UNIT-III
IoT Design Methodology

3.1 IoT Design Methodology: Purpose and requirement specification, Process specification, Domain model specification, Information model specification, Service specification, IoT level specification, Device and component integration, Functional view specification, operational view specification, Device and component integration, application development.

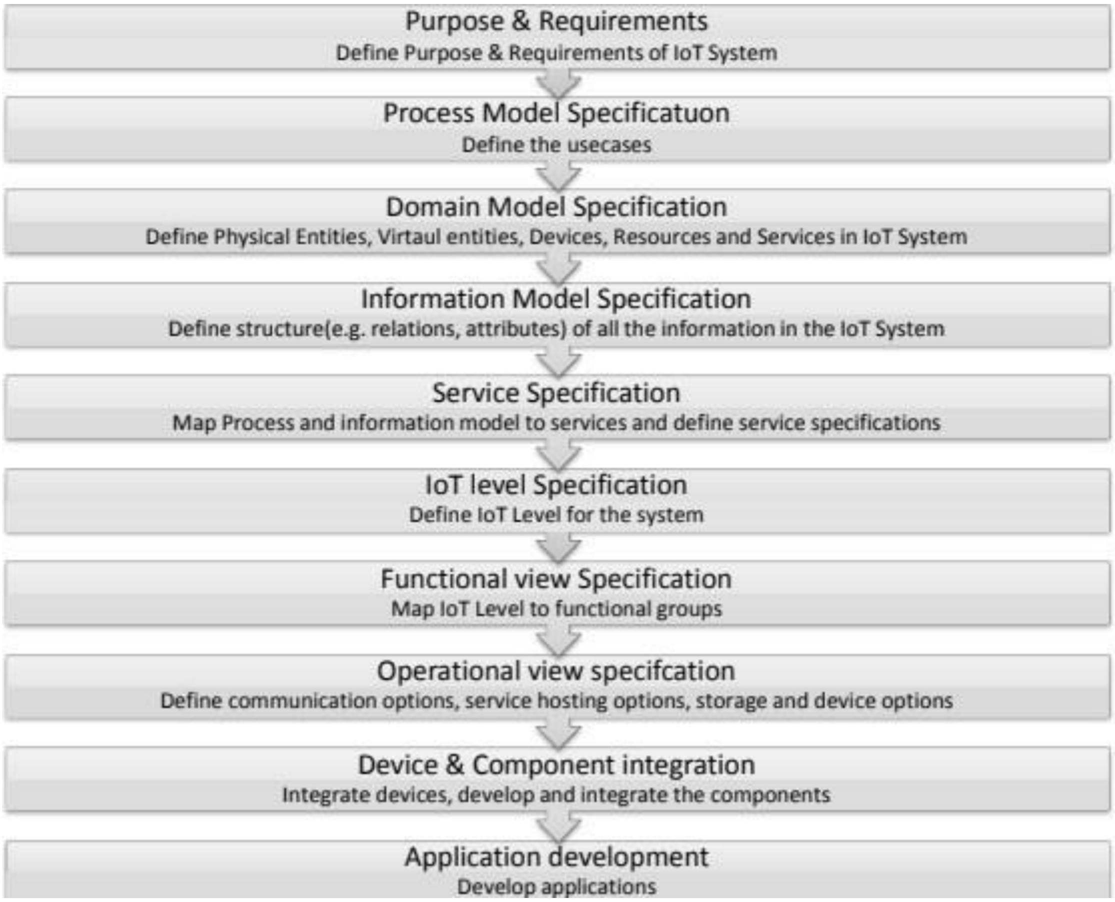
3.2 Logical Designing using programming languages.

3.3 Configuration of hardware platform.

IoT Design Methodology:

Steps involved in IoT System Design Methodology:

1. Purpose and Requirement Specification
2. Process Specification
3. Domain model specification
4. Information Model Specification
5. Service specifications
6. IoT Level Specifications
7. Functional view specification
8. Operational View Specification
9. Device and Component integration
10. Application development



Step 1 : Purpose and Requirement Specification:

First step is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements are captured. Requirements can be:

- Data collection requirements
- Data analysis requirements
- System management requirements
- Security requirements
- User interface requirements

For home automation system the purpose and requirements specification is as follows:

