DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Code	Course Title
U20CSCT23	**Internet Of Things**

UNIT I INTRODUCTION TO IoT

Internet of Things - Physical Design- Logical Design- IoT Enabling Technologies - IoT Levels & Deployment Templates - Domain Specific IoTs - IoT and M2M - IoT System Management with NETCONF-YANG- IoT Platforms Design Methodology - IoT - Challenges and Issues.

Internet of Things:

- The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.
- IoT is an ecosystem of connected physical objects that are accessible through Internet
- IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit.

Characteristics of IoT:

1. **Connectivity**
   Things in I.O.T. should be connected to the infrastructure, without connection nothing makes sense.
2. **Intelligence**
   Extraction of knowledge from the generated data is important, sensor generate data and this data and this data should be interpreted properly.
3. **Scalability**
   The no. of things getting connected to the I.O.T. infrastructure is increased day by day. Hence, an IOT setup shall be able to handle the massive expansion.
4. **Unique Identity**
   Each IoT device has a unique identity and a unique identifier(IP address).
5. **Integrated into Information Network**
   It allow them to communicate and exchange data with other devices and systems.

How IoT Works?

Sensors/Devices

Sensors or devices are a key component that helps you to collect live data from the surrounding environment.

Connectivity

All the collected data is sent to a cloud infrastructure. The sensors should be connected to the cloud using various mediums of communications. These communication mediums include mobile or satellite networks, Bluetooth, WI-FI, WAN, etc.

Data Processing

Once that data is collected, and it gets to the cloud, the software performs processing on the gathered data. identifying objects, using computer vision on video.

User Interface

The information needs to be available to the end-user in some way which can be achieved by triggering alarms on their phones or sending them notification through email or text message.

IoT enabling technologies:

1. Wireless Sensor Network(WSN) :

A WSN comprises distributed devices with sensors which are used to monitor the environmental and physical conditions. A wireless sensor network consists of end nodes, routers and coordinators. End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers. The coordinator also acts as the gateway that connects WSN to the internet.

Example –

Weather monitoring system

Indoor air quality monitoring system

Soil moisture monitoring system

Surveillance system

Health monitoring system

2. Cloud Computing :

It provides us the means by which we can access applications as utilities over the internet. Cloud means something which is present in remote locations.

With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet.

Characteristics –

Broad network access

On demand self-services

Rapid scalability

Measured service

Pay-per-use

3. Big Data Analytics:

It refers to the method of studying massive volumes of data or big data. Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases.

Big data is gathered from a variety of sources including social network videos, digital images, sensors and sales transaction records.

Examples –

Bank transactions

Data generated by IoT systems for location and tracking of vehicles

E-commerce and in Big-Basket

Health and fitness data generated by IoT system such as a fitness bands

4. Communications Protocols :

They are the backbone of IoT systems and enable network connectivity and linking to applications. Communication protocols allow devices to exchange data over the network. Multiple protocols often describe different aspects of a single communication. A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack.

They are used in

Data encoding

Addressing schemes

5. Embedded Systems :

It is a combination of hardware and software used to perform special tasks.

It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).

It collects the data and sends it to the internet.

Embedded systems used in

Examples –

Digital camera

DVD player, music player

Industrial robots

Wireless Routers etc.

Physical Design of IoT

- The "Things" in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.

IoT devices can:

- Exchange data with other connected devices and applications (directly or indirectly).
- Collect data from other devices and process the data locally.
- Send the data to centralized servers or cloud-based application back-ends for processing the data.
- Perform some tasks locally and other tasks within the IoT infrastructure, based on temporal and space constraints.

Logical Design of IoT

- Logical design of an IoT system refers to an abstract representation of the entities and processes without going into the low-level specifics of the implementation.
- IoT Functional Blocks
- IoT Communication Models
- IoT Communication APIs

IoT Functional Blocks

- Provide the system the capabilities for identification, sensing, actuation, communication and management.

Device: An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.

Communication: handles the communication for IoT system.

Services: for device monitoring, device control services, data publishing services and services for device discovery.

Management: Provides various functions to govern the IoT system.

Security: Secures IoT system and priority functions such as authentication, authorization, message and context integrity and data security.

Application: IoT application provide an interface that the users can use to control and monitor various aspects of IoT system.

