Unit II

Embedded IoT Platform Design Methodology

Outline

- Purpose and Requirement Specification
- Process Specification
- Domain model specification
- Information Model Specification
- Service specifications
- IoT Level Specifications
- Functional view specification
- Operational View Specification
- Device and Component integration
- Application development

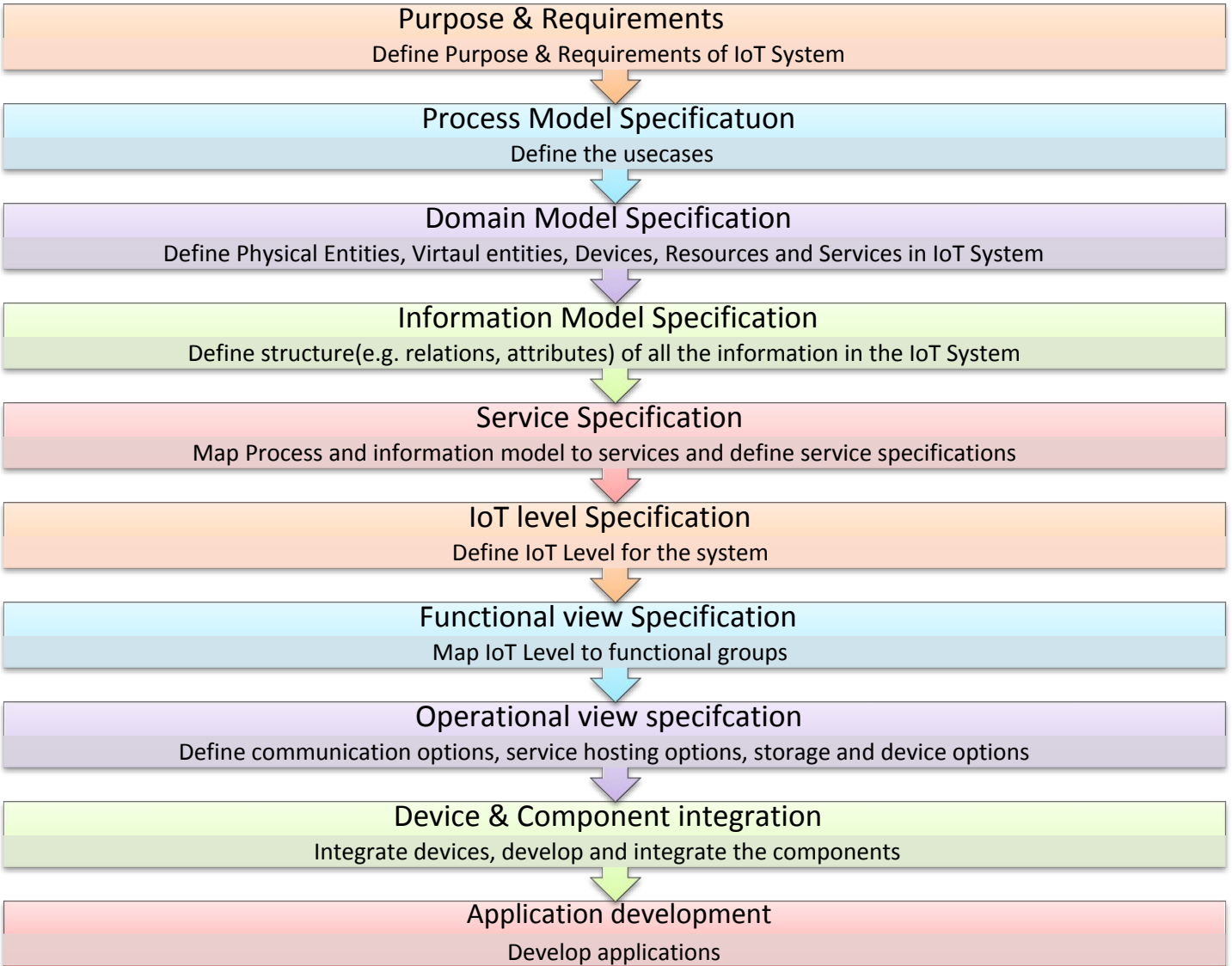
May 2019

- Q.3 a) With the help of diagram list and briefly explain the steps involved in the IoT system design methodology [5M]
- Q.3 b) Explain the application development step of IoT design methodology, consider smart IoT-based home automation system as an example. 5M
- Q. 4 a) Explain the purpose and requirement specification in IoT design methodology, consider smart IoT-based home automation system as an example.[5M]
- Q.4 b) Explain the service specification step of IoT design methodology, consider smart IoT-based home automation system as an example. 5M

Mar 2018

- Q3) a) Explain purpose and requirements specifications step of IoT system design methodology, consider smart IoT - based home automation system as an example. [5]
- b) Explain information model specification step of IoT system design methodology, consider smart IoT - based home automation system as an example. [5]
- Q4) a) With the help of diagram list and briefly explain the steps involved in the IoT system design methodology. [5]
- b) Explain process model specification step of IoT system design methodology, consider smart IoT-based home automation system as an example. [5]

Steps involved in IoT System Design Methdology



Advantages of Using Design methodology

- Reducing the design, testing and maintenance time
- Provide better interoperability
- Reduce the complexity

Step 1 : Purpose and Requirement Specification

Defines

- **System purpose**
- **behavior and Requirements** (such as **data collection** requirements,
- **data analysis** requirement,
- **system management** requirements,
- data **privacy and security** requirements,
- **User interfaces** requirements)

Step 1 : Purpose and Requirement Specification

- **Purpose :** An automated irrigation mechanism which turns the pumping motor ON and OFF on detecting the moisture content of the earth without the intervention of human
- **Behavior :** System should monitor the amount of soil moisture content in soil. In case the soil moisture of the soil deviates from the specified range, the watering system is turned ON/OFF. In case of dry soil, it will activate the irrigation system, pumping water for watering the plants.
- **System Management Requirements :** system should remotely provide monitoring and control functions

Step 1 : Purpose and Requirement Specification

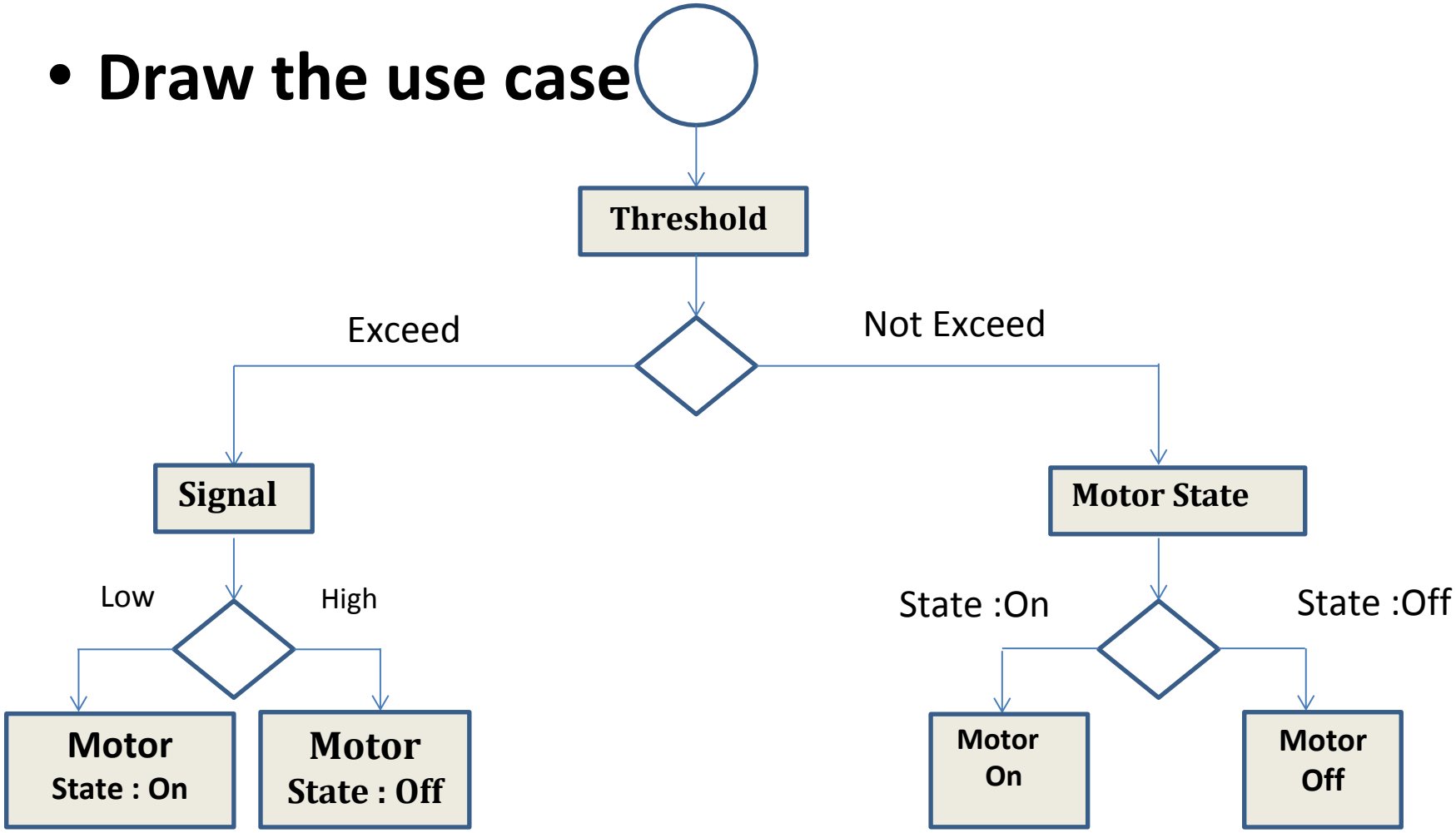
- **Data Analysis Requirements :** system should perform local analysis of data
- **Application Deployment Requirement :** Deployed locally on device, but acts remotely without manual intervention.
- **Security :** Authentication to Use the system must be available

Step 2 : Process Specification

- Define the process with the help of use cases
- The use cases are formally described based on Purpose & requirement specification
- In this use case :
  - **Circle** denotes a state or an attribute

Step 2 : Process Specification

- **Draw the use case**



Step 3 : Domain Model Specification

- **Describes the main concepts, entities and objects** in the domain of IoT system to be designed
- Entities , Objects and Concepts include the following : Physical entity, Virtual entity , Device, Resource, Service

Step 3 : Domain Model Specification

- **Physical Entity:**
  - Discreet identifiable entity in physical environment
  - For eg. Pump, motor, LCD
  - The IoT System provides the information about the physical entity (using sensors) or performs actuation upon the Physical entity(like switching a motor on etc.)
  - In smart irrigation example, there are three Physical entities involved :
    - **Soil (whose moisture content is to be monitored)**
    - **Motor (to be controlled)**
    - **Pump (To be controlled)**
- **Virtual Entity:**
  - Representation of physical entity in digital world
  - For each physical entity there is a virtual entity

