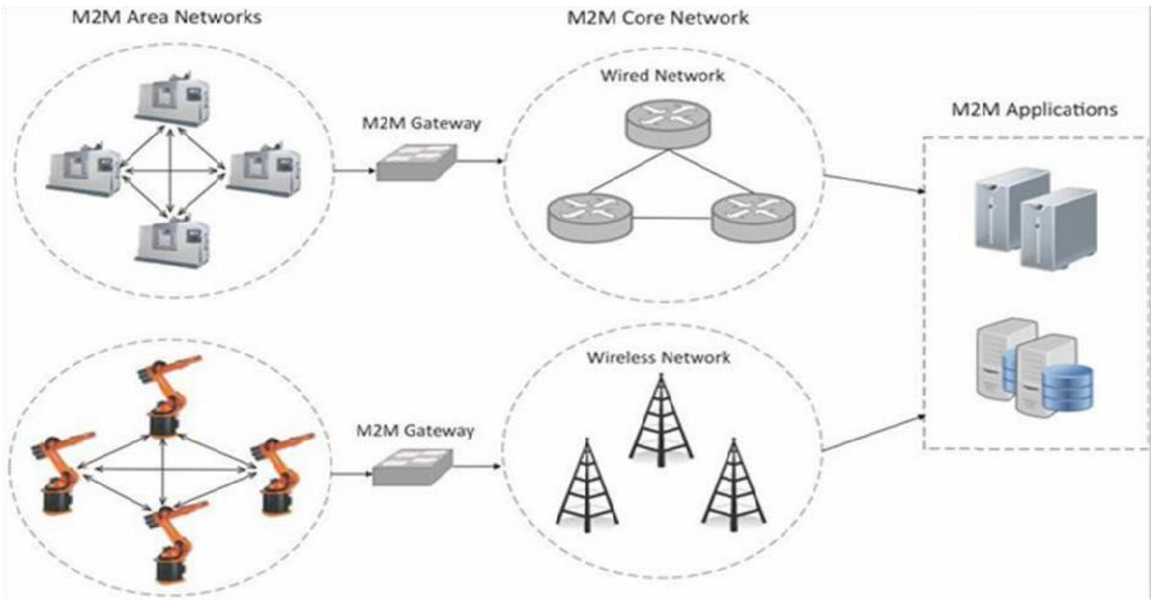
UNIT 3

M2M

Machine-to-Machine(M2M)referstonetworkingofmachines(ordevices)forthepurpose of remote monitoring and control and data exchange.

- Termwhichisoftensynonymouswith IoTisMachine-to-Machine(M2M).
- IoTandM2Mareoftenused interchangeably.

Fig.Showstheend-to-endarchitectureofM2MsystemscomprisesofM2Mareanetworks, communication networks and application domain.



- An M2M area network comprises of machines(or M2M nodes) which have embedded network modules for sensing, actuation and communicating various communication protocols can beusedfor M2M LAN such as ZigBee, Bluetooth, M-bus, Wireless M-Bus etc., These protocols provide connectivity between M2M nodes within an M2M area network.
- The communication network provides connectivity to remote M2M area networks. The communication network provides connectivity to remote M2M area network. The communication network can use either wired or wireless network(IP based). While the M2M are networks use either properietorary or non-IP based communication protocols, the communication network uses IP-based network. Since non-IP based protocols areused within M2M area network, the M2M nodes within one networkcannot communicate with nodes in an external network.
- To enable the communication between remote M2M are network, M2M gateways are used.