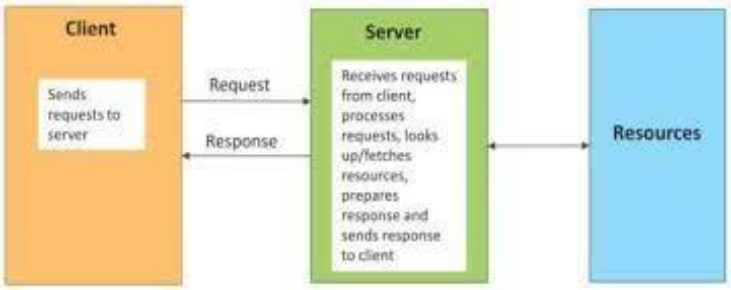
IoT Communication Models

- Request-Response Model
- Publish- Subscribe Communication Models

- Push-Pull Model
- Exclusive Pair Model

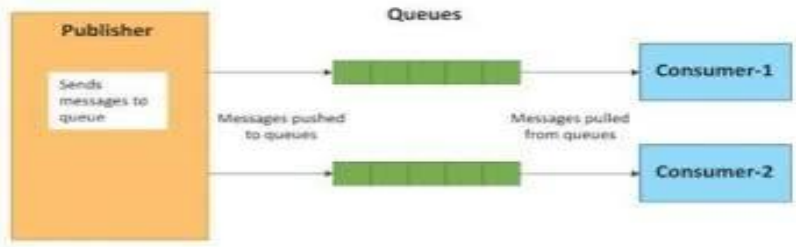
Request-Response communication model:

- Request-Response is a communication model in which the client sends requests to the server and the server responds to the requests.
- When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then sends the response to the client.



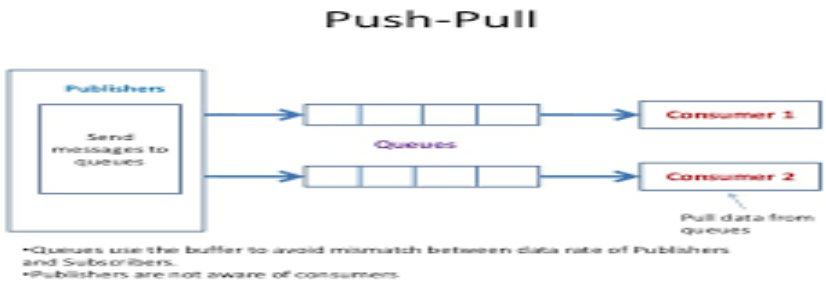
Publish-Subscribe communication model

- Publish-Subscribe is a communication model that involves publishers, brokers and consumers.
- Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.
- Consumers subscribe to the topics which are managed by the broker.
- When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.



Push-Pull communication model:

- Push-Pull is a communication model in which the data producers push the data to queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers.
- Queues help in decoupling the messaging between the producers and consumers.
- Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumers pull data.

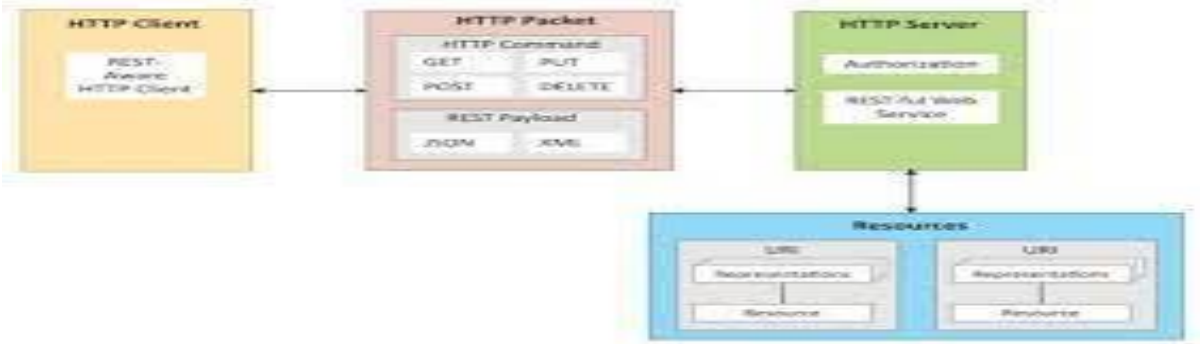


Exclusive Pair communication model

- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.
- Once the connection is setup it remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after connection setup.

IoT Communication APIs

- REST based communication APIs(Request-Response Based Model)
- Web Socket based Communication APIs(Exclusive Pair Based Model)
- Representational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system's resources and how resource states are addressed and transferred.
- REST APIs follow the request response communication model.
- The REST architectural constraints apply to the components, connectors, and data elements, within a distributed hypermedia system.