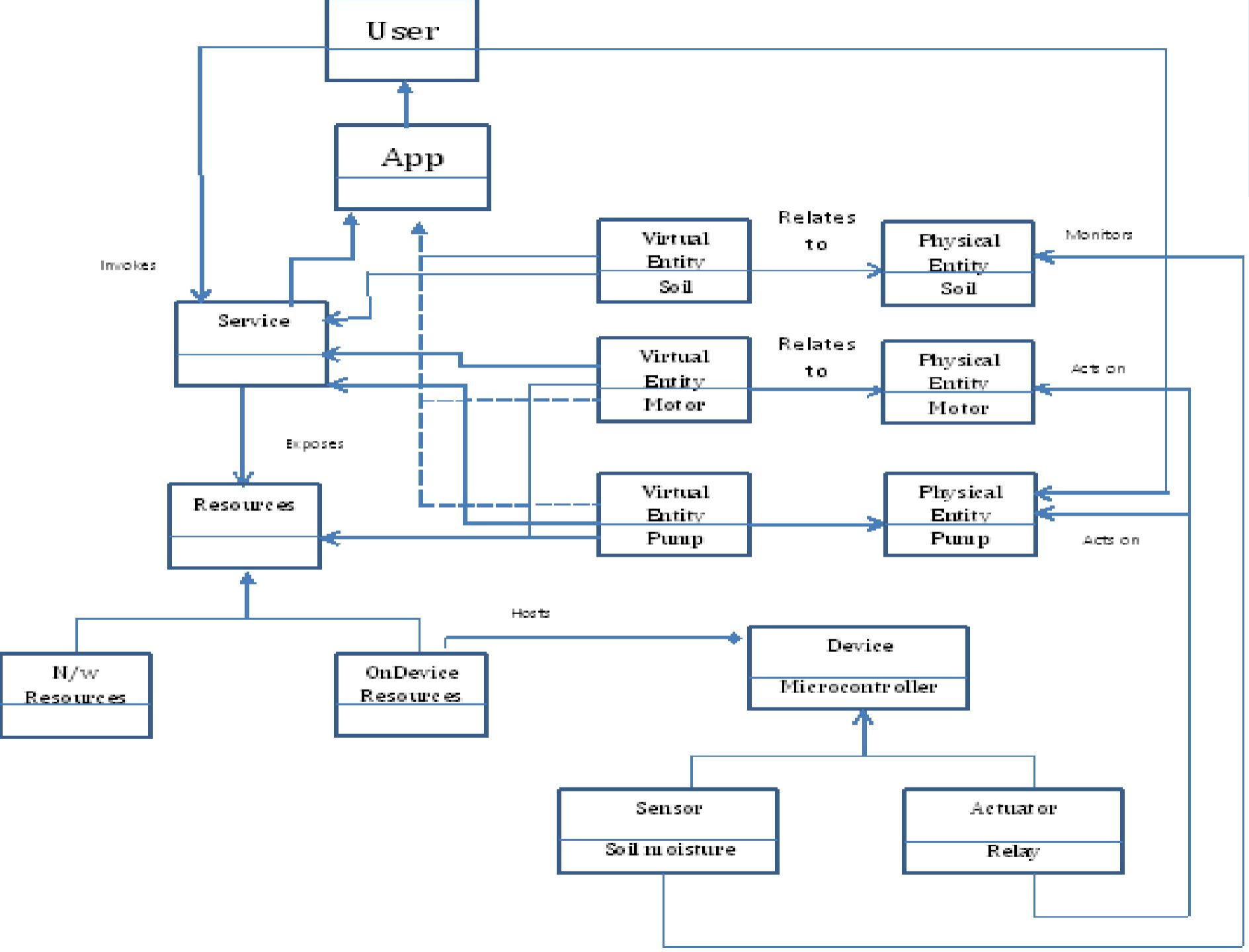
Step 3 : Domain Model Specification

- **Device:**
  - Medium for interactions between Physical and Virtual Entities.
  - Devices (Sensors) are used to gather information from the physical entities
  - Devices are used to identify Physical entities (Using Tags)
  - In Smart Irrigation System, device is soil moisture sensor and buzzer as well as the actuator (relay switch) attached to it.

Step 3 : Domain Model Specification

- In smart irrigation system there are three services :
  - A service that sets the signal to low/ high depending upon the threshold value
  - A service that sets the motor state on/off
  - A controller service that runs and monitors the threshold value of the moisture and switches the state of motor on/off depending upon it.

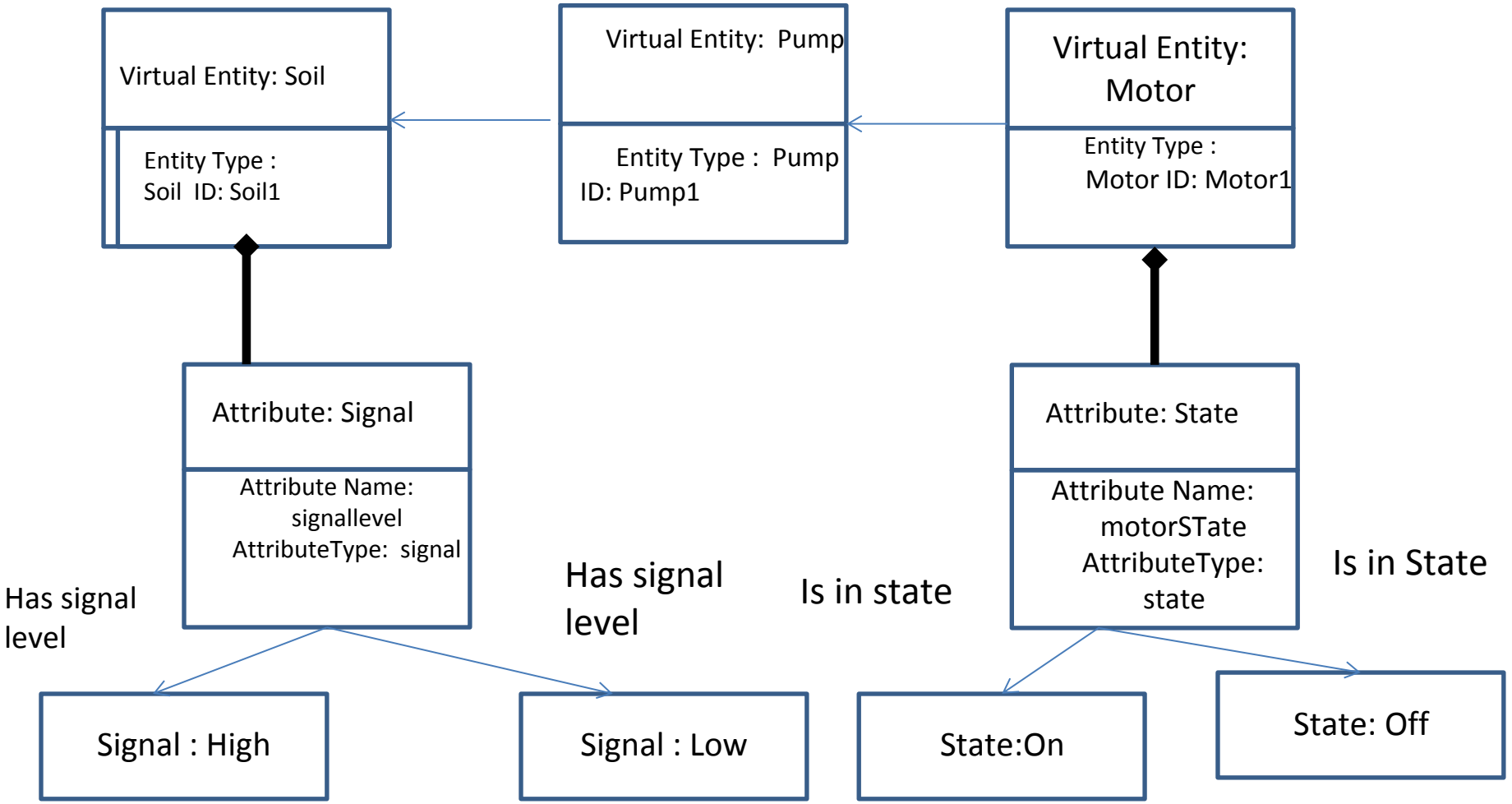
    When threshold value is not crossed the controller retrieves the motor status from database and switches the motor on/off.


User
App
Invokes
Service
Exposes
Resources
Virtual Entity Soil
Relates to
Physical Entity Soil
Monitors
Virtual Entity Motor
Relates to
Physical Entity Motor
Acts on
Virtual Entity Pump
Physical Entity Pump
Acts on
Hosts
N/w Resources
OnDevice Resources
Device
Microcontroller
Sensor
Soil moisture
Actuator
Relay

Step 4 : Information Model Specification

- Defines the structure of all the information in the IoT system (such as attributes, relations etc.)
- It does not describe the specifics of how the information is represented or stored.
- This adds more information to the Virtual entities by defining their attributes and relations
- **I: e, Draw Class diagram**

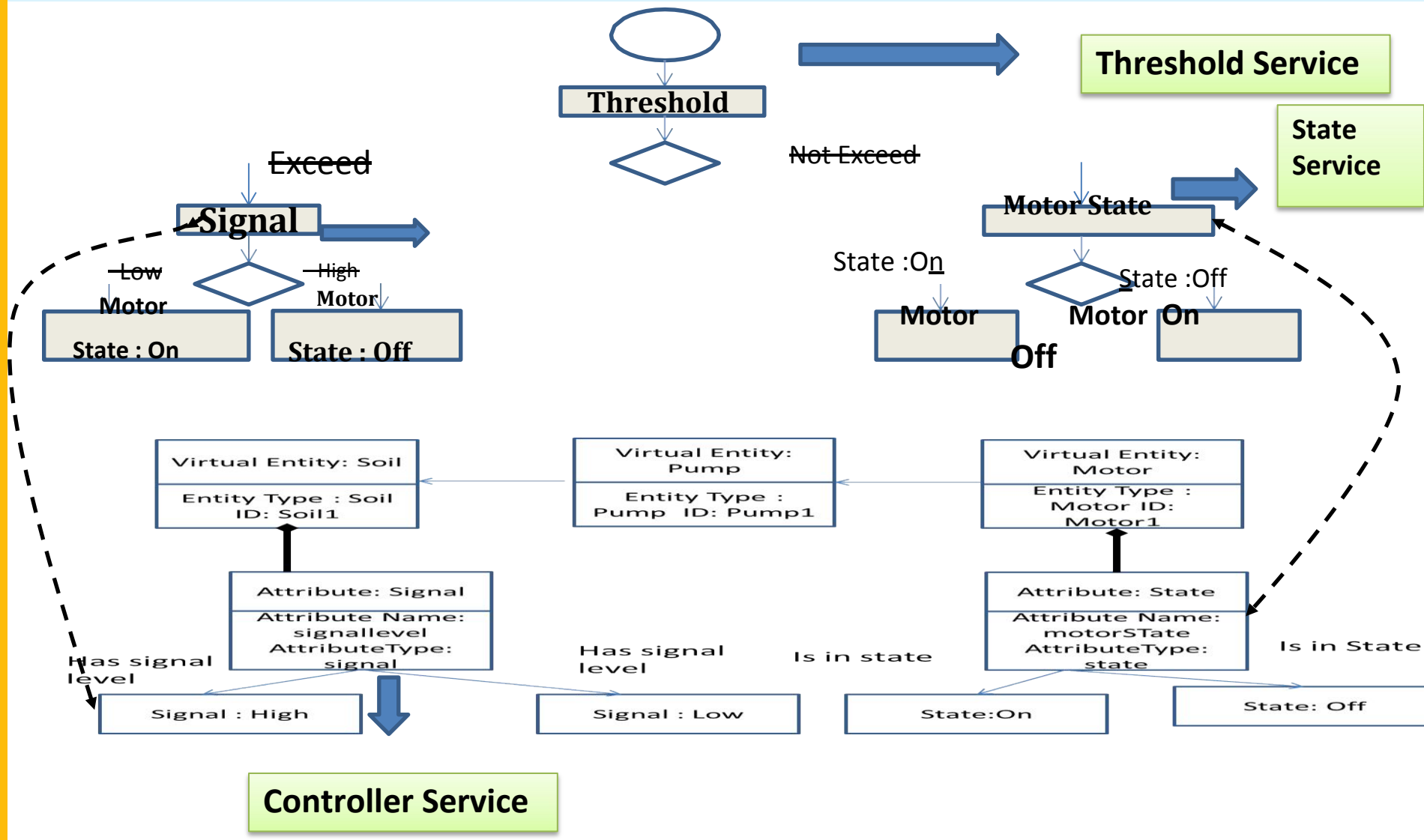
Step 4 : Information Model Specification