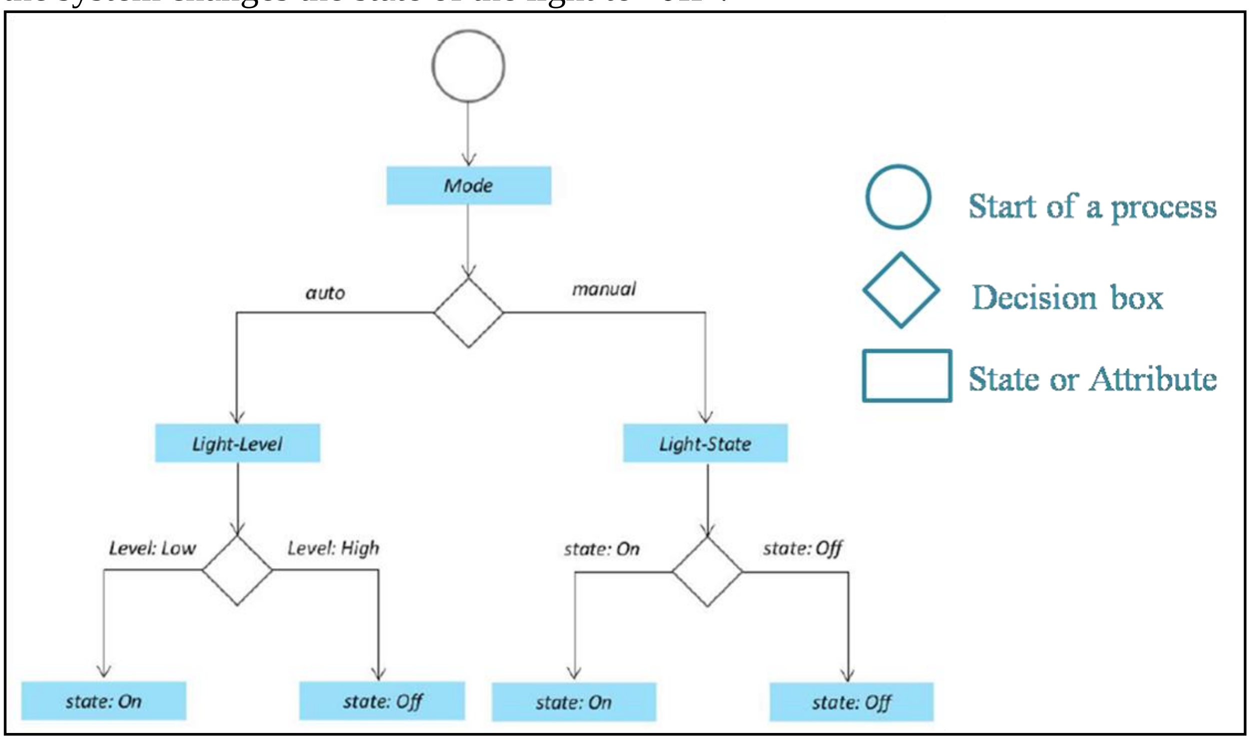
Purpose	A home automation system that allows controlling the lights remotely using a web application
Behavior	Home automation system should support two modes: auto and manual **Auto:** System measures the light level in the room and switches on the light when it is dark **Manual:** Allows remotely switching lights on and off
System Management	System should provide remote monitoring and control functions
Data Analysis	System should perform local analysis of the data

Application Deployment	Application should be deployed locally, but should be accessible remotely
Security	Should provide basic security like user authentication

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. Figure shows the process diagram for the home automation system.

- The process diagram shows the two modes of the system - auto and manual.
- In a process diagram, the circle denotes the start of a process, diamond denotes a decision box and rectangle denotes a state or attribute.
- When the auto mode is chosen, the system monitors the light level. If the light level is low, the system changes the state of the light to "on". Whereas, if the light level is high, the system changes the state of the light to "off".



- When the manual mode is chosen, the system checks the light state set by the user. If the light state set by the user is "on", the system changes the state of light to "on", whereas, if the light state set by the user is "off", the system changes the state of light to "off.

