IOT DATA MANAGEMENT AND ANALYTICS

- As more and more devices are added to IoT networks, the data generated by these systems becomes overwhelming.
- Traditional data management systems are can not handle big data.
- Big data analytics are used in IoT to process data.
- Big data is characterized by 3 vs. Velocity : rate of growth of data

  volume : size of data and variety :different forms of data

Different approaches are used to process data depending upon the combinations of these three dimensions.

Batch processing and stream processing are two approaches.

Cloud and fog computing are the two big data platforms used in IoT

# CLOUD COMPUTING

- Cloud computing is the delivery of on-demand computing services, typically over the internet and on a pay-as-you-go basis.
- Cloud computing can be broken down into
- Infrastructure as a service
- Platform as a service
- Software as a service

Cloud can process data in both batch or stream approach.

Based on the data the clod can be private, public or hybrid.

# FOG COMPUTING

- Fog computing is a decentralized computing infrastructure in which data, compute, storage and applications are located somewhere between the data source and the cloud.
- Fog computing brings the advantages and power of the cloud closer to where data is created and acted upon.
- In an IoT fog environment, data from devices will be processed by fog nodes in the middle layer. In this way sensitive data will be processed faster to provide instant response.
- The data which needs to be stored and processed in the cloud will be sent to the cloud.

**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 |

# COMPARISON OF IOT PROTOCOLS: IOT PROTOCOLS ARE ALREADY DESCRIBED IN PPT-2. COMPARISON OF SOME PROTOCOLS ARE GIVEN BELOW.

| Protocol Name | Transport Protocol | Messaging Model | Security | Best-Use Cases | Architecture |
|---|---|---|---|---|---|
| AMPQ | 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 |

# MOD1-PART4

# IoT applictaions

- IoT applications range from home automation to sophisticated environments such as smart cities or e-governance.
- Industry based applications: logistic and transportation, supply chain management, aviation industry , automation systems etc.
- Society based applications: healthcare systems, smart cities and buildings, smart shopping etc.
- Environment based applications : disaster management, environmental monitoring, smart irrigation, smart metering etc.

- Based on the usage domain IoT applications are categorised in to three.
- monitoring and actuating
- Business process and data analysis
- Information gathering and collaborative consumption

# Monitoring and actuating

- Monitoring and actuating devices through APIs can be helpful in multiple domains.
- APIs can report power usage, equipment performance, sensor status, and they can perform actions based on predefined commands.
- Real time applications can make use of these features to report current system status.
- Smart grids can increase productivity by identifying performance defects by applications of anomaly detection on the collected data.
- By incorporating IoT in buildings, or in the construction process helps to move towards green solutions , save energy, consequently minimise operation cost.

# Business process and data analysis

- Big data analytics comes under the following levels.
- Society level: where IoT mainly influences and improves government services by reducing  cost and increasing transparency and accountability
- Industry level : manufacturing, emergency services education and retailing
- Organization level: same as society level
- Individual level: daily life improvements, individual efficiency, and productivity growth are considered as IoT benefits

# Information gathering and collaborative consumption

- Social IoT is where IoT meet social networks .It links objects around us with our social media and daily interaction with other people making them more smarter and interactable.
- SIoT concepts affects peoples life styles
- IoT is helpful in evaluation of trust of crowds involved in an IoT process.
- Using humans and their relations interactions and communities are useful to find effective IoT services and objects

# security

- In IoT the heterogeneous and distributed nature makes the end devices vulnerable
- End devices are less protected than servers  and therefore easily accessible to attackers
- Since devices are closer to users security leads to leaking of valuable information
- In an IoT environment the resource constraints are the key barrier for implementing standard security mechanisms.
- Wireless networks used by major sensor networks are more vulnerable to eavesdropping or proxy attacks

- Normal cryptographic algorithms require more bandwidth and energy to provide end to end protection against attacks on confidentiality and authenticity. RFID and WSN solutions are to use light weight algorithms.
- Symmetric cryptography is used in constrained environments since it requires few resources.

# Identity management and authentication

- Methods of identifying devices and setting their access levels are very important in IoT ecosystem.
- Consumers data sources and service providers need to be connected by identity management and authentication methods to create the IoT system
- End devices or objects should have unique identifiers. RFID tags ,electric product code, URI etc are used to provide identification numbers
- IoT sensors and smart devices may share the same geographical coordinates or may fall under the same group of devices, so identity management can be given to a local system.
- Local identity management system can enforce and monitor access control policies and establish connections with external partners.

# privacy

- Secure data  storage and access mechanisms are important.
- Sensors are capturing very private and sensitive data, data privacy should ensure that users have control over the data they share and the people who have the access to the data.
- In IoT privacy can be achieved by a centralized approach or by having each entity manage its own data a technique known as privacy by design.
- Distributed privacy preserving algorithms are developed to handle data scattering
- Privacy enhancing technologies have been developed for protecting collaborative protocols.

# Standardization and regulatory limitations

- Defining and broad casting standards will help more users and providers to join IoT.
- But increase in IoT cause difficulties in standardization
- Strict regulations about accessing radio frequency levels , creating sufficient level of interoperability among devices, authentication, identification, authorization and communication protocols are all open challenges facing IoT standardization.