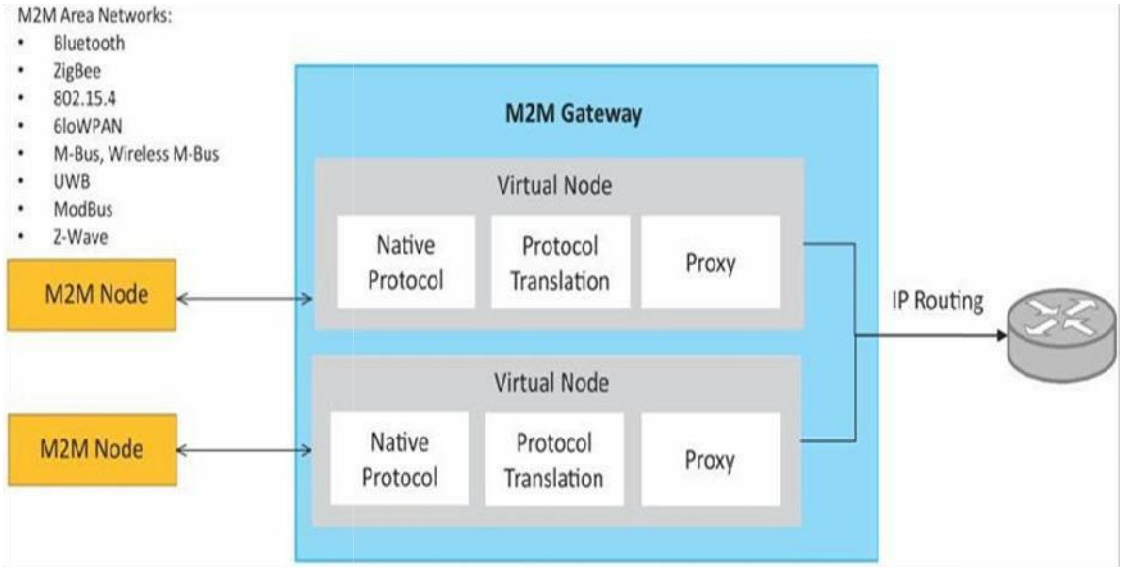


Fig. Shows a block diagram of an M2M gateway. The communication between M2M nodes and the M2M gateway is based on the communication protocols which are naive to the M2M are network. M2M gateway performs protocol translations to enable Ip-connectivity for M2M are networks.M2M gatewayacts as aproxyperformingtranslationsfrom/tonativeprotocolsto/from Internet Protocol(IP). With an M2M gateway, each mode in an M2M area network appears as a virtualized node for external M2M area networks.