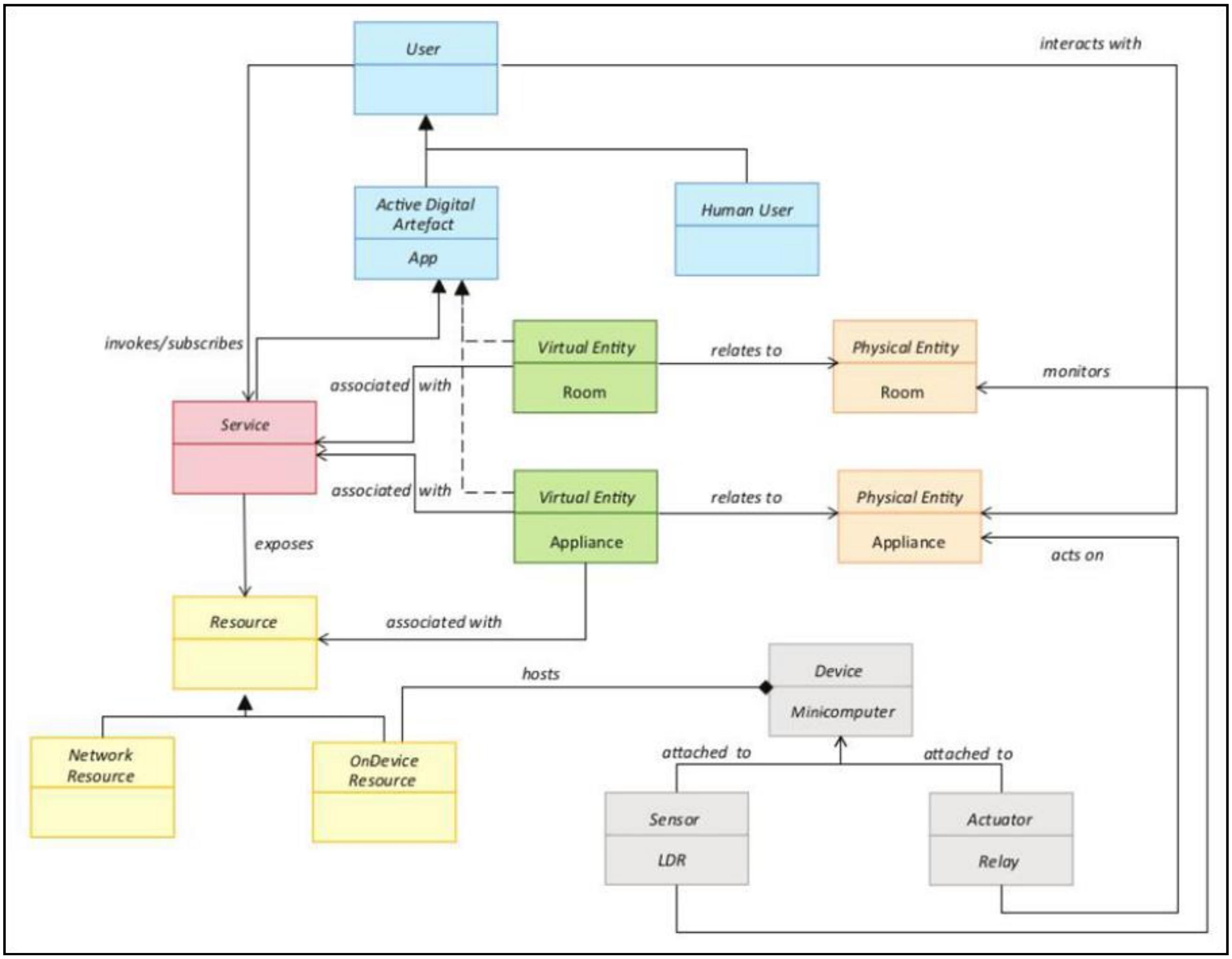
Step 3: Domain Model Specification

- 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.

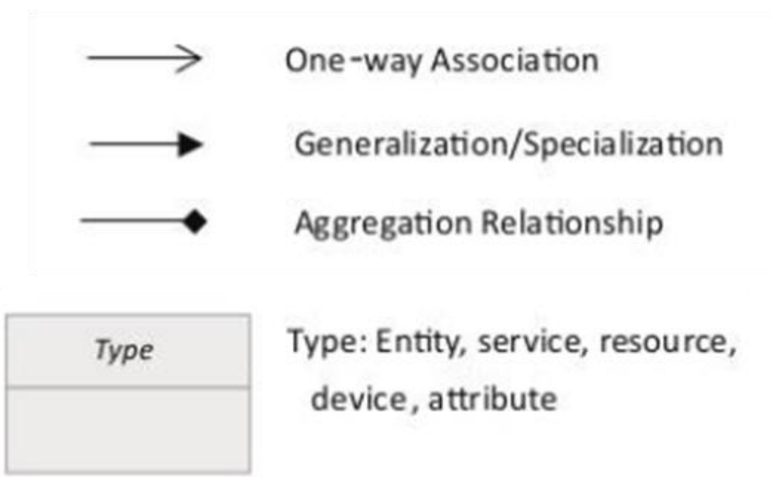
- The domain model describes the main concepts, entities and objects in the domain of the IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. The domain model is independent of any specific technology or platform.

- Using domain model, system designers can get an understanding of the IoT domain for which the system is to be designed. The entities, objects and concepts defined in the domain model of home automation system include the following:

Physical Entity	• The physical identifiable objects in the environment • IoT system provides information about the physical entity (using sensors) or performs actuation upon the physical entity
Virtual Entity	• Virtual entity is a representation of the physical entity in the digital world • For every physical entity there is a virtual entity
Device	• Devices provide a medium for interaction between physical and virtual entities • Devices are used to gather information from or perform actuation on physical entities
Resource	• Resources are software components which can be either on-device or network-resources • On-device resources are hosted on the device and provide sensing or actuation (eg: operating system) • Network-resources include software components that are available on the network (eg: database)
Service	• Services provide an interface for interacting with the physical entity • Services access resources to perform operations on physical entities