Web Socket-based Communication APIs

- Web Socket APIs allow bidirectional, full duplex communication between clients and servers.
- Web Socket APIs follow the exclusive pair communication model.

IoT Levels & Deployment Templates

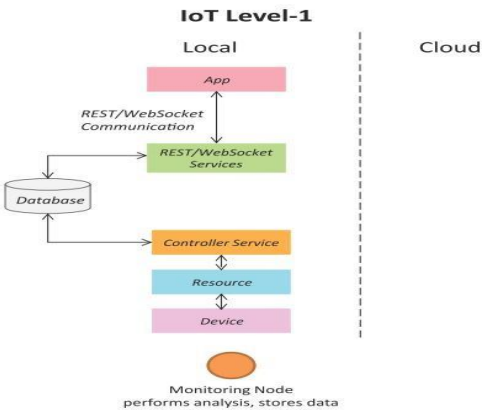
An IoT system comprises of the following components:

- Device: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities. You learned about various examples of IoT devices in section
- Resource: Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.
- Controller Service: Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

IoT Level-1

- A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application
- Level-1 IoT systems are suitable for modeling low- cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.

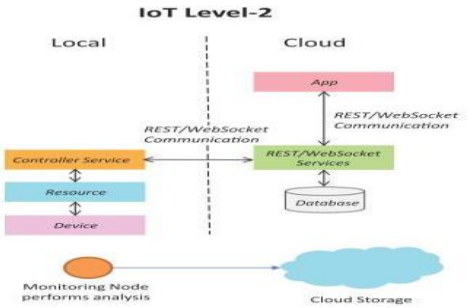
  Example: Home automation



IoT Level-2

- A level-2 IoT system has a single node that performs sensing and/or actuation and local analysis.
- Data is stored in the cloud and application is usually cloud- based.
- Level-2 IoT systems are suitable for solutions where the data involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself.

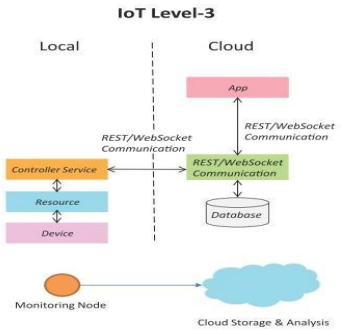
  Example: Smart irrigation



IoT Level-3

- A level-3 IoT system has a single node.Data is stored and analyzed in the cloud and application is cloud- based.
- Level-3 IoT systems are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.

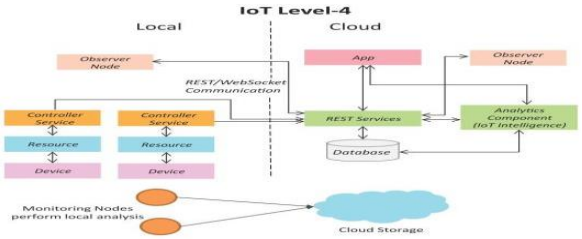
  Example: Tracking package handling



IoT Level-4

- A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based.
- Level-4 contains local and cloud- based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.
- Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.

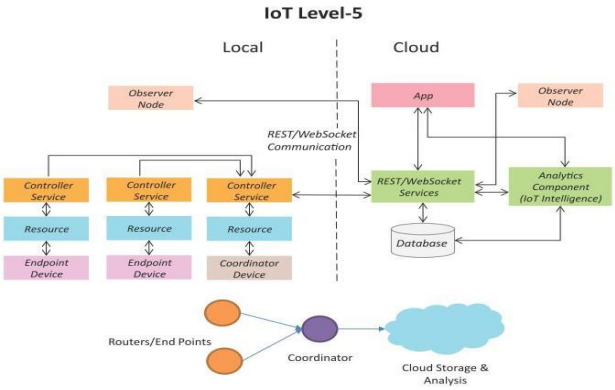
  Example: Noise monitoring



IoT Level-5

- A level-5 IoT system has multiple end nodes and one coordinator node.
- The end nodes that perform sensing and/or actuation.
- Coordinator node collects data from the end nodes and sends to the cloud.
- Data is stored and analyzed in the cloud and application is cloud-based.
- Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.