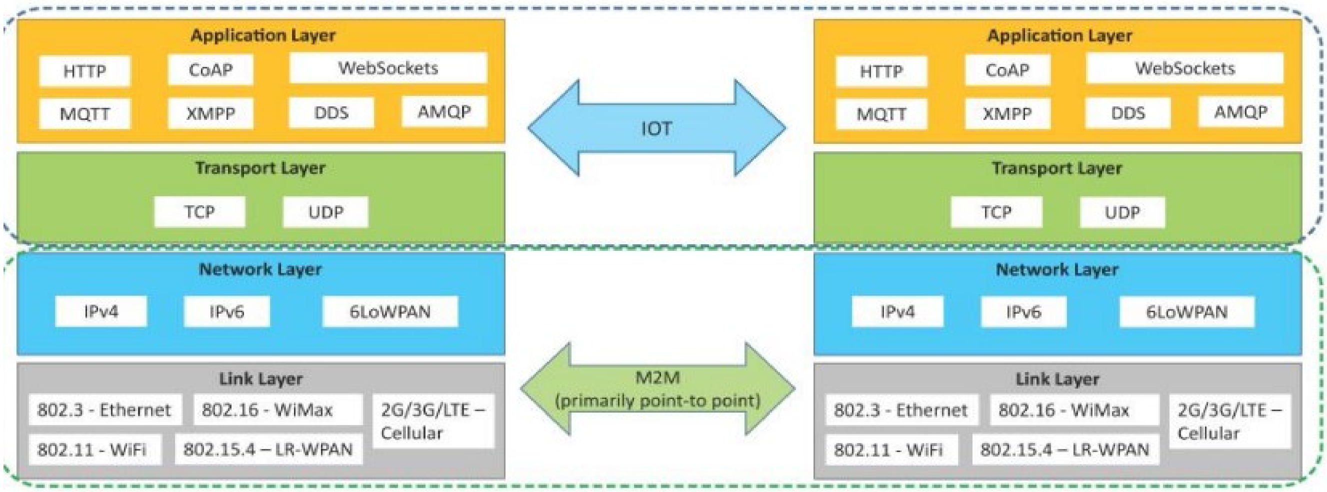
DifferencesbetweenIoTandM2M

1) **CommunicationProtocols:**
    - CommonlyusesM2MprotocolsincludeZigBee,Bluetooth,ModBus,M-Bus, Wireless M-Bus tec.,
    - InIoTusesHTTP,CoAP,WebSocket,MQTT,XMPP,DDS,AMQPetc.,
2) **MachinesinM2MVsThingsinIoT:**
    - MachinesinM2MwillbehomogenouswhereasThingsinIoTwillbe heterogeneous.
3) **HardwareVsSoftwareEmphasis:**
    - theemphasisofM2Mismoreonhardwarewithembeddedmodules,theemphasis ofIoT is more on software.
4) **DataCollection&Analysis**
    - M2Mdataiscollectedinpointsolutionsandofteninon-premisesstorage infrastructure.
    - ThedatainIoTiscollectedinthecloud(canbepublic,privateorhybridcloud).

5) **Applications**

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

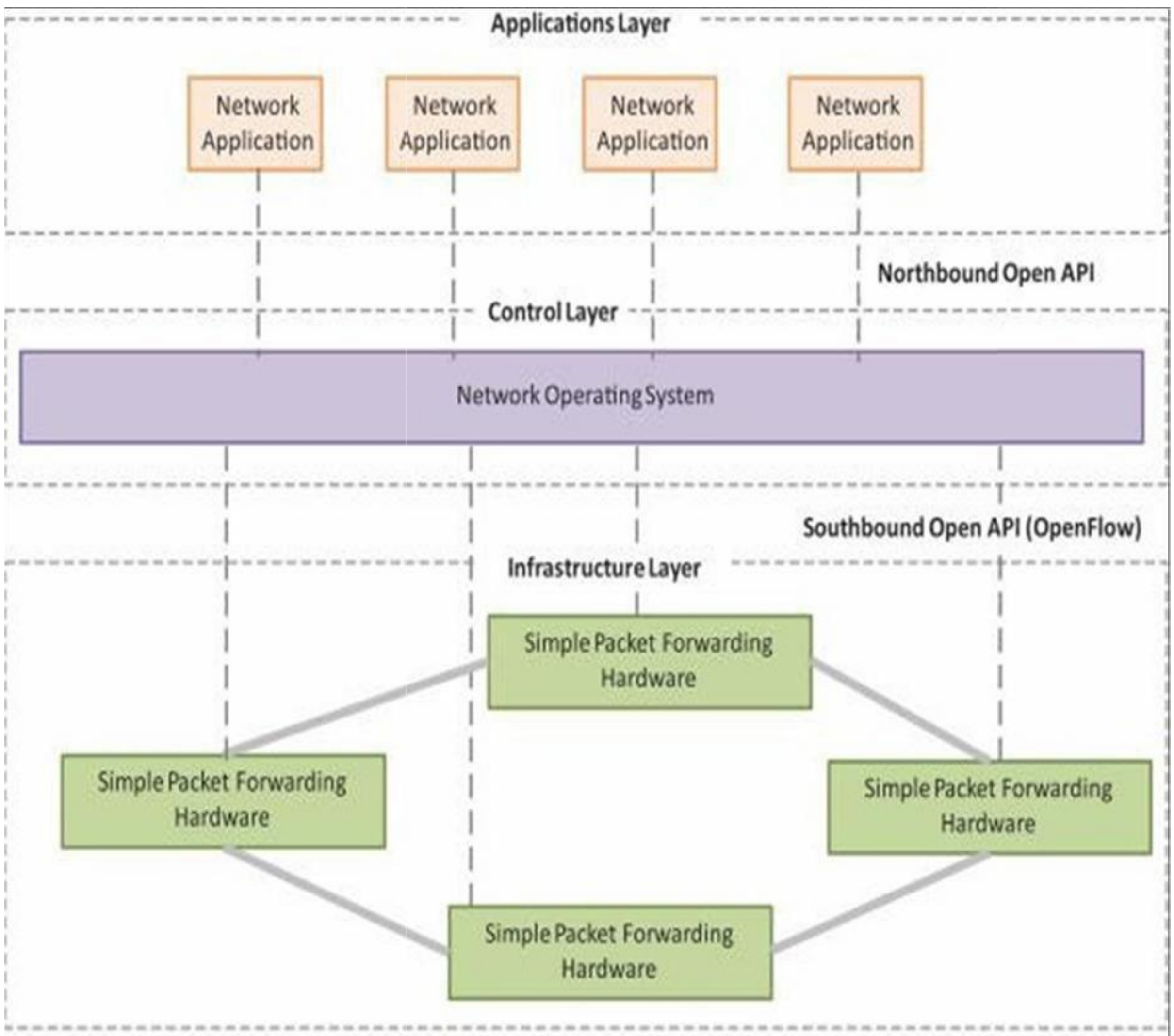
Communication in IoT vs M2M



SDNandNVFforIoT

SoftwareDefinedNetworking(SDN):