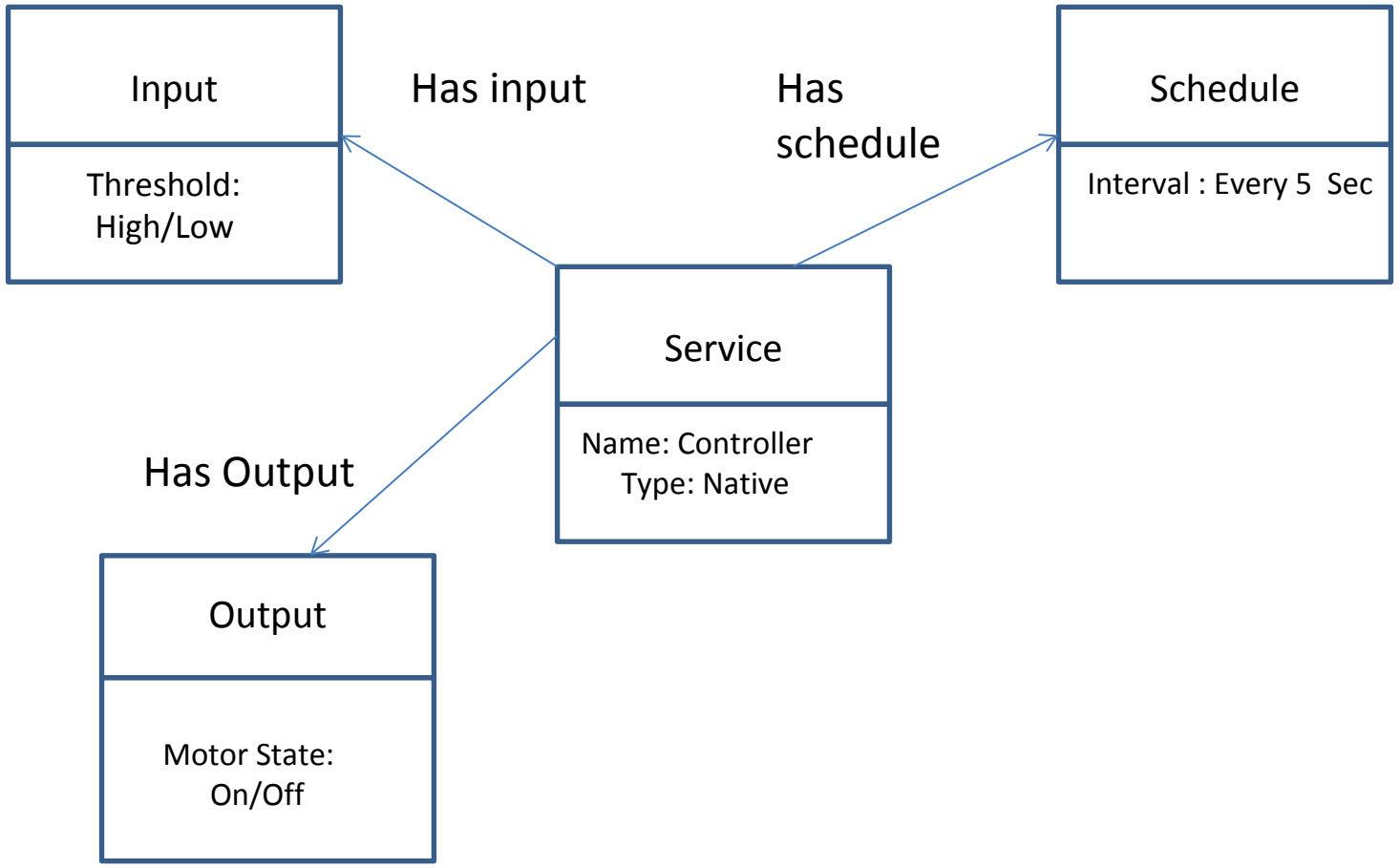
Step 5 : Service Specification

- Define the services in IoT System, service **types**, service **inputs/outputs**, service **endpoints**, service **schedules**, service **preconditions** and service **effects**
- **Services can be controller service, Threshold service, state service, for smart irrigation system**
- These services either change the state/attribute values or retrieve the current vlues.
- For eg.
  - Threshold service sets signal to high or low depending upon the soil moisture value.
  - State service sets the motor state : on or off
  - Controller service monitors the threshold value as well as the motor state and switches the motor on/off and updates the status in the database

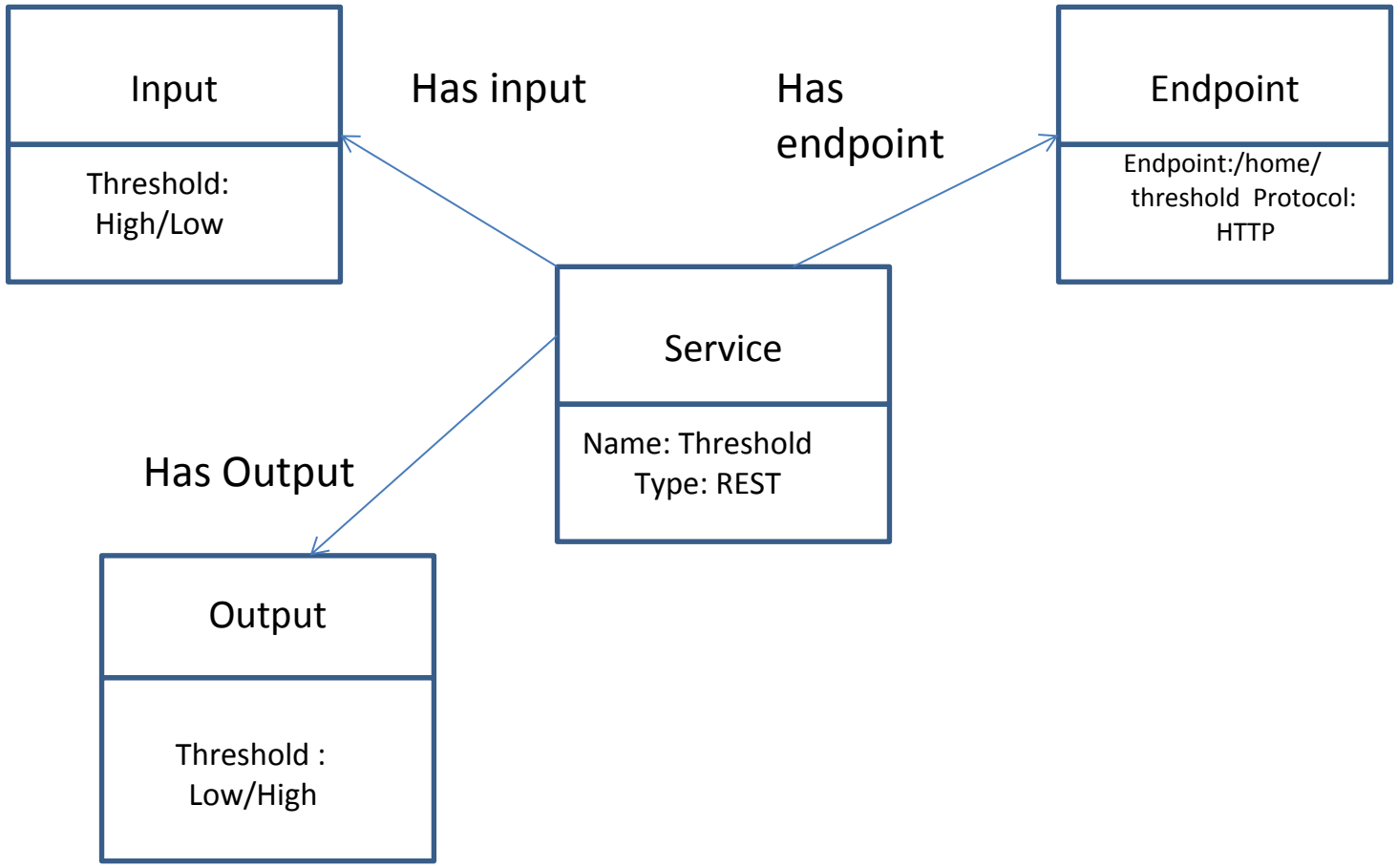
Step 5 : Service Specification



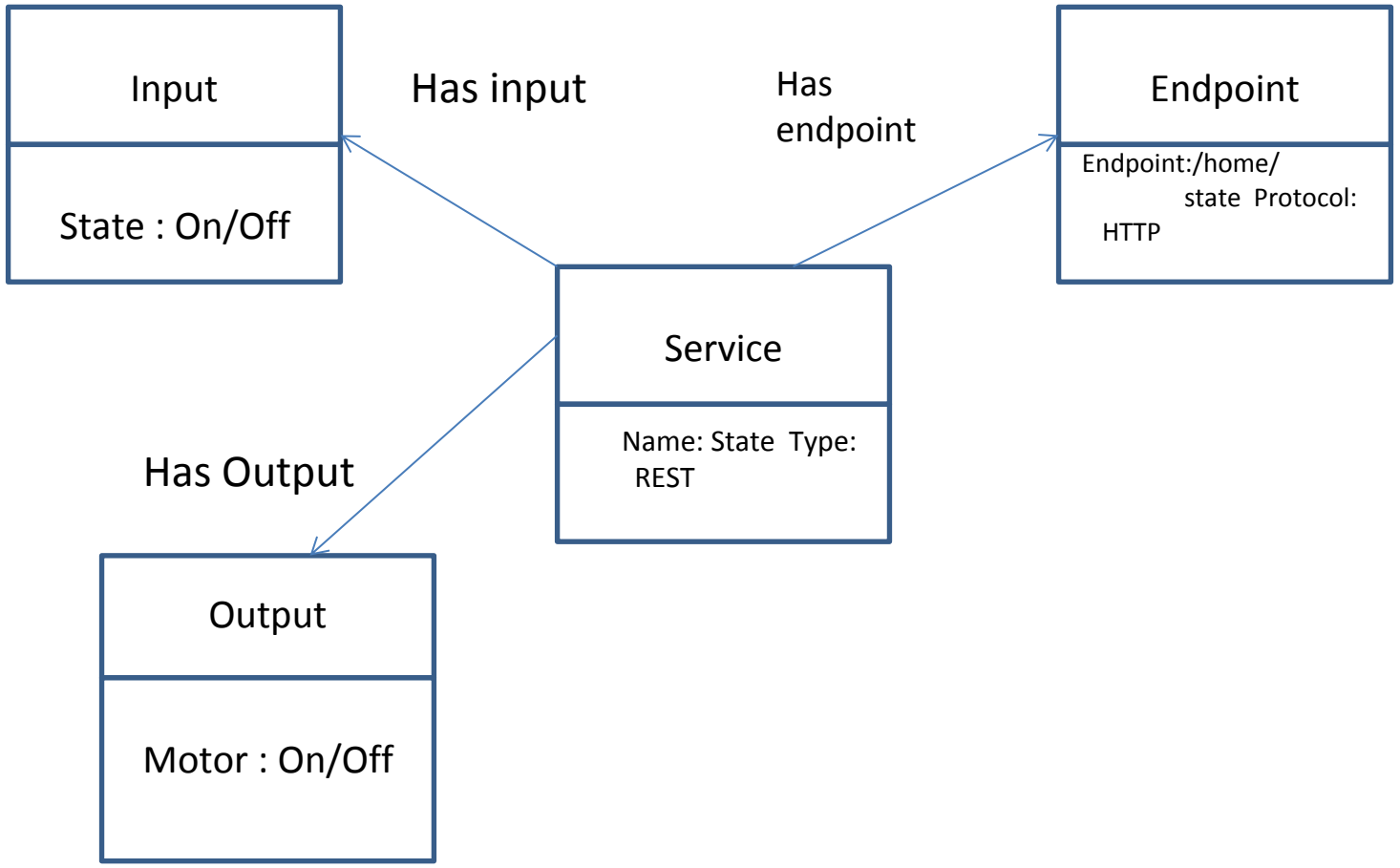
Step 5 : Service Specification ..>Controller Service



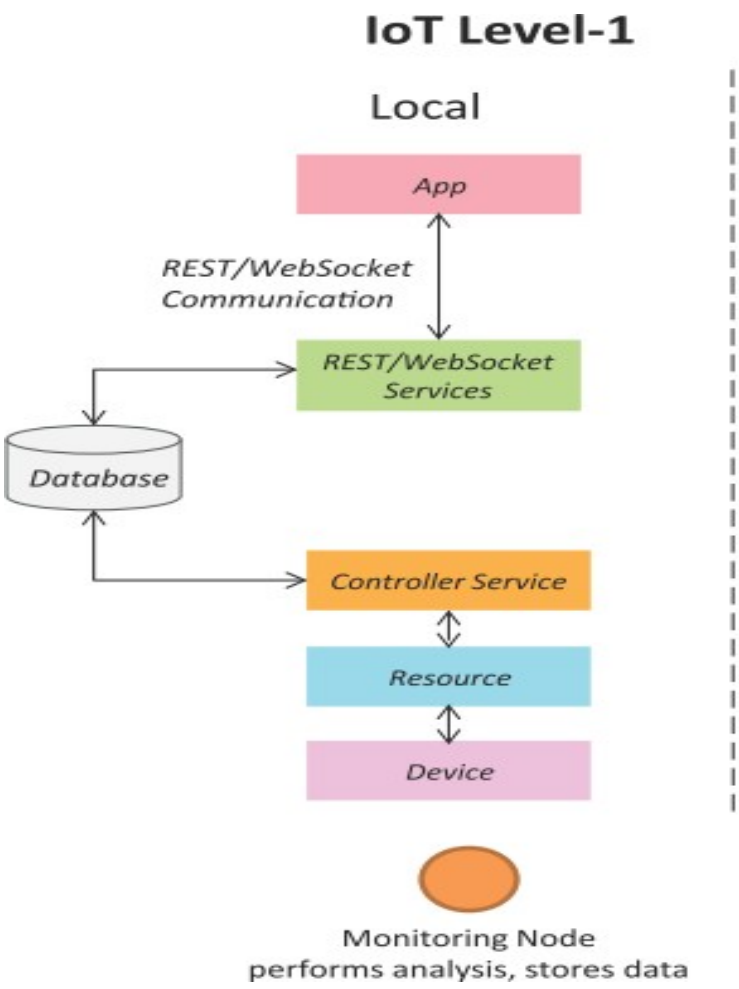
Step 5 : Service Specification
..>Threshold Service