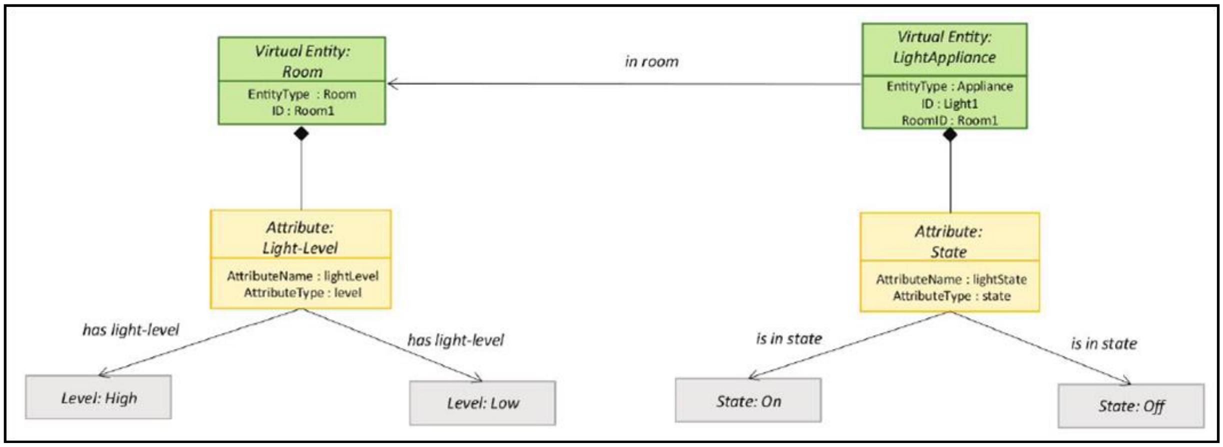
User
interacts with
Active Digital Artefact
App
Human User
invokes/subscribes
Virtual Entity
Room
relates to
Physical Entity
Room
monitors
Service
associated with
associated with
Virtual Entity
Appliance
relates to
Physical Entity
Appliance
acts on
exposes
Resource
associated with
hosts
Device
Minicomputer
Network Resource
OnDevice Resource
attached to
attached to
Sensor
LDR
Actuator
Relay



One-way Association
Generalization/Specialization
Aggregation Relationship
Type
Type: Entity, service, resource, device, attribute

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.

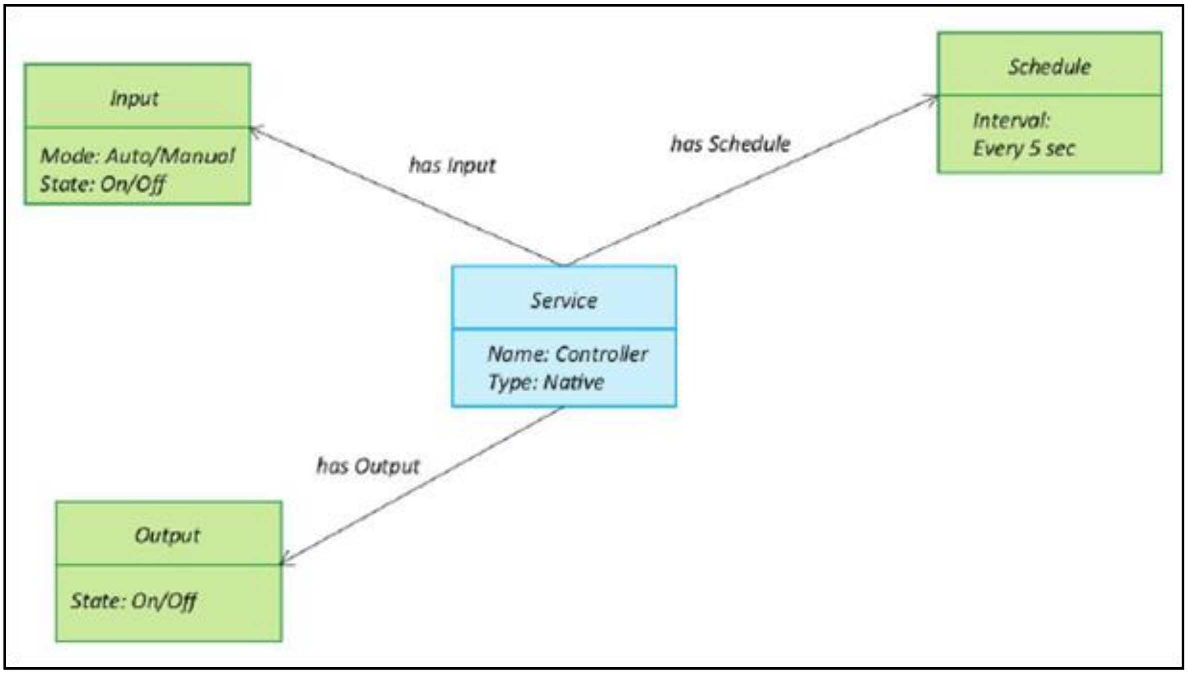


Step 5:Service Specifications

The service specification defines the following:

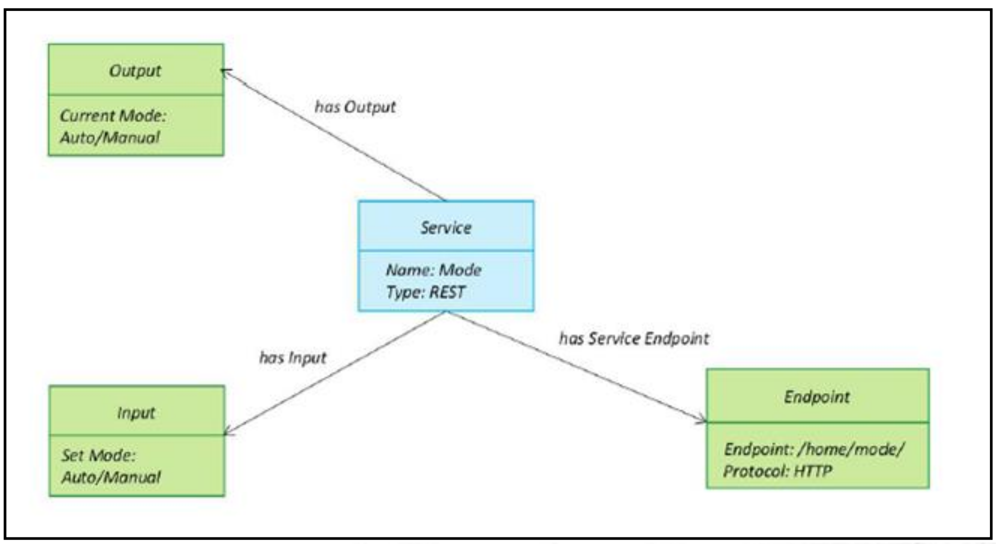
- Services in the system
- Service types
- Service inputs/output
- Service endpoints
- Service schedules
- Service preconditions
- Service effects

- For each state and attribute in the process specification and information model, we define a service. Services either change the state of attributes or retrieve their current values. The service specification for each state in home automation systems are as shown below:
- The Controller service monitors the light level in auto mode and switches the light on/off and updates the status in the status database. In manual mode the controller service, retrieves the current state from the database and switches the light on/off.



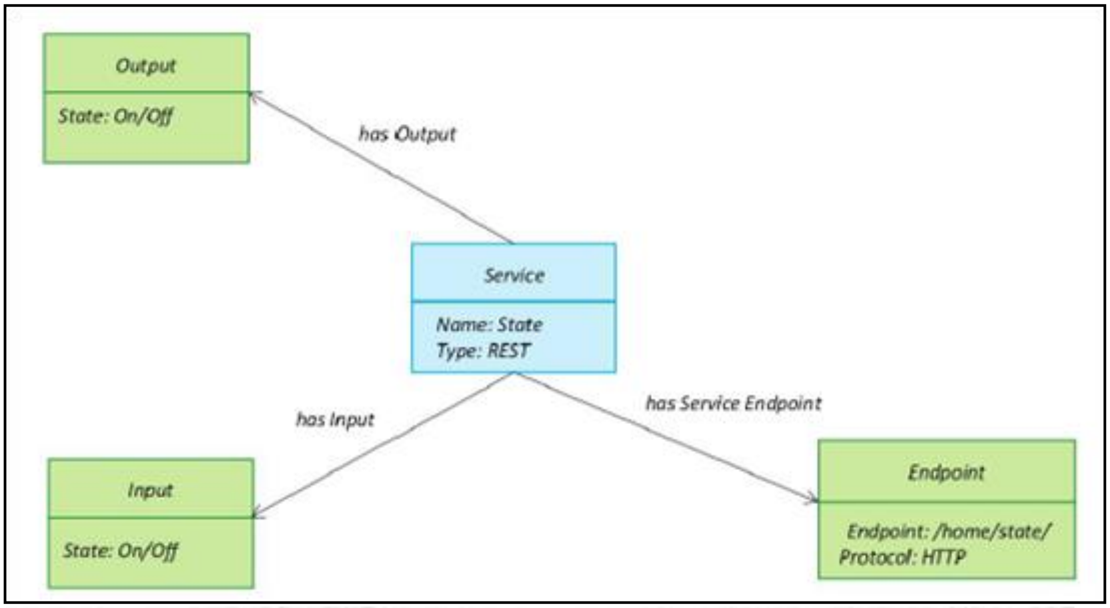
Controller service of the home automation IoT system

- The Mode service sets mode to auto or manual or retrieves the current mode.



Service specification for home automation IoT system - mode service

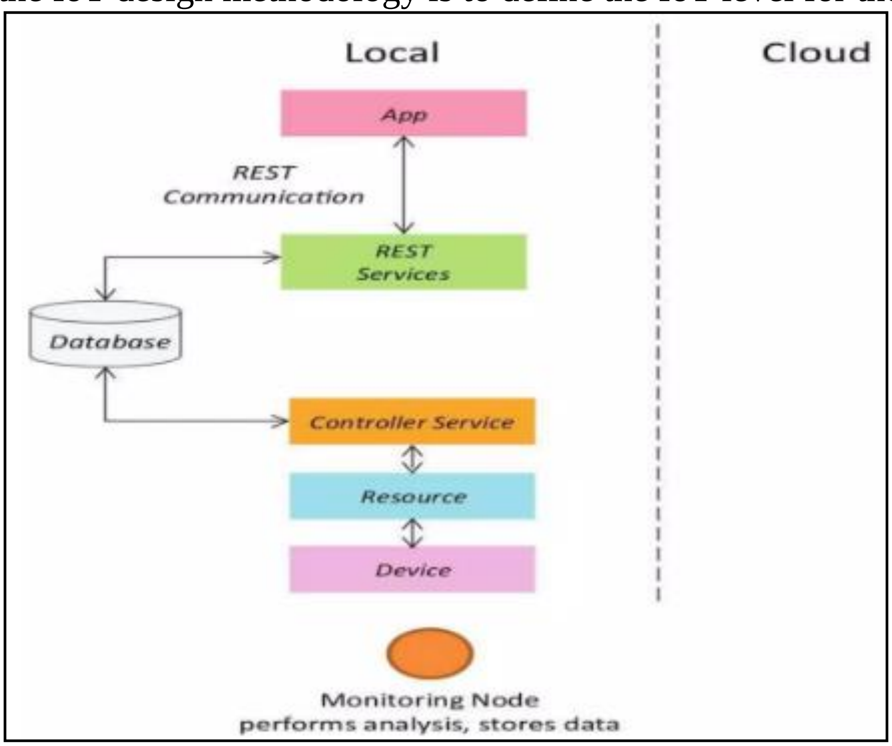
- The State service sets the light appliance state to on/off or retrieves the current light state.