- Software-Defined Networking (SDN)is a networking architecturethatseparatesthe control plane from the data plane and centralizes the network controller.
- Software-basedSDNcontrollersmaintainaunitedviewofthenetwork
- The underlying infrastructure in SDNuses simple packet forwardinghardwareas opposed to specialized hardware in conventional networks.



SDN Architecture

KeyelementsofSDN:

1) **CentralizedNetworkController**

   With decoupled control and data planes and centralized network controller, the network administrators can rapidly configure the network.

2) **ProgrammableOpenAPIs**

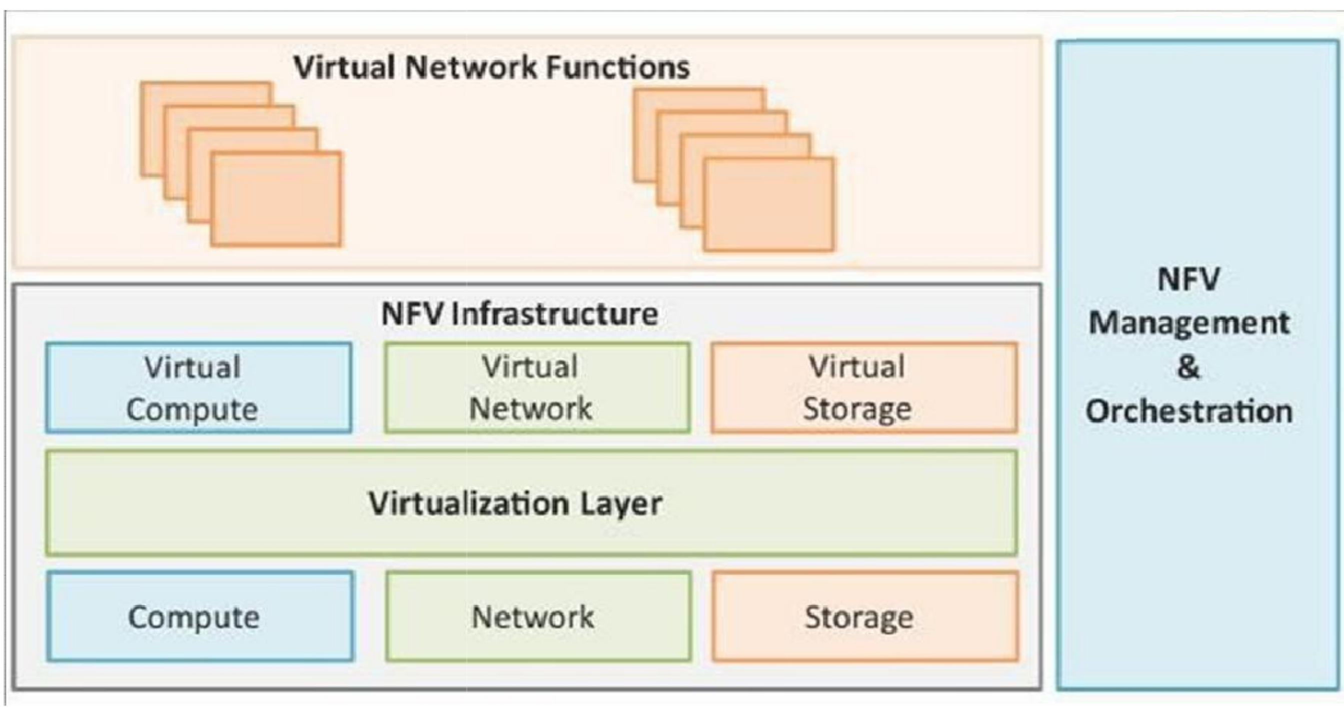
   SDN architecture supports programmable open APIs for interface between the SDN application and control layers (Northbound interface).