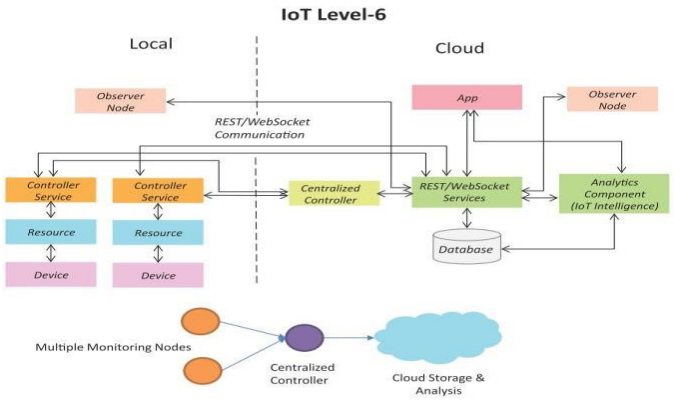
  Example: Forest fire detection



IoT Level-6

- A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.
- Data is stored in the cloud and application is cloud-based.
- The analytics component analyzes the data and stores the results in the cloud database.
- The results are visualized with the cloud-based application.
- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.

  Example: Weather monitor system



DOMAIN SPECIFIC IoTs

Home Automation:

- Smart Lighting: helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or diming the light when needed.
- Smart Appliances: make the management easier and also provide status information to the users remotely.
- Intrusion Detection: use security cameras and sensors(PIR sensors and door sensors) to detect intrusion and raise alerts. Alerts can be in the form of SMS or email sent to the user.
- Smoke/Gas Detectors: Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Alerts raised by smoke detectors can be in the form of signals to a fire alarm system. Gas detectors can detect the presence of harmful gases such as CO, LP Getc.,

Cities:

- Smart Parking: make the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the no. of empty parking slots and send information over internet to smart application back ends.
- Smart Lighting: for roads, parks and buildings can help in saving energy.
- Smart Roads: Equipped with sensors can provide information on driving condition, travel time estimating and alert in case of poor driving conditions, traffic condition and accidents.
- Structural Health Monitoring: uses a network of sensors to monitor the vibration levels in the structures such as bridges and buildings.
- Surveillance: The video feeds from surveillance cameras can be aggregated in cloud based scalable storage solution.

Environment:

- Weather Monitoring: Systems collect data from a no. of sensors attached and send the data to cloud based applications and storage back ends. The data collected in cloud can then be analyzed and visualized by cloud based applications.

- Air Pollution Monitoring: System can monitor emission of harmful gases(CO2, CO, NO, NO2 etc.,) by factories and automobiles using gaseous and meteorological sensors. The collected data can be analyzed to make informed decisions on pollutions control approaches.
- Noise Pollution Monitoring: Due to growing urban development, noise levels in cities have increased and even become alarmingly high in some cities

Energy

- Smart Grids: is a data communication network integrated with the electrical grids that collects and analyze data captured in near-real-time about power transmission, distribution and consumption. Smart grid technology provides predictive information and recommendations to utilities, their suppliers, and their customers on how best to manage power. By using IoT based sensing and measurement technologies, the health of equipment and integrity of the grid can been valuated.
- Renewable Energy Systems: IoT based systems integrated with the transformers at the point of interconnection measure the electrical variables and how much power is fed into the grid. For wind energy systems, closed-loop controls can be used to regulate the voltage at point of interconnection which coordinate wind turbine outputs and provides power support.

Retail

- Inventory Management: IoT systems enable remote monitoring of inventory using data collected by RFIDreaders.
- Smart Payments: Solutions such as contact-less payments powered by technologies such as Near Field Commu

Logistics

- Route generation & scheduling: IoT based system backed by cloud can provide first response to the route generation queries and can be scaled upto serve a large transportation network.
- Fleet Tracking: Use GPS to track locations of vehicles inreal-time.

Agriculture:

- Smart Irrigation: to detemine moisture amount insoil.
- Green House Control: to improveproductivity.

Industry

- Machine diagnosis andprognosis
- Indoor Air Quality Monitoring

Health and LifeStyle:

- Health & Fitness Monitoring

- Wearable Electronics

Machine-to-Machine (M2M)

- Machine-to-Machine (M2M) refers to networking of machines (or devices) for the purpose of remote monitoring and control and data exchange.
- An M2M area network comprises of machines (or M2M nodes) which have embedded hardware modules for sensing, actuation and communication.
- Various communication protocols can be used for M2M local area networks such as ZigBee, Bluetooh, ModBus, M-Bus, Wirless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, etc.
- The communication network provides connectivity to remote M2M area networks. The communication network can use either wired or wireless networks (IPbased).
- While the M2M area networks use either proprietary or non-IP based communication protocols, the communication network uses IP-based networks.