Service specification for home automation IoT system - state service

Step 6: loT Level Specficatlon

The sixth step in the IoT design methodology is to define the IoT level for the system.

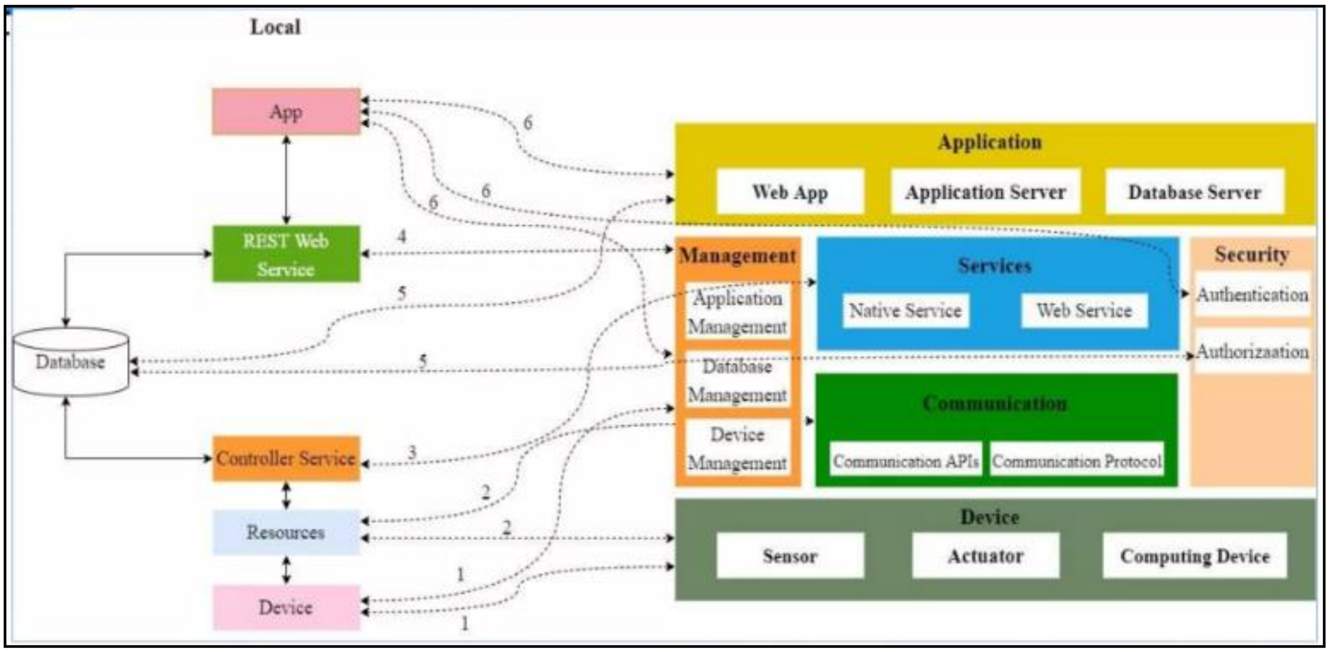


Step 7:Functional view Specification

The Functional Groups (FG) included in a Functional View include:

- Device: The device FG contains devices for monitoring and control. In the home automation example. the device FG includes a single board mini-computer, a light sensor and relay switch(actuator).
- Communication: The communication FG handles the communication for the IoT system. The communication FG includes the communication protocols that form the backbone of IoT systems and enable network connectivity.

- The communication FG also includes the communication APis (such as REST and Web Socket) that are used by the services and applications to exchange data over the network.
- Services: The service FG includes various services involved in the IoT system such as services for device monitoring, device control services, data publishing **services and services for device discovery.**
- Management: The management FG includes all functionalities that are needed to configure and manage the loT system .
- Security: The security FG includes security mechanisms for the IoT system such as authentication, authorization, data security, etc.
- Application: The application FG includes applications that provide an interface to the users to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and the processed data.