Step 5 : Service Specification ..>State Service



Step 6 : IoT Level Specification

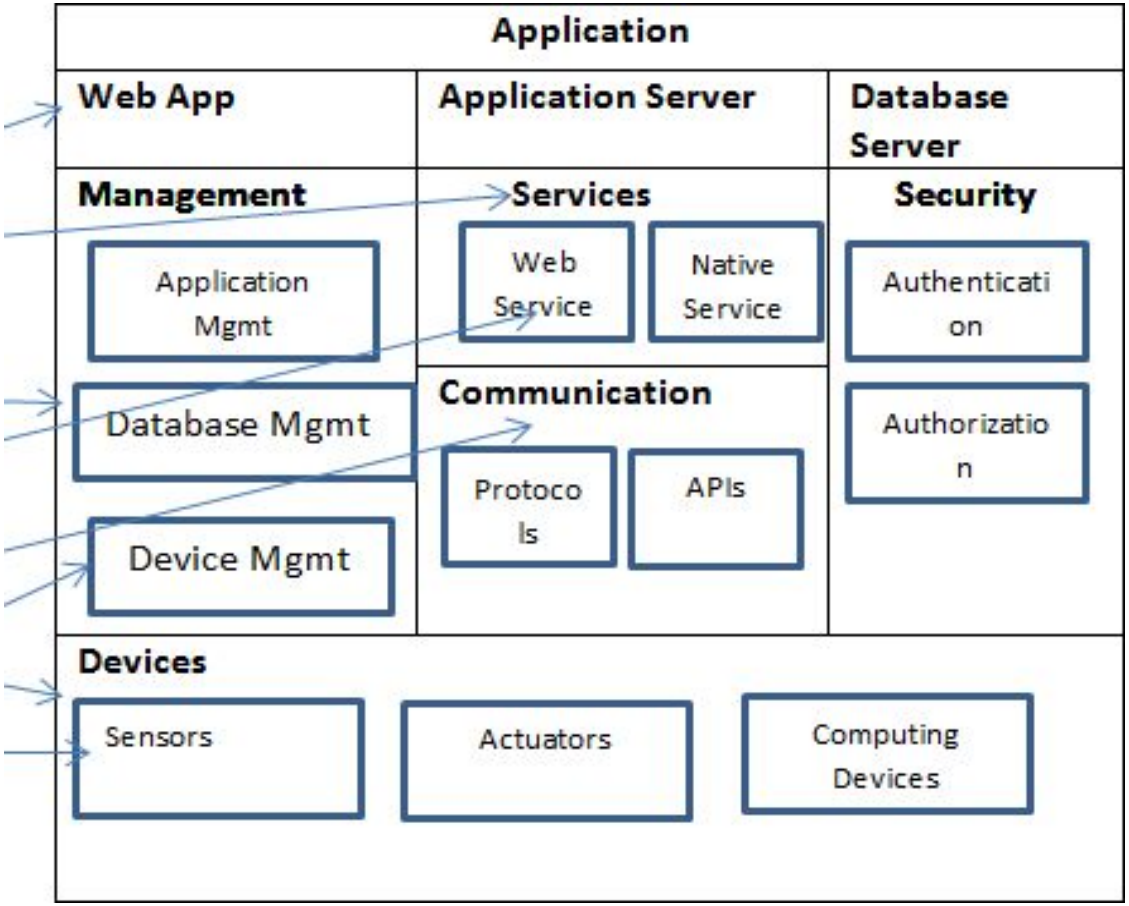


- Decide the deployment level of IoT System.
  Here I am using Deployment Level 1.

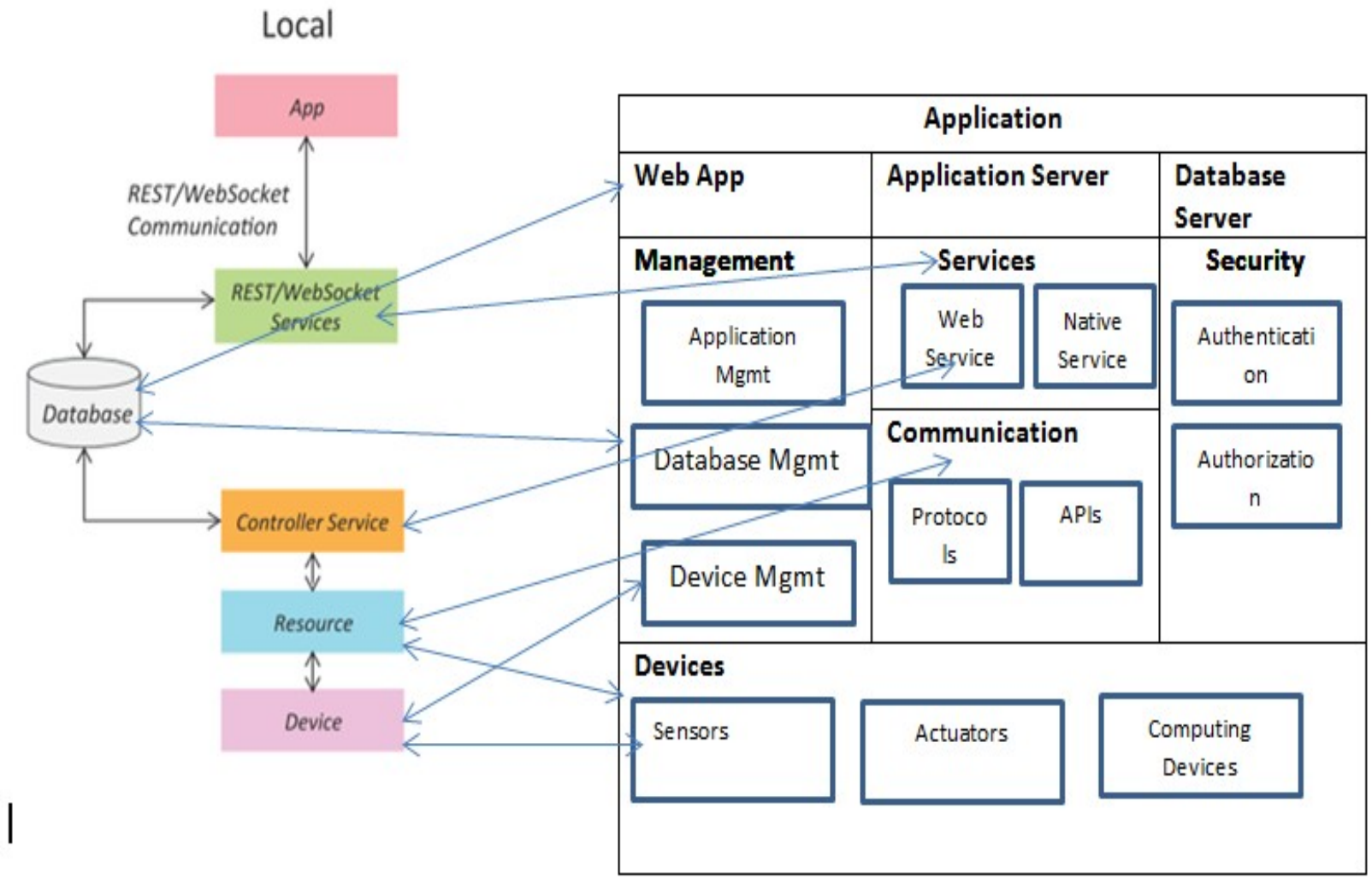
Step 7 : Functional View Specification

- Define the functions of IoT System grouped into various functional groups.
- These functional groups provide functionalities for interacting with the concepts defined in Domain model specification.

Step 7 : Functional View Specification…Drawing Functional Groups



Step 7 : Functional View Specification
…Deployment level to Functional Group Mapping



Step 8 : Operational View Specification

- Define the Operations/options related to IoT System development
- Such as Device options, Storage options, Application hosting option

Step 8 : Operational View Specification of automated irrigation system

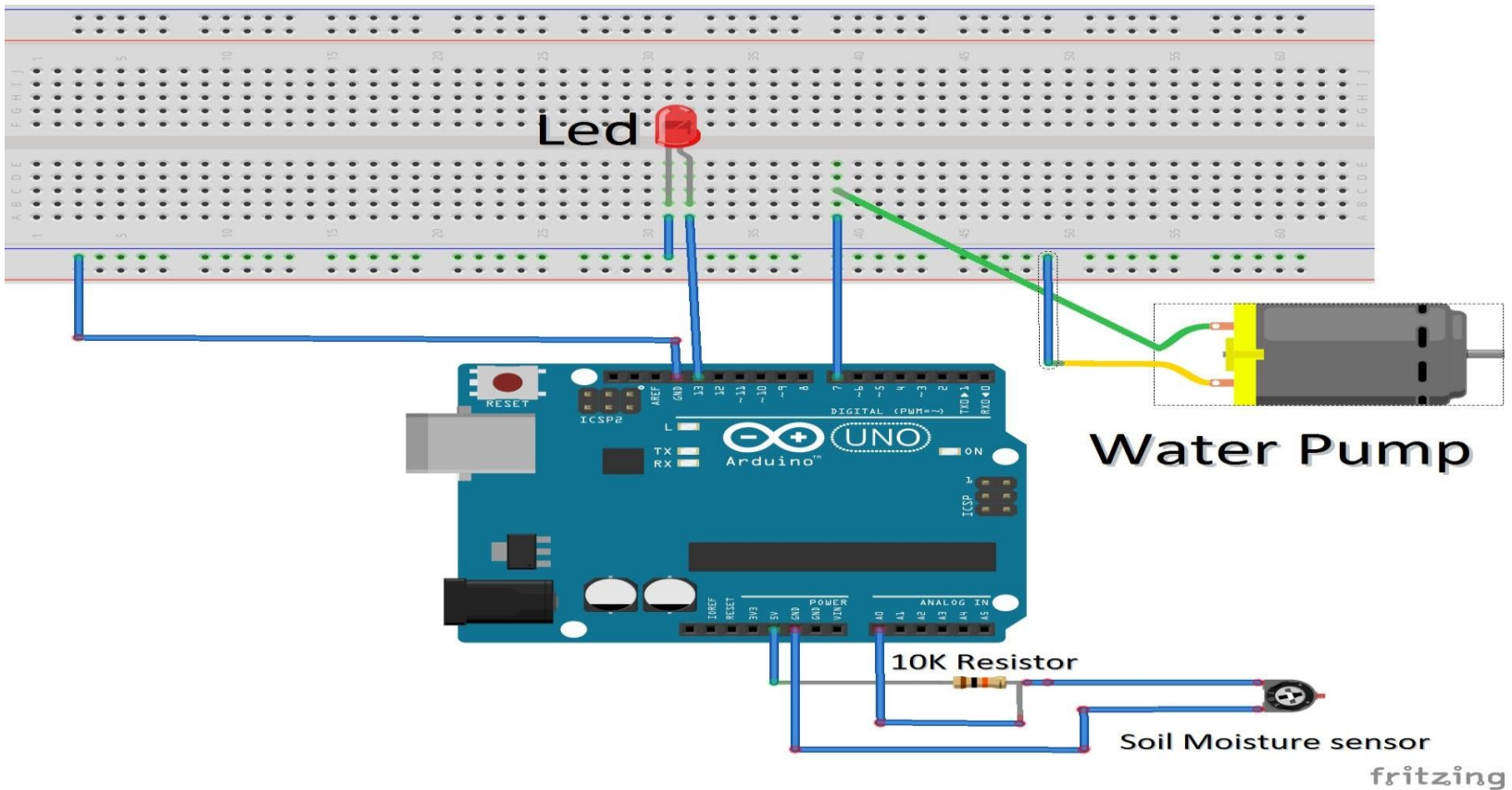
- Application
  - Web App : PhP WebApp
  - Application Server : Google App engine
  - Database Server : MySQL
- Services
  - Native : Controller Service
  - Web : REST
- Communication
  - Communication APIs : REST APIs
  - Communication Protocol :
    - Link Layer: 802.11
    - N/w : IPV6
    - Transport : TCP
    - Application : HTTP