Difference between IoT and M2M:

Communication Protocols:

- M2M and IoT can differ in how the communication between the machines or devices happens.
- M2M uses either proprietary or non-IP based communication protocols for communication within the M2M area networks.

Machines in M2M vs Things in IoT:

- The "Things" in IoT refers to physical objects that have unique identifiers and can sense and communicate with their external environment (and user applications) or their internal physical states.
- M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area network.

Hardware vs Software Emphasis:

- While the emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is more on software.

Data Collection & Analysis:

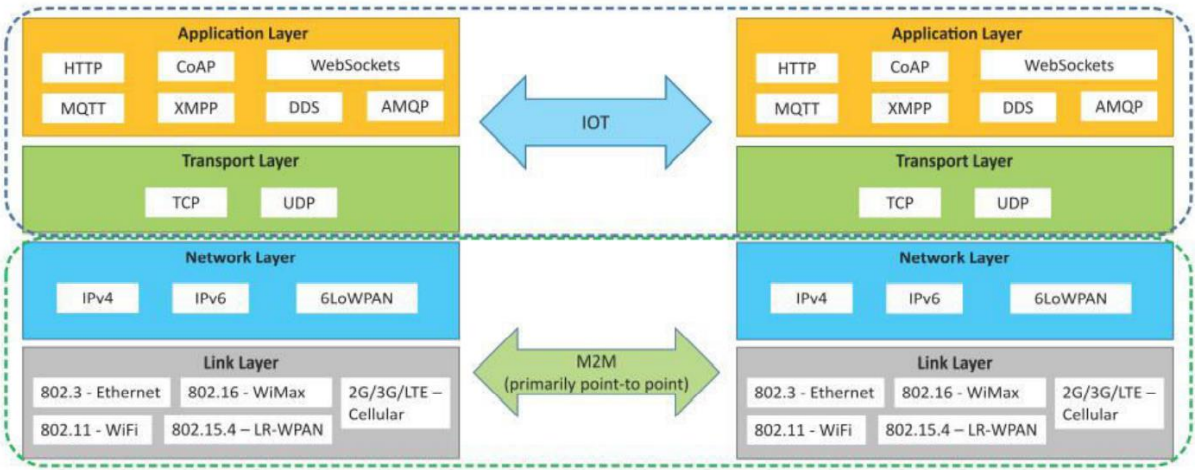
- M2M data is collected in point solutions and often in on-premises storage infrastructure.
- In contrast to M2M, the data in IoT is collected in the cloud (can be public, private or hybrid cloud).

Applications:

- M2M data is collected in point solutions and can be accessed by on-premises applications such as diagnosis applications, service management applications, and onpremisis enterprise applications.

- IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc.

Communication in IoT vs M2M:



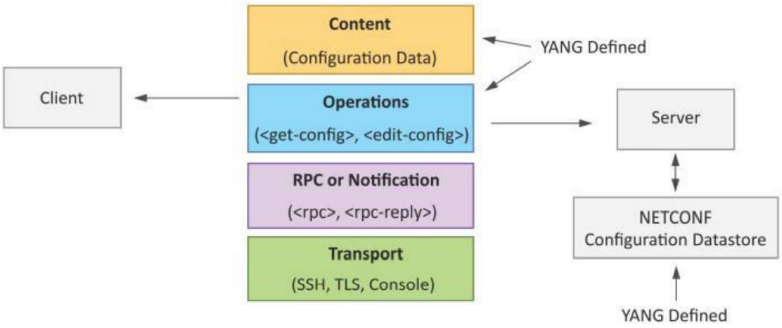
IoT System Management with NETCONF-YANG

Need for IoT Systems Management:

- Automating Configuration
- Monitoring Operational & Statistical Data
- Improved Reliability
- System Wide Configurations
- Multiple System Configurations
- Retrieving & Reusing Configurations

NETCONF

Network Configuration Protocol (NETCONF) is a session-based network management protocol. NETCONF allows retrieving state or configuration data and manipulating configuration data on network devices



- NETCONF works on SSH transport protocol.
- Transport layer provides end-to-end connectivity and ensure reliable delivery of messages.