Step 8:Operational View Specification

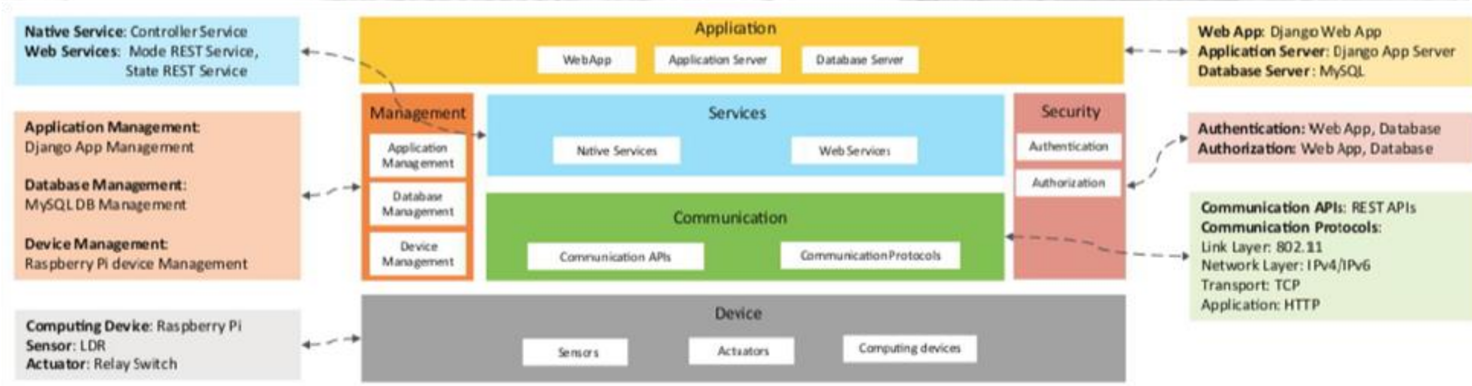
Operational View specifications for the home automation example are as follows:

- Devices: Computing device (Raspberry Pi), light dependent resistor (sensor), relay switch (actuator).
- Communication APIs: REST APIs
- Communication Protocols: Link Layer - 802.11, Network Layer - IPv4/IPv6, Transport - TCP, Application - HTTP.
- Services:
  1. Controller Service - Hosted on device, implemented in Python and run as a native service.
  2. Mode service - REST-ful web service, hosted on device, implemented with

Django-REST Framework.

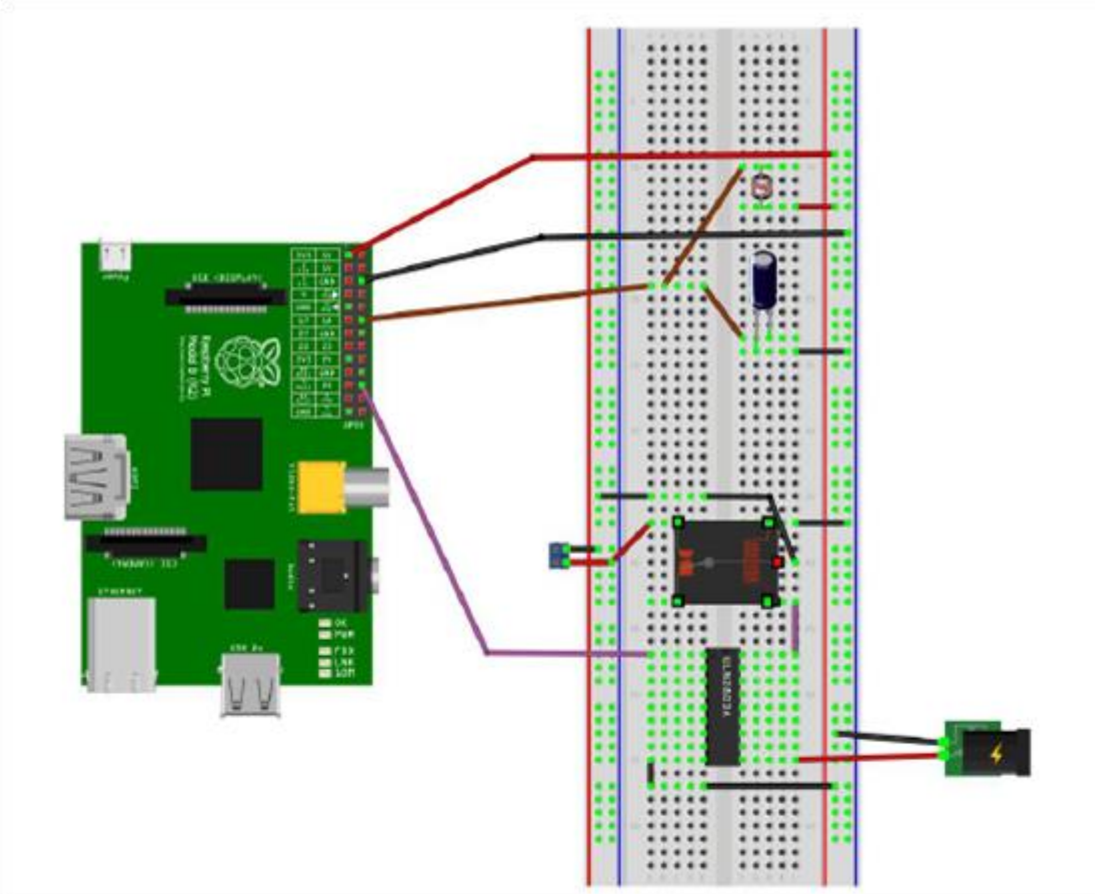
3. State service - REST-ful web service, hosted on device, implemented with Django-REST Framework.