Step 8 : Operational View Specification of automated irrigation system

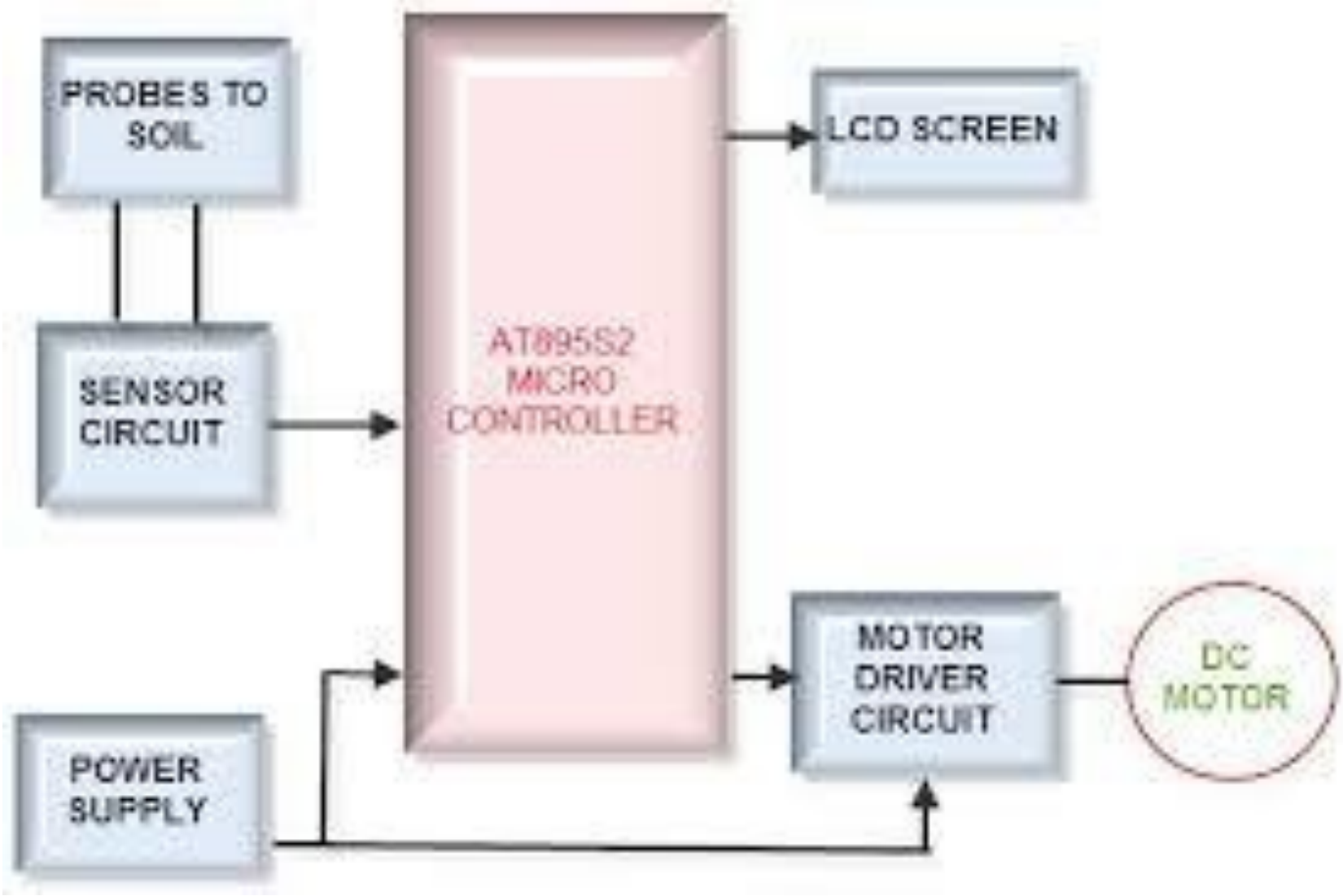
- Management
  - Device Management: Arduino device management
  - Application Management : PHP App Management
  - Database Management: MySQL Db Mgmt
- Security
  - Login Management

Step 9 : Device and Component Integration

- Integrates thedevices and components and draw a schematic diagram showing the same

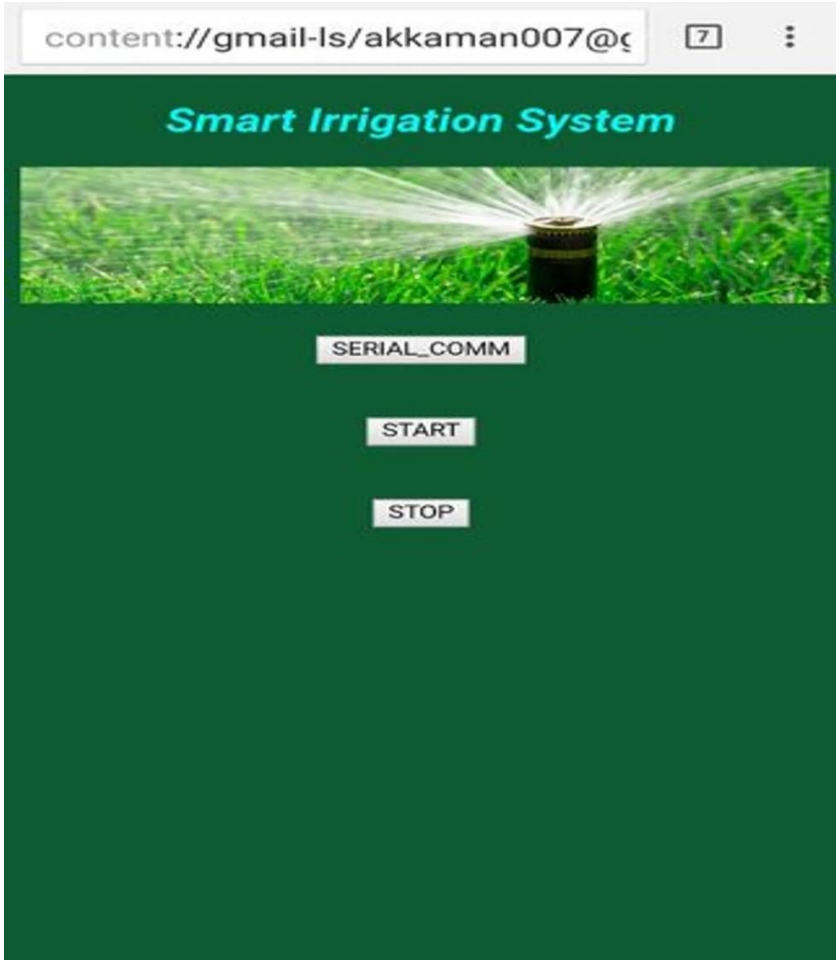


Device and Component Integration : Alternate Diagram



Step 10 : Application development

- GUI / Screenshot of IoT Application



IoT Health Monitoring Application(HMA)
Purpose and Requirement Specification

- **Purpose of the Project:** The Health System Monitoring is basically used to monitor the Patient Body Temperature and Pulse by respective sensors. This information is captured and stored, so that the authorized personnel can view and analyze the Data remotely at any time.

IoT Health Monitoring Application(HMA)

Purpose and Requirement Specification

- **Behavior:** The Monitoring is done in real time to identify the state of the patient. This information can be used to analyze the state of a patient or to get sensitive data in order to be sequentially captured for medical diagnosis. The system Alarms in Emergency Situation and notifies the staff and relatives of the patient.

IoT Health Monitoring Application(HMA)

Purpose and Requirement Specification

- **System Management Requirement:** The system provides remote monitoring and control functions
- **Data Analysis Requirement:** System allows the analysis of data and can be visualized in graphical format.
- **Application Deployment Requirement:** Application will be deployed locally on the device, but can be monitored remotely.
- **System Requirements:** Only Authorized users can access and control the Application

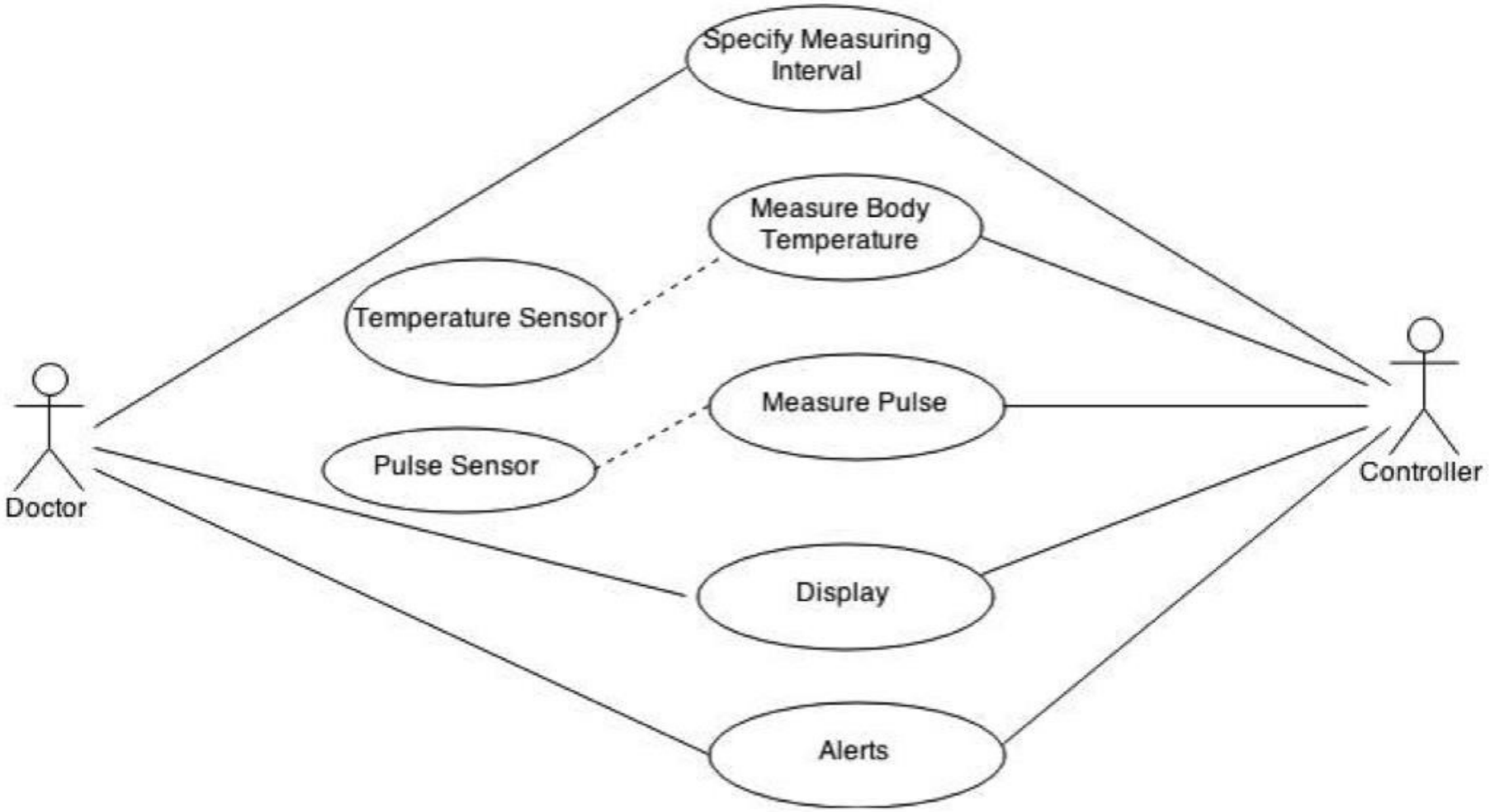
IoT Health Monitoring Application(HMA)
Process Specification

- The Use Case Diagram describes the use case's of the system and the actors involved.
- The Process diagram shows the steps involved in the process.

The sensors read the information from the Human Body and stores it in Database, when the values go beyond the threshold limit it sends alerts.

IoT Health Monitoring Application(HMA)

Process Specification



IoT Health Monitoring Application(HMA)
Domain Specification

- **Physical Entity :** In HMA, human body is the physical entity where body temperature and pulse are monitored using respective sensors.

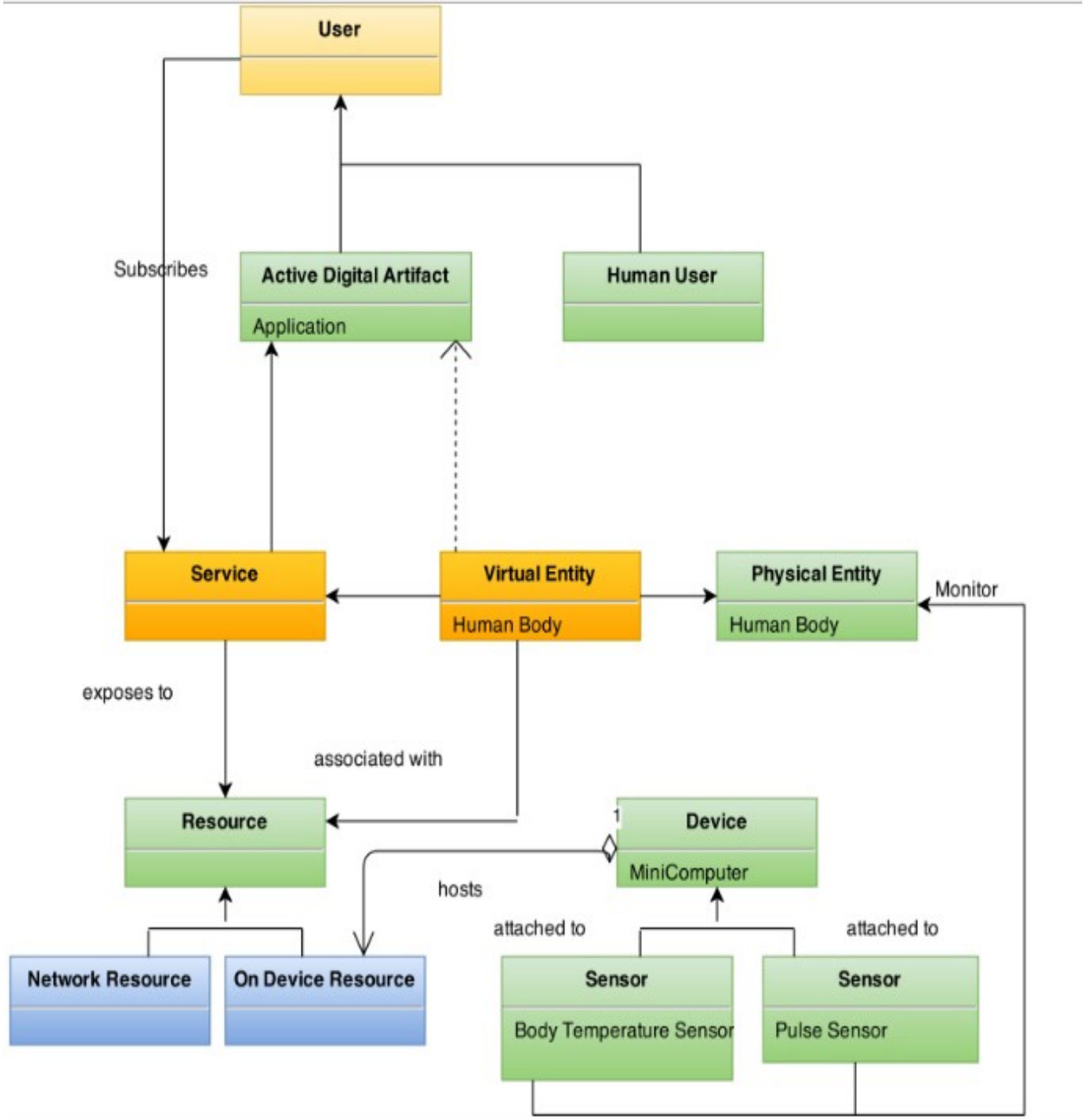
- **Virtual Entity :** It's a representation of Physical Entity in Digital World.For each physical entity there exist one virtual entity in Domain Model.