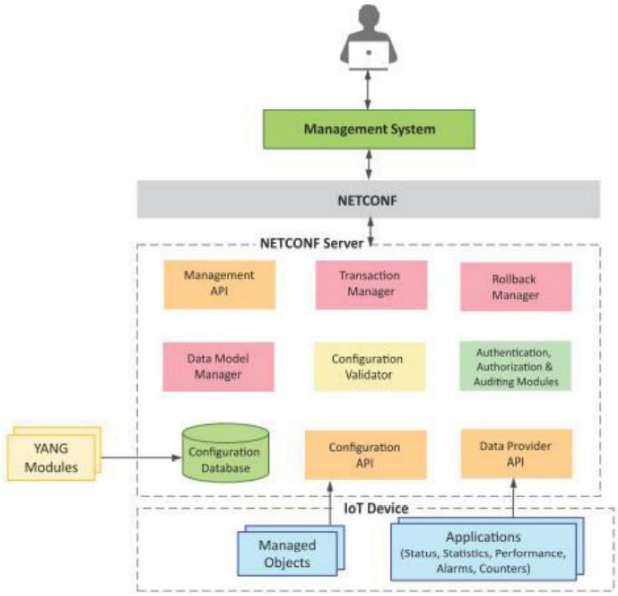
- NETCONF uses XML-encoded Remote Procedure Calls (RPCs) for framing request and response messages.
- The RPC layer provides mechanism for encoding of RPC calls and notifications.
- NETCONF provides various operations to retrieve and edit configuration data from network devices.
- The Content Layer consists of configuration and state data which is XML-encoded.
- The schema of the configuration and state data is defined in a data modeling language called YANG.
- NETCONF provides a clear separation of the configuration and state data.
- The configuration data resides within a NETCONF configuration datastore on the server.

YANG

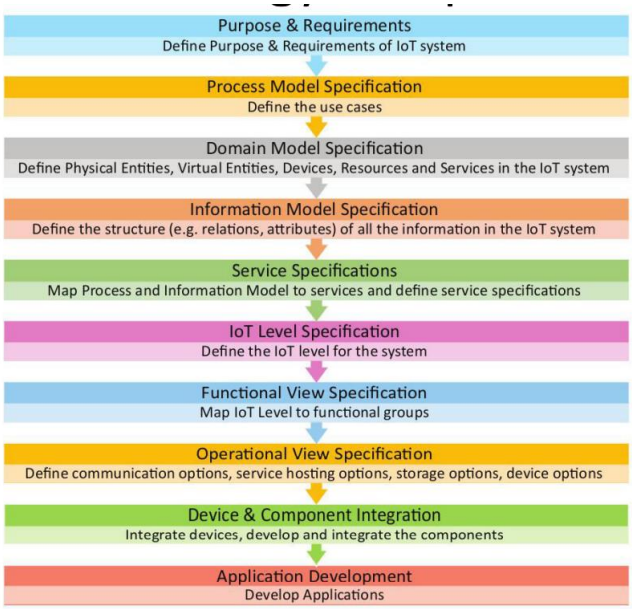
- YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol
- YANG modules contain the definitions of the configuration data, state data, RPC calls that can be issued and the format of the notifications.
- YANG modules defines the data exchanged between the NETCONF client and server.
- A module comprises of a number of 'leaf' nodes which are organized into a hierarchical tree structure.
- The 'leaf' nodes are specified using the 'leaf' or 'leaf-list' constructs.
- Leaf nodes are organized using 'container' or 'list' constructs.
- A YANG module can import definitions from other modules.
- Constraints can be defined on the data nodes, e.g. allowed values.
- YANG can model both configuration data and state data using the 'config' statement.

IoT Systems Management with NETCONF-YANG:

- Management System
- Management API
- Transaction Manager
- Rollback Manager
- Data Model Manager
- Configuration Validator
- Configuration Database
- Configuration API
- Data Provider API



IoT Platforms Design Methodology Steps:



Take an example of home automation case study. Various design methodology steps for an home automation setup are as follows:

Step 1: Purpose & Requirements Specification

The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.

Applying this to our example of a smart home automation system, the purpose and requirements for the system may be described as follows:

Purpose : A home automation system that allows controlling of the lights in a home remotely using a web application.

Behavior : The home automation system should have auto and manual modes. In auto mode, the system measures the light level in the room and switches on the light when it gets dark. In manual mode, the system provides the option of manually and remotely switching on/off the light.

System Management Requirement : The system should provide remote monitoring and control functions.