3) **StandardCommunicationInterface(OpenFlow)**

   SDN architecture uses a standard communication interface between the control and infrastructure layers (Southbound interface). OpenFlow, which is defined by the Open Networking Foundation (ONF) is the broadly accepted SDN protocolfor the Southbound interface.

NetworkFunctionVirtualization(NFV)

- Network Function Virtualization (NFV) is a technology that leverages virtualization to consolidate the heterogeneous network devices onto industry standard high volume servers, switches and storage.
- NFV is complementaryto SDN as NFV can provide the infrastructure on whichSDN can run.



Keyelementsof NFV:

1) **VirtualizedNetworkFunction(VNF):**

   VNF is a software implementation of a network function which is capable of running over the NFV Infrastructure (NFVI).

2) **NFVInfrastructure(NFVI):**

   NFVIincludescompute,networkandstorageresourcesthatarevirtualized.

3) **NFVManagementandOrchestration:**

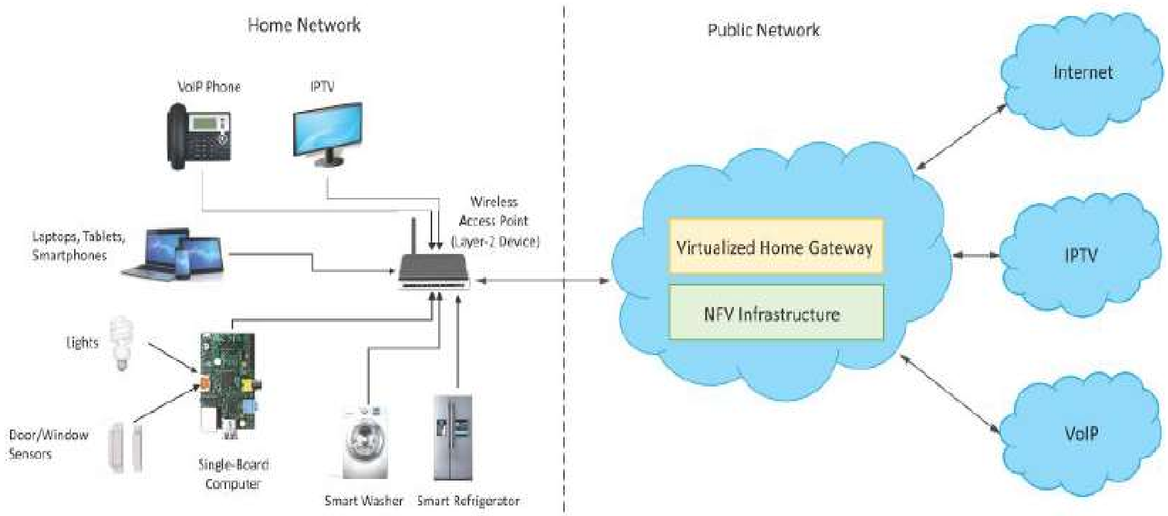
   NFV Management and Orchestration focuses on all virtualization-specific management tasks and covers the orchestration and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of VNFs.

NeedforIoTSystemsManagement
Managingmultipledeviceswithinasinglesystemrequiresadvancedmanagementcapabilities.

1) **AutomatingConfiguration:**IoTsystemmanagementcapabilitiescanhelpin automating the system configuration.
2) **Monitoring Operational & Statistical Data :**Management systems can help in monitoring opeartional and statistical data of a system. This data can be used for fault diagnosis or prognosis.
3) **Improved Reliability:** A management system that allows validating the system configurations before they are put into effect can help in improving the systemreliability.
4) **System Wide Configurations :**For IoT systems that consists of multiple devices or nodes, ensuring system wide configuration can be critical for the correct functioning of the system.
5) **Multiple System Configurations :**For some systems it may be desirable to have multiple valid configurations which are applied at different times or in certainconditions.
6) **Retrieving&ReusingConfigurations:**Managementsystemswhichhavethecapability of retrieving configurations from devices can help in reusing the configurations for other devices of the same type.