- **Device:** Provides medium for interactions between Physical and Virtual Entities.In HMA, device is a single board (Arduino) which has temperature and pulse sensor attached to it.

IoT Health Monitoring Application(HMA)
Domain Specification

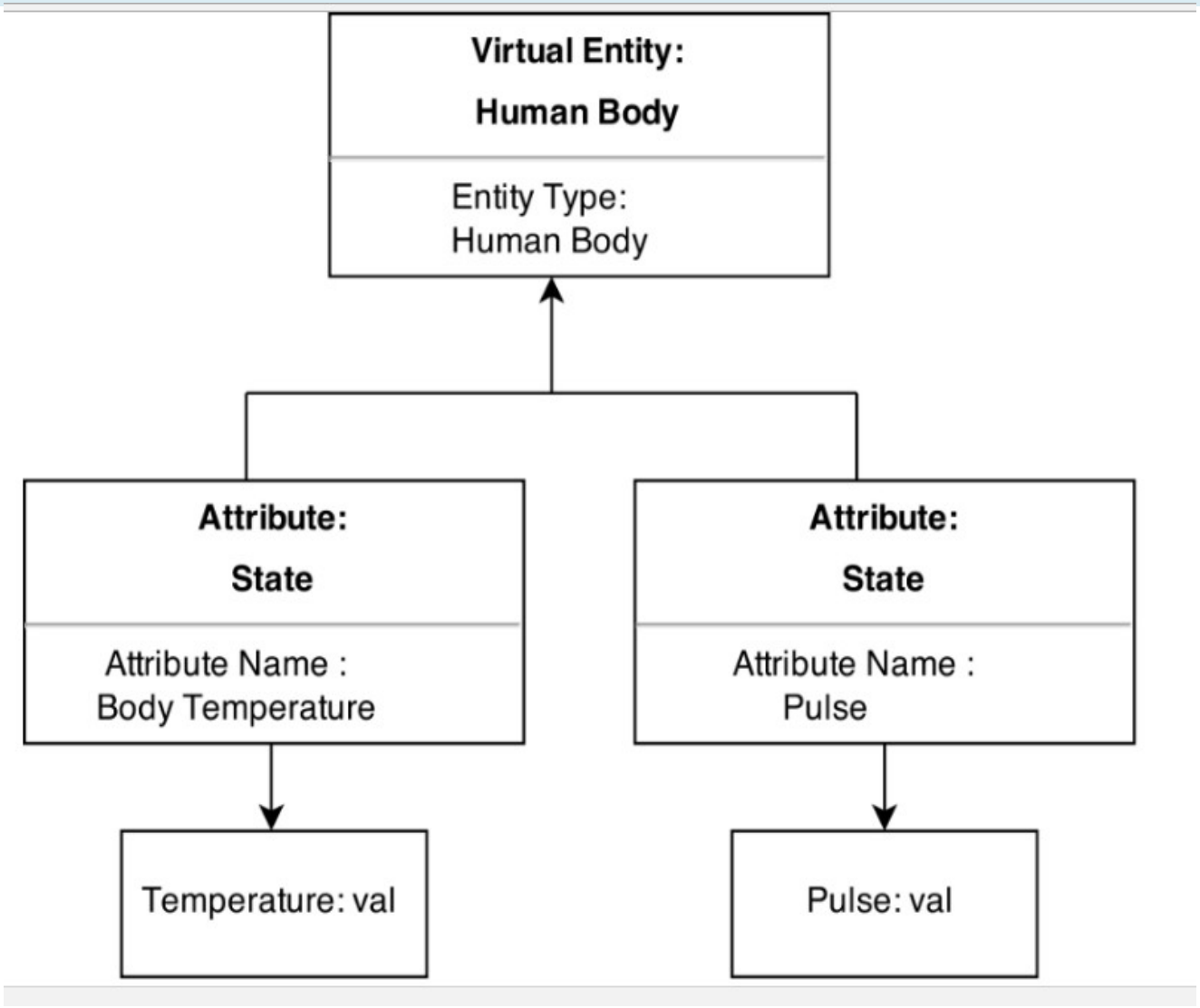
- **Resource :** Can be Software Components which can be either "on-device" or "network- resources". On-Device resources are hosted on device and include software components that provide information about physical entity. Network Resources are software components on network such as Database.

- **Service :** In HMA, the services will be, service that retrieve current information, native service that runs on the device.


User
Subscribes
Active Digital Artifact
Application
Human User
Service
Virtual Entity
Human Body
Physical Entity
Human Body
Monitor
exposes to
associated with
Resource
1
Device
MiniComputer
hosts
attached to
attached to
Network Resource
On Device Resource
Sensor
Body Temperature Sensor
Sensor
Pulse Sensor

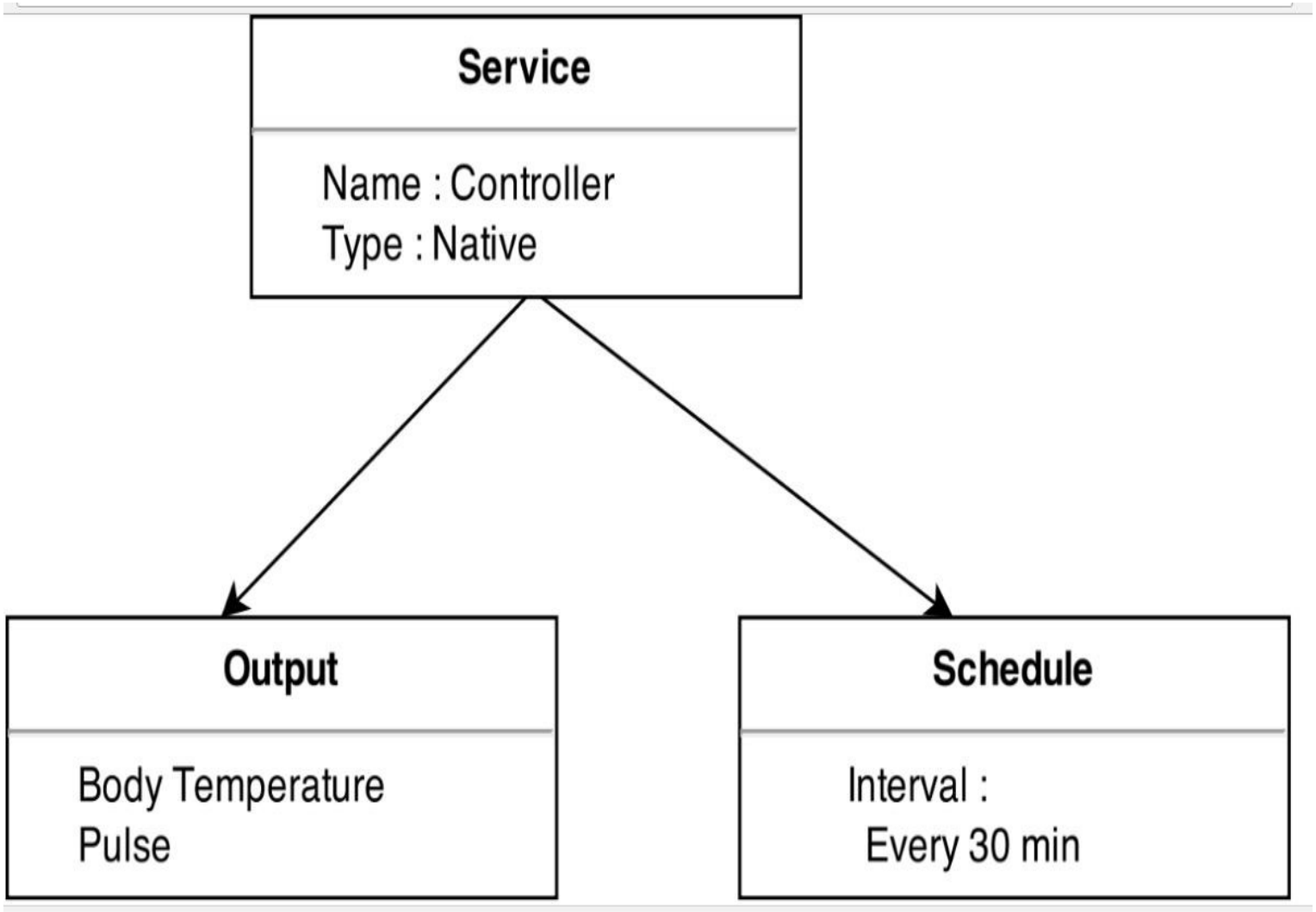
IoT Health Monitoring Application(HMA)

Information Model Specification



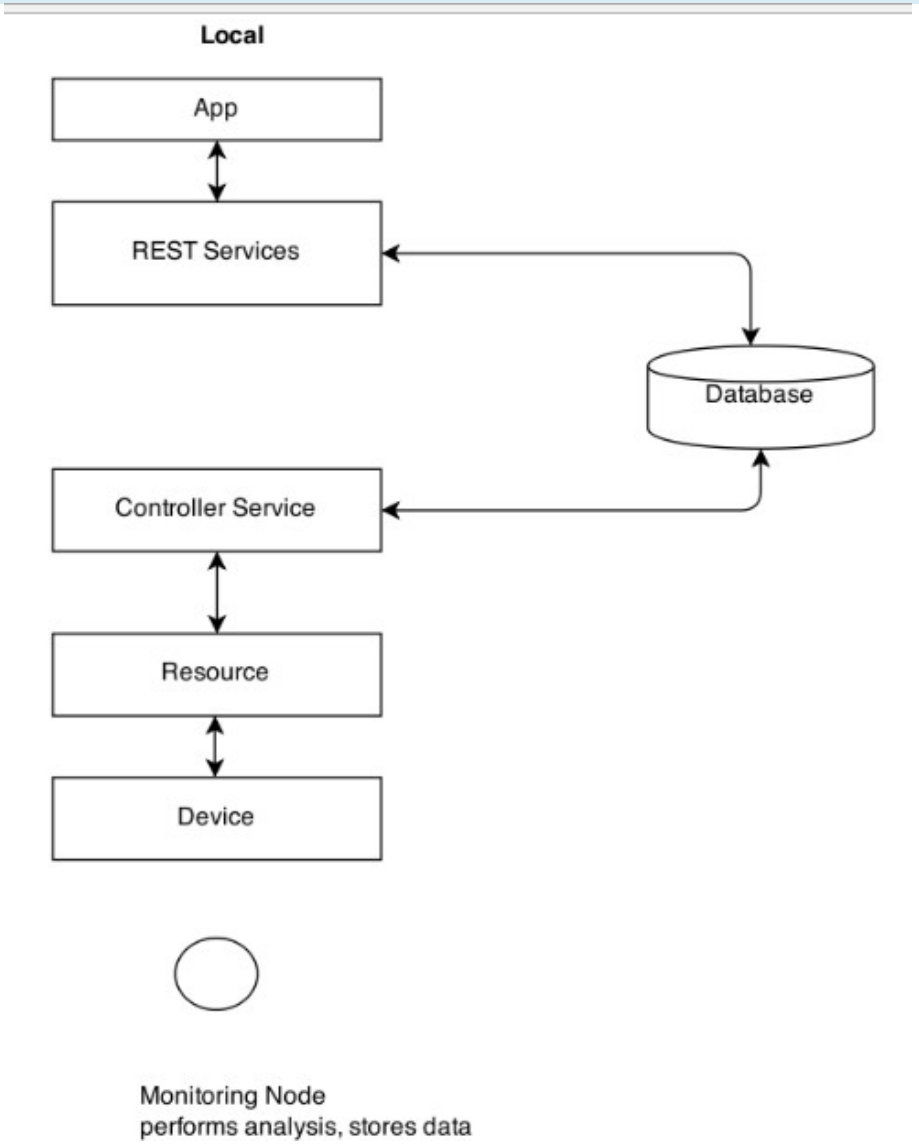
IoT Health Monitoring Application(HMA)