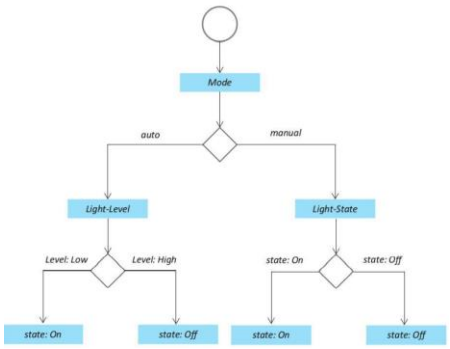
Data Analysis Requirement : The system should perform local analysis of the data.

Application Deployment Requirement : The application should be deployed locally on the device, but should be accessible remotely.

Security Requirement : The system should have basic user authentication capability.

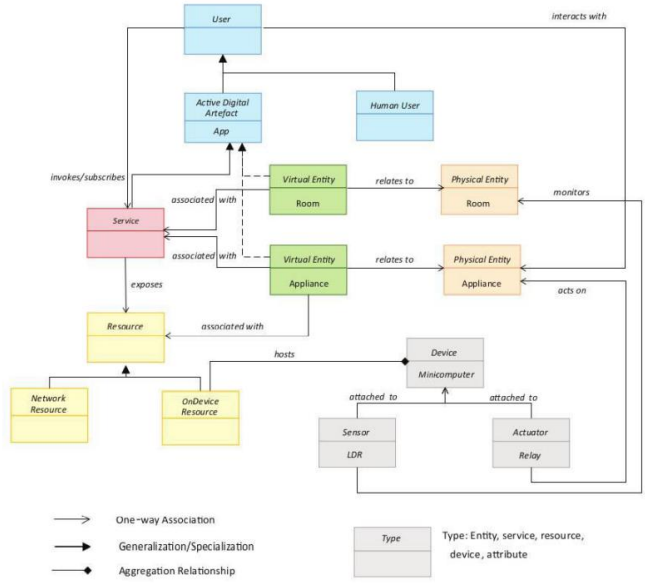
Step 2: Process Specification

The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.



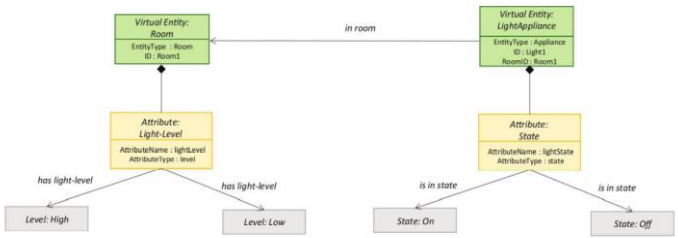
Step 3: Domain Model Specification

The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.



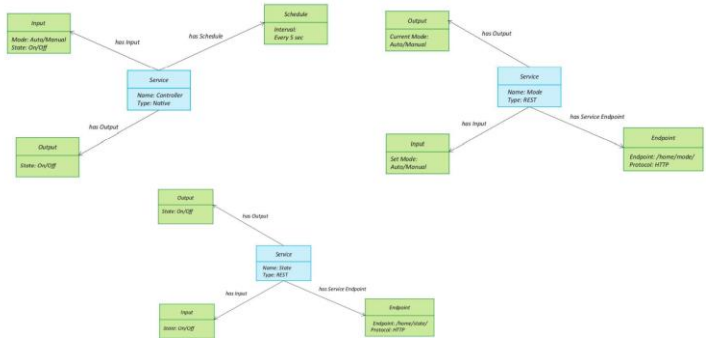
Step 4: Information Model Specification

The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.



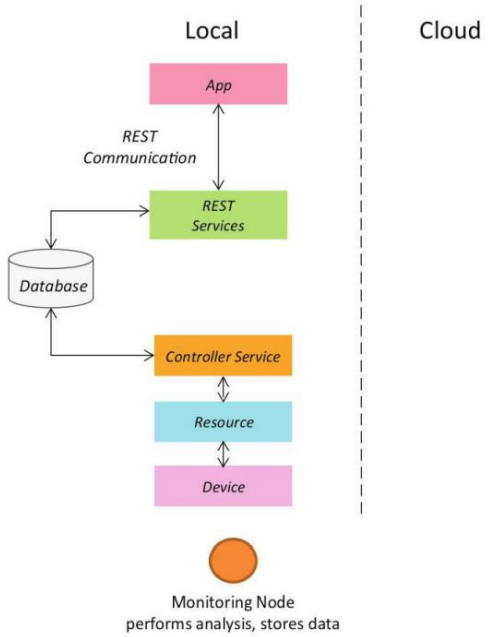
Step 5: Service Specifications

The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.