NFV Use Case:

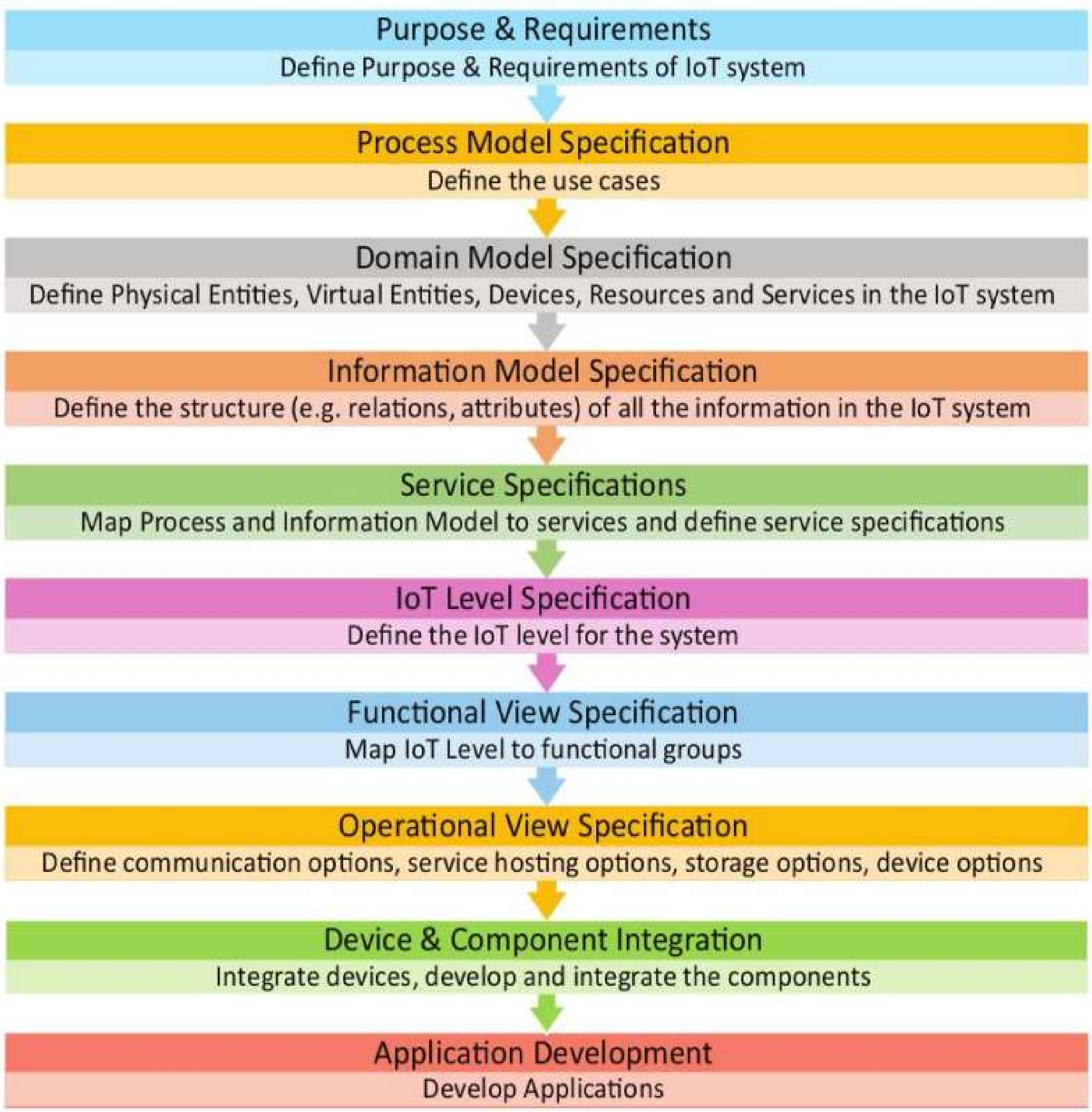
NFV can be used to virtualize the Home Gateway. The NFV infrastructure in the cloud hosts a virtualized Home Gateway. The virtualized gateway provides private IP addresses to the devices in the home. The virtualized gateway also connects to network services such as VoIP and IPTV.



IOT DESIGN METHODOLOGY

<u>IoT Design Methodology that includes:</u>

- Purpose & Requirements Specification
- Process Specification
- Domain Model Specification
- Information Model Specification
- Service Specifications
- IoT Level Specification
- Functional View Specification
- Operational View Specification
- Device & Component Integration
- Application Development



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, behaviour and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...)

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 specification.

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