- Application:
  Web Application - Django Web Application,
  Application Server - Django App Server,
  Database Server - MySQL.
- Security:
  Authentication: Web App, Database
  Authorization: Web App, Database
- Management:
  Application Management - Django App Management
  Database Management - MySQL DB Management,
  Device Management - Raspberry Pi device Management.



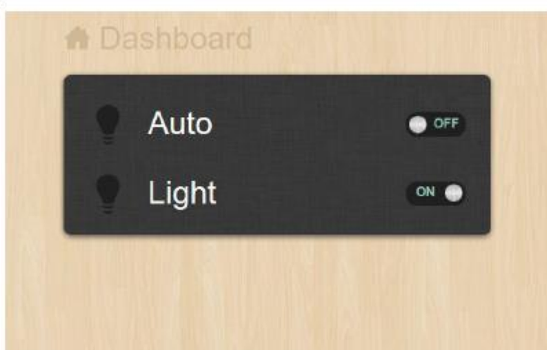
Step 9:Device and Component Integration

In this step the devices like sensors, computing devices and other components are integrated together. The interconnection of different components Raspberry Pi mini computer, LDR sensor and relay switch actuator in home automation system is as shown in the figure given below.



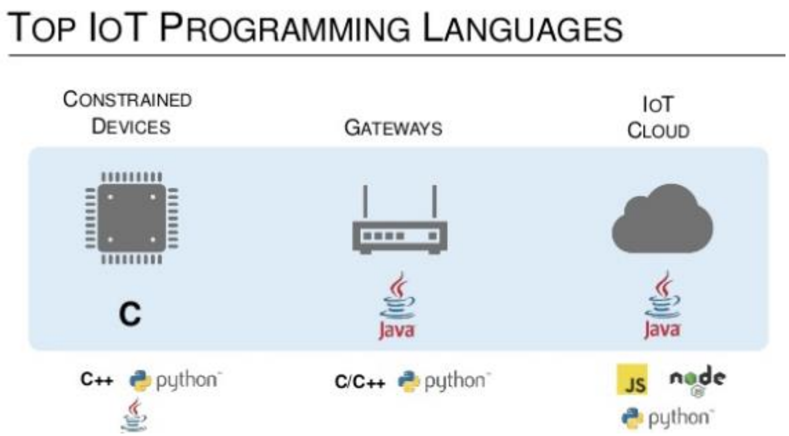
Step 10:Application Development

Using all the information from previous steps, we will develop the application (code) for the IoT system. The application interface for home automation system is shown below.

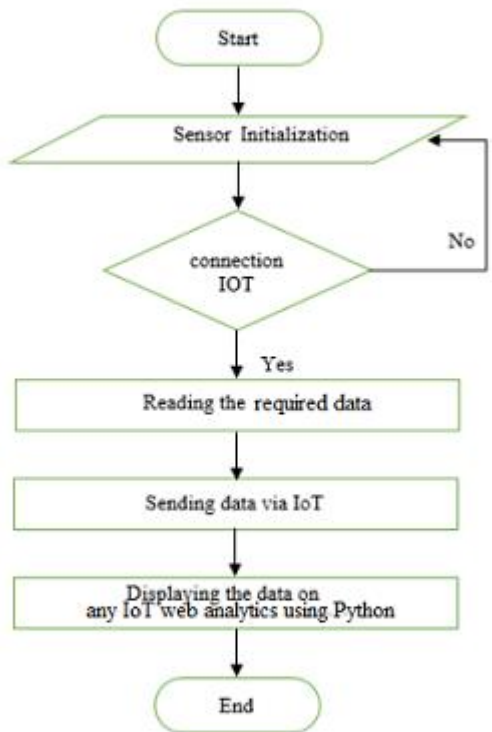


Logical Designing using programming languages.

- Logical design is a high-level design and doesn't provide any detail.
- Logical design can be textual, graphic, or both.
- The logical design of using Python in IoT is easy enough to develop from the standpoint of an IoT system architecture. Using modern frameworks for IoT and modern protocols for transferring data from devices, we can effectively build high-performing IoT systems.
- Even if we rewrite some of your code during production to C, C++, or Java to improve performance, in general, the system will function perfectly in Python.



- Logical Design of any IoT application Flowchart



Common IoT Hardware Platforms:

- Raspberry Pi:
- Arduino (Genuino):
- ESP8266:
- Intel Edison:
- Intel Galileo:
- BeagleBone (BeagleBoard):
- Banana Pi:
- NodeMCU Dev Kit
- Pycom
- Adafruit
- SparkFun
- Espressif