Step 6: IoT Level Specification

The sixth step in the IoT design methodology is to define the IoT level for the system. In Chapter-1, we defined five IoT deployment levels.



Step 7: Functional View Specification

The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

Step 8: Operational View Specification

The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

Step 9: Device & Component Integration

The ninth step in the IoT design methodology is the integration of the devices and components.

Step 10: Application Development

The final step in the IoT design methodology is to develop the IoT application. For home automation:

- **Auto**
  Controls the light appliance automatically based on the lighting conditions in the room
- **Light**

  When Auto mode is off, it is used for manually controlling the light appliance.

When Auto mode is on, it reflects the current state of the light appliance.

Challenges in Internet of things (IoT)

The Internet of Things (IoT) has fast grown to be a large part of how human beings live, communicate and do business. All across the world, web-enabled devices are turning our global rights into a greater switched-on area to live in.
There are various types of challenges in front of IoT.

Security challenges in IoT :

1. **Lack of encryption –**
   Although encryption is a great way to prevent hackers from accessing data, it is also one of the leading IoT security challenges.
   These drives like the storage and processing capabilities that would be found on a traditional computer.
   The result is an increase in attacks where hackers can easily manipulate the algorithms that were designed for protection.
2. **Insufficient testing and updating –**
   With the increase in the number of IoT(internet of things) devices, IoT manufacturers are more eager to produce and deliver their device as fast as they can without giving security too much of although.
   Most of these devices and IoT products do not get enough testing and updates and are prone to hackers and other security issues.
3. **Brute forcing and the risk of default passwords –**
   Weak credentials and login details leave nearly all IoT devices vulnerable to password hacking and brute force.
   Any company that uses factory default credentials on their devices is placing both their business and its assets and the customer and their valuable information at risk of being susceptible to a brute force attack.
4. **IoT Malware and ransomware –**
   Increases with increase in devices.
   Ransomware uses encryption to effectively lock out users from various devices and platforms and still use a user's valuable data and info.
   Example –
   A hacker can hijack a computer camera and take pictures.
   By using malware access points, the hackers can demand ransom to unlock the device and return the data.
5. **IoT botnet aiming at cryptocurrency –**
   IoT botnet workers can manipulate data privacy, which could be massive risks for an open Crypto market. The exact value and creation of cryptocurrencies code face danger from mal-intentioned hackers.
   The blockchain companies are trying to boost security. Blockchain technology itself is not particularly vulnerable, but the app development process is.

Design challenge in IoT :

1. **Battery life is a limitation –**
   Issues in packaging and integration of small-sized chip with low weight and less power consumption. If you've been following the mobile space, you've likely see how every yr it looks like there's no restriction in terms of display screen size. Take the upward thrust of 'phablets', for instance, which can be telephones nearly as huge as tablets. Although helpful, the bigger monitors aren't always only for convenience, rather, instead, display screen sizes are growing to accommodate larger batteries. Computers have getting slimmer, but battery energy stays the same.

2. **Increased cost and time to market –**
   Embedded systems are lightly constrained by cost.
   The need originates to drive better approaches when designing the IoT devices in order to handle the cost modelling or cost optimally with digital electronic components.
   Designers also need to solve the design time problem and bring the embedded device at the right time to the market.
3. **Security of the system –**
   Systems have to be designed and implemented to be robust and reliable and have to be secure with cryptographic algorithms and security procedures.
   It involves different approaches to secure all the components of embedded systems from prototype to deployment.

Deployment challenges in IoT :

1. **Connectivity –**
   It is the foremost concern while connecting devices, applications and cloud platforms.
   Connected devices that provide useful front and information are extremely valuable. But poor connectivity becomes a challenge where IoT sensors are required to monitor process data and supply information.
2. **Cross platform capability –**
   IoT applications must be developed, keeping in mind the technological changes of the future.
   Its development requires a balance of hardware and software functions.
   It is a challenge for IoT application developers to ensure that the device and IoT platform drivers the best performance despite heavy device rates and fixings.
3. **Data collection and processing –**
   In IoT development, data plays an important role. What is more critical here is the processing or usefulness of stored data.
   Along with security and privacy, development teams need to ensure that they plan well for the way data is collected, stored or processed within an environment.
4. **Lack of skill set –**
   All of the development challenges above can only be handled if there is a proper skilled resource working on the IoT application development.
   The right talent will always get you past the major challenges and will be an important IoT application development asset.