Service Specification

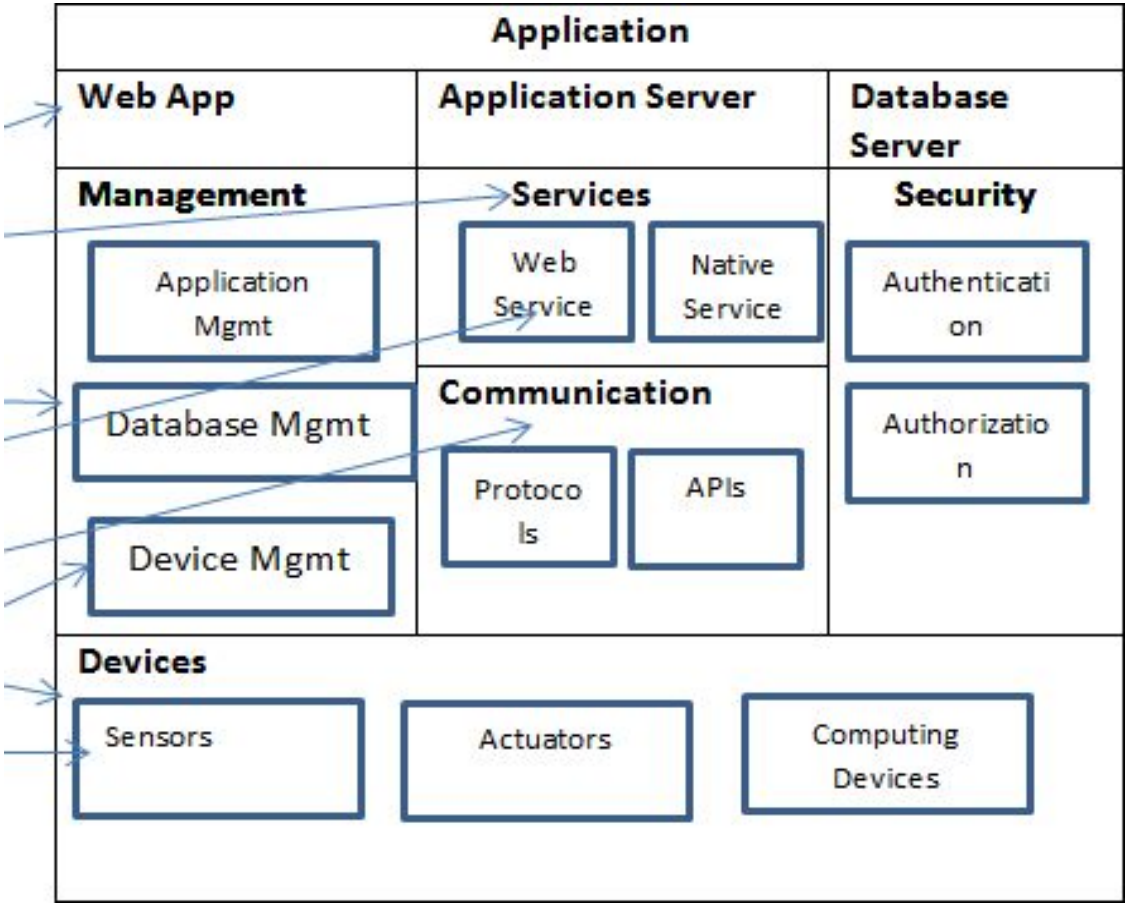


IoT Health Monitoring Application(HMA)
IoT Level Specification

The single system has device that performs sensing, stores data perform analysis and host the Application. Thus, IoT Level 1

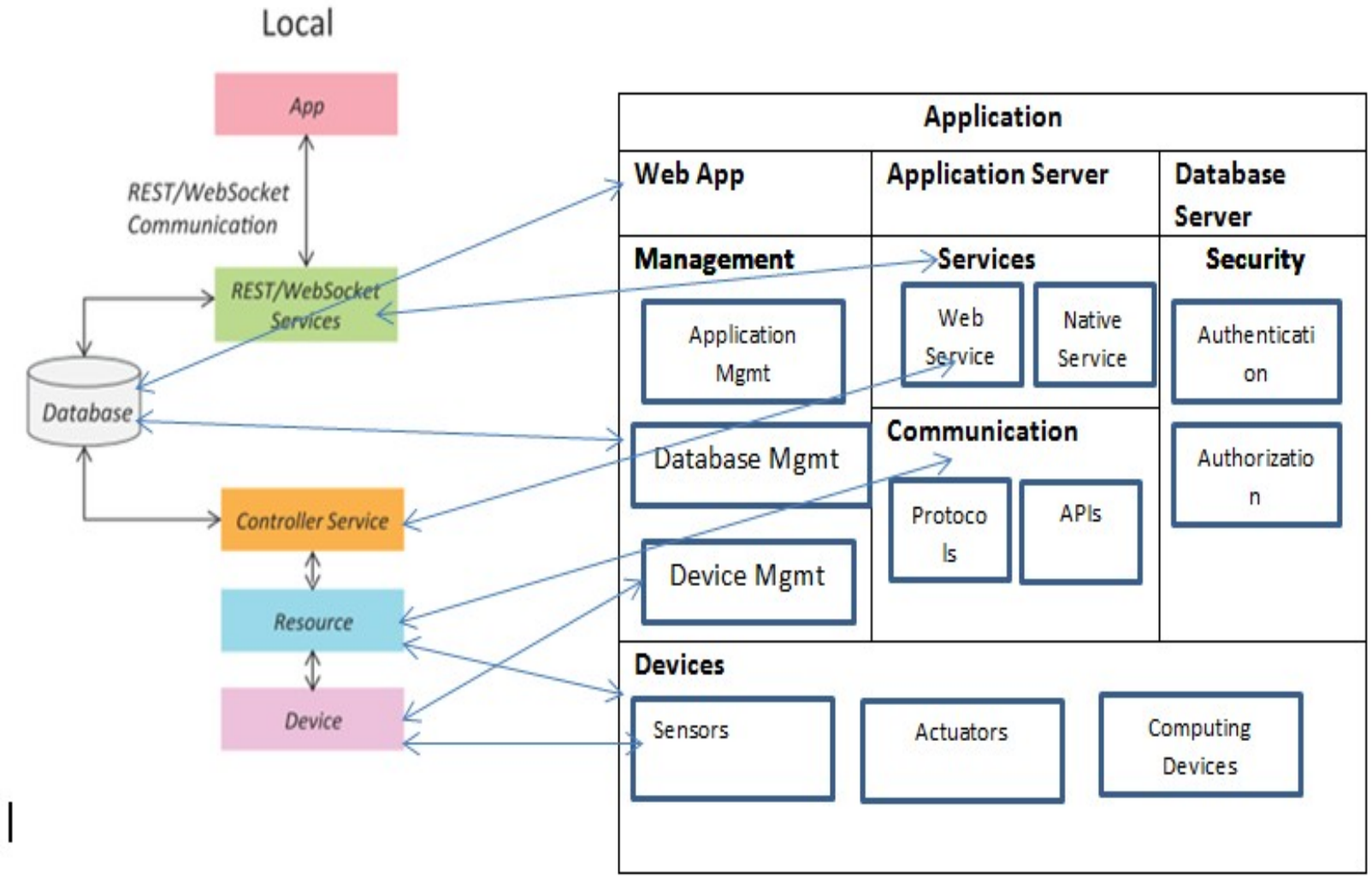


IoT Health Monitoring Application(HMA)
Functional Specification



IoT Health Monitoring Application(HMA)

Functional Specification



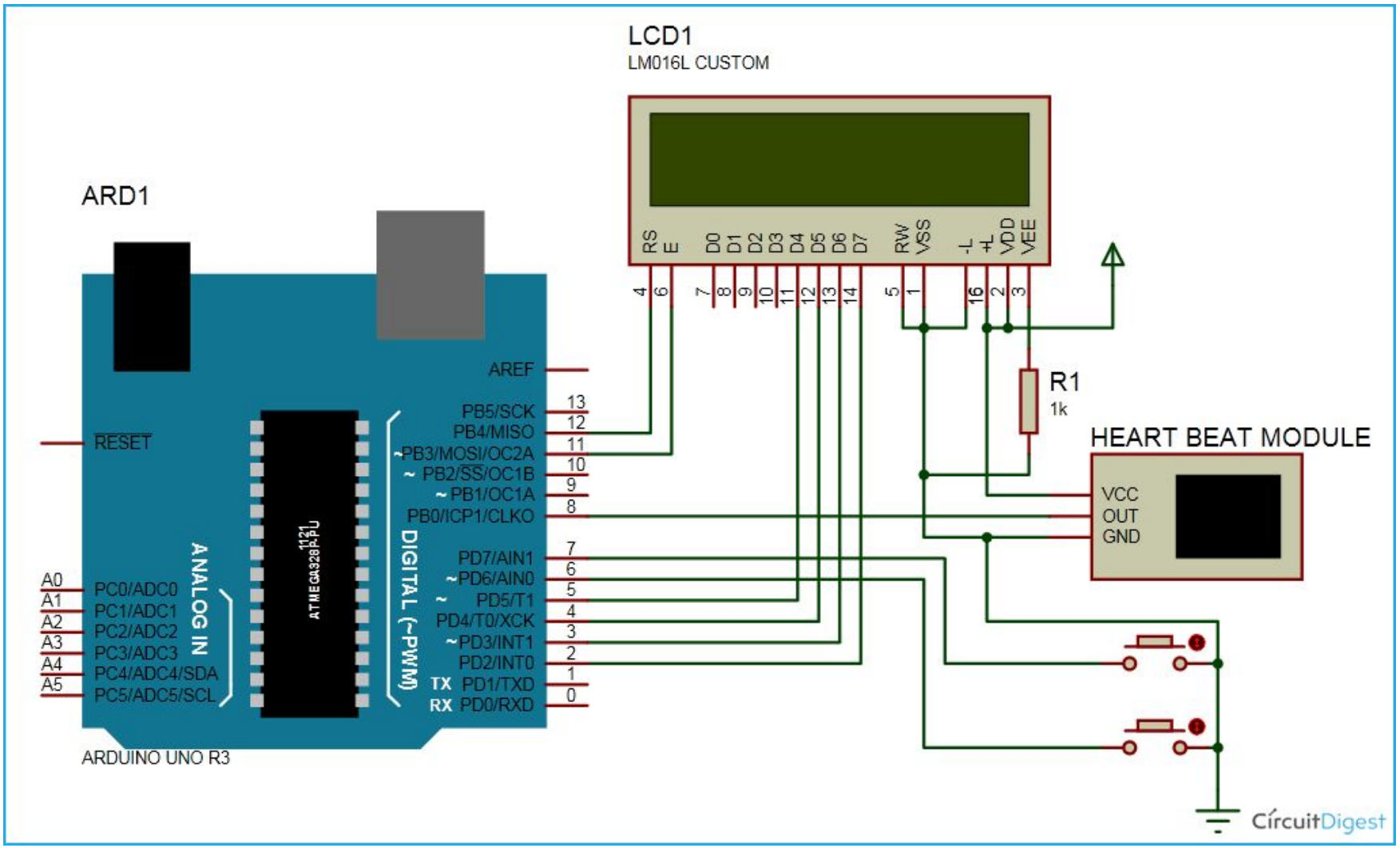
Step 8 : Operational View Specification of HMA

- Application
  - **Web App** : PhP WebApp
  - **Application Server** : Google App engine
  - **Database Server** : MySQL
- Services
  - **Native** : Controller Service
  - **Web** : REST
- Communication
  - **Communication APIs** : REST APIs
  - **Communication Protocol** :
    - Link Layer: 802.11
    - N/w : IPV6
    - Transport : TCP
    - Application : HTTP

Step 8 : Operational View Specification of HMA

- Management
  - **Device Management:** Arduino device management
  - **Application Management** : PHP App Management
  - **Database Management**: MySQL Db Mgmt
- Security
  - **Login Management**

Step 9 : Device and Component Integration



Step 10 : Application Specification

Home Intrusion Detection System

HID
Purpose and Requirement Specification

- **Purpose of the Project:**

The purpose of home intrusion detection system is to detect intrusions using sensors and raise alerts, if necessary.

HID

Purpose and Requirement Specification

- **Behavior:**

In case of any intrusion, I intend to capture a picture of the intruder, mail the image to the respective end users and alert them. I would like to use an alarm which goes on in case of an intrusion.

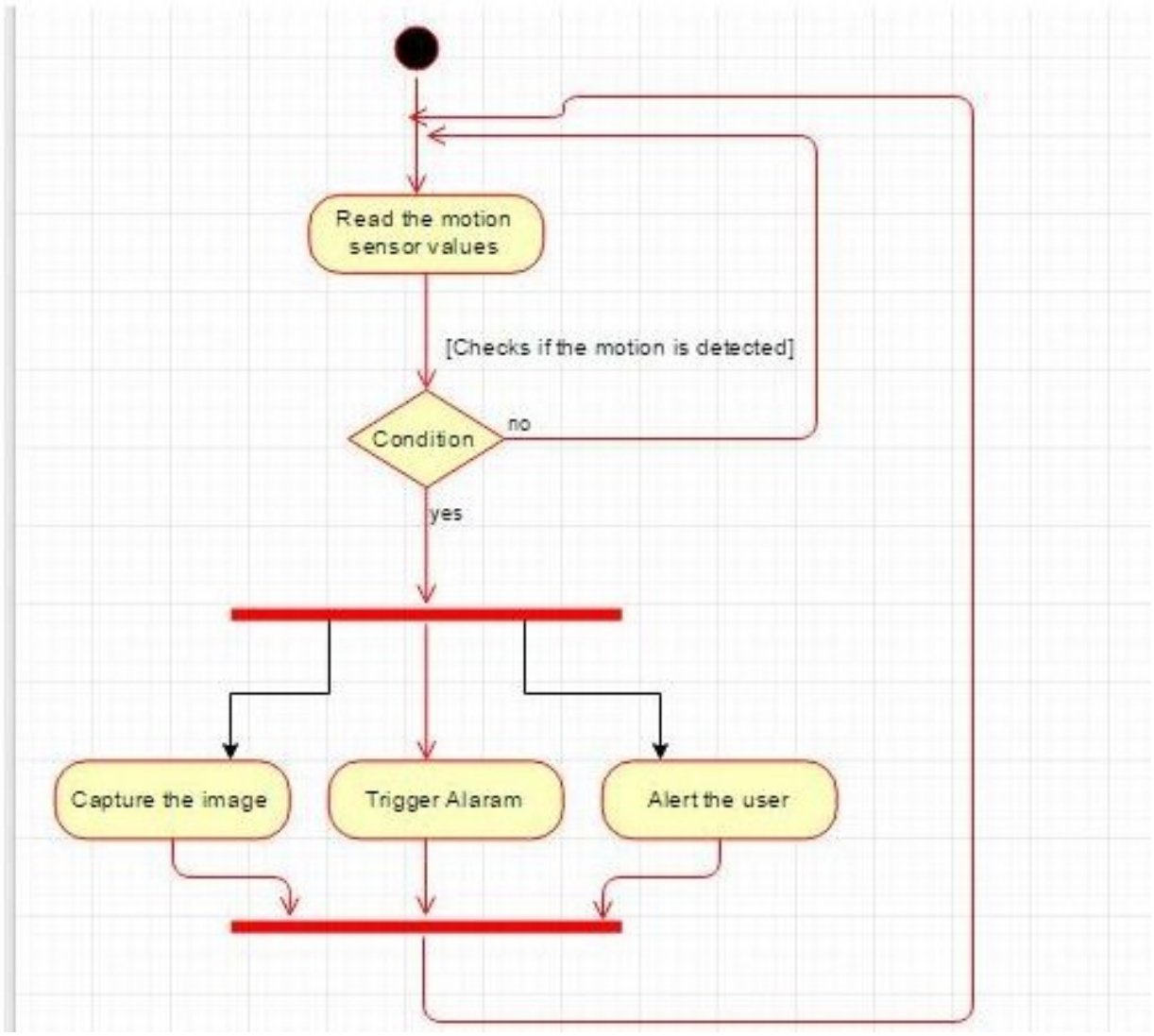
HID
Purpose and Requirement Specification

- **Data Analysis Requirement:** The light dependent resistor data is analyzed locally. If the sensor reading falls below the threshold value, an image is captured and send as an alert to the user.
- **Application Deployment Requirement:** The application is deployed locally on the device and can be accessible from anywhere via node-red.
- **System Requirements:** Only Authorized users can access and control the Application

HID **Process Specification**

- **The Use Case Diagram describes the use case's of the system and the actors involved.**
- **The Process diagram shows the steps involved in the process.**
- With the help of Light dependent resistor and PIR motion sensor, I am going to detect the motions in the room.
- If a motion is detected, I intend to capture the image with the help of a webCam and store locally
- Now the alerts are sent to the user with the captured image.
- Also I am using the buzzer which turns on in case of any intrusion.

HID **Process Specification**



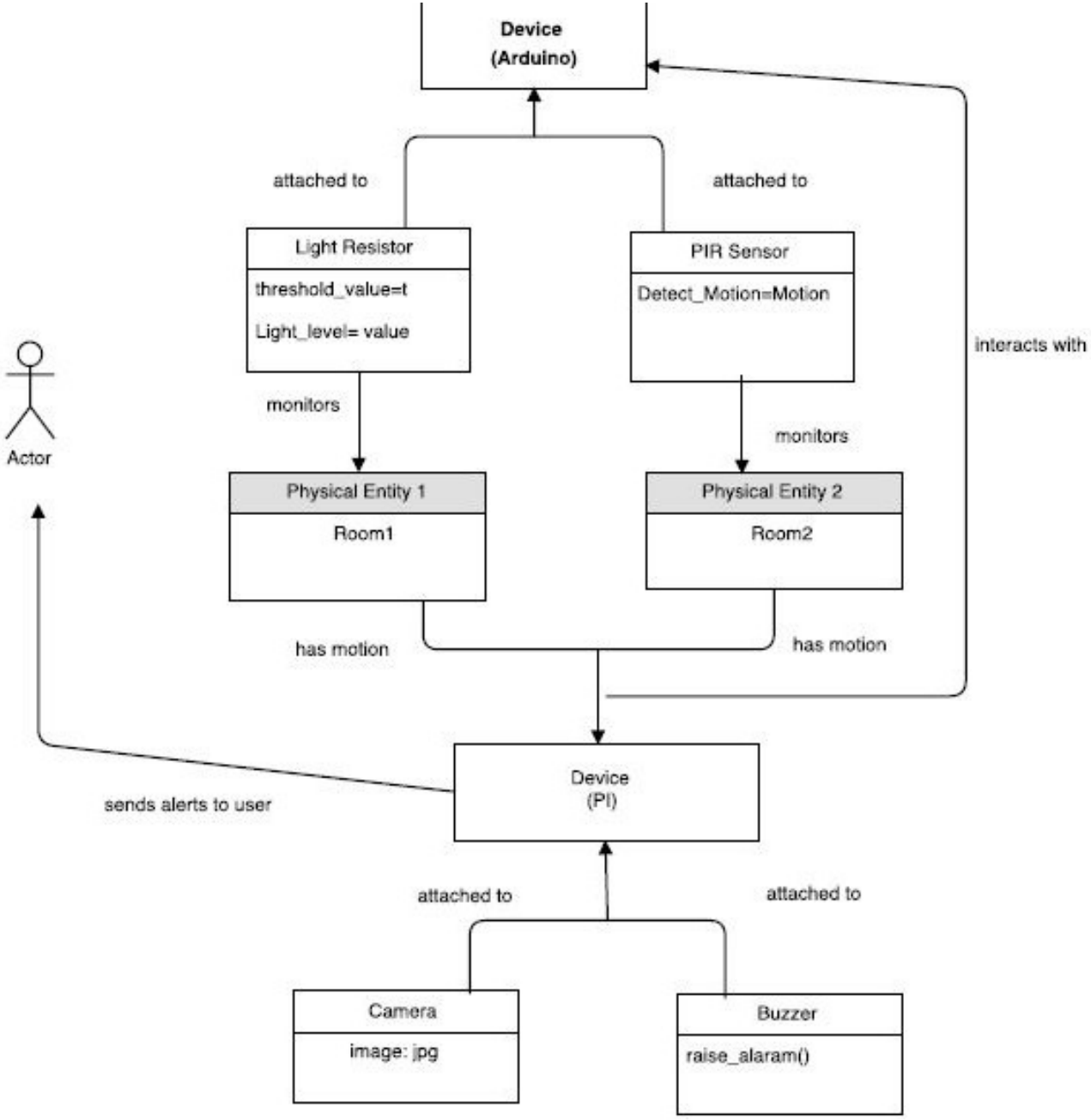
HID
Process Specification

- From the above process diagram, we can see that the process starts at the circle. The PIR motion sensor which is shown in the rectangular box in the above figure, is placed in a room.It detects the motion in the room. If there is a motion then an image is captured and an alert will be sent to the user. And also it turns on the buzzer. This decision (yes/no) to raise an alert is shown in the diamond box in the above figure.

HID
Domain Specification

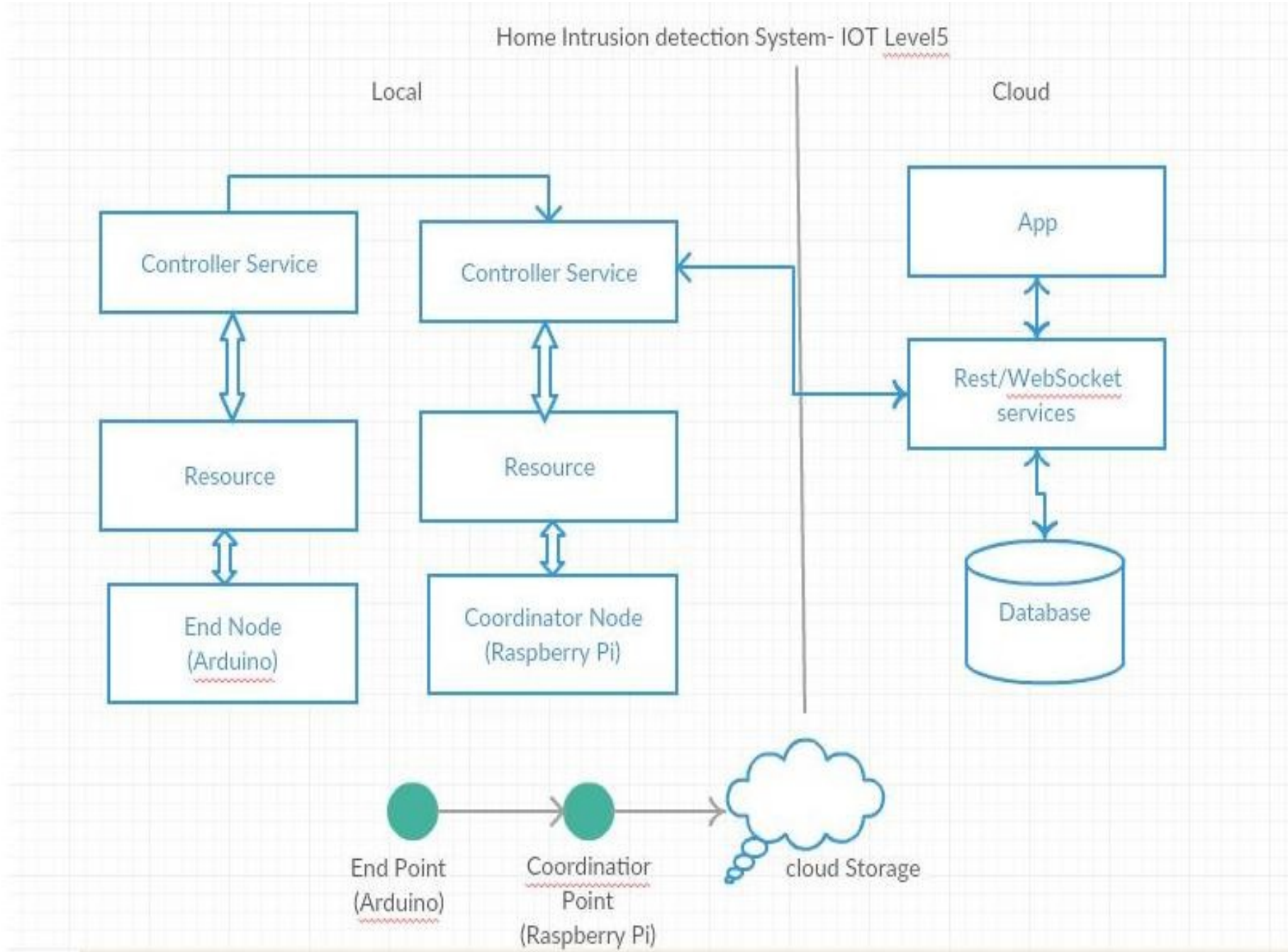
• The domain model	describes	the	main
concepts,entities and	objects	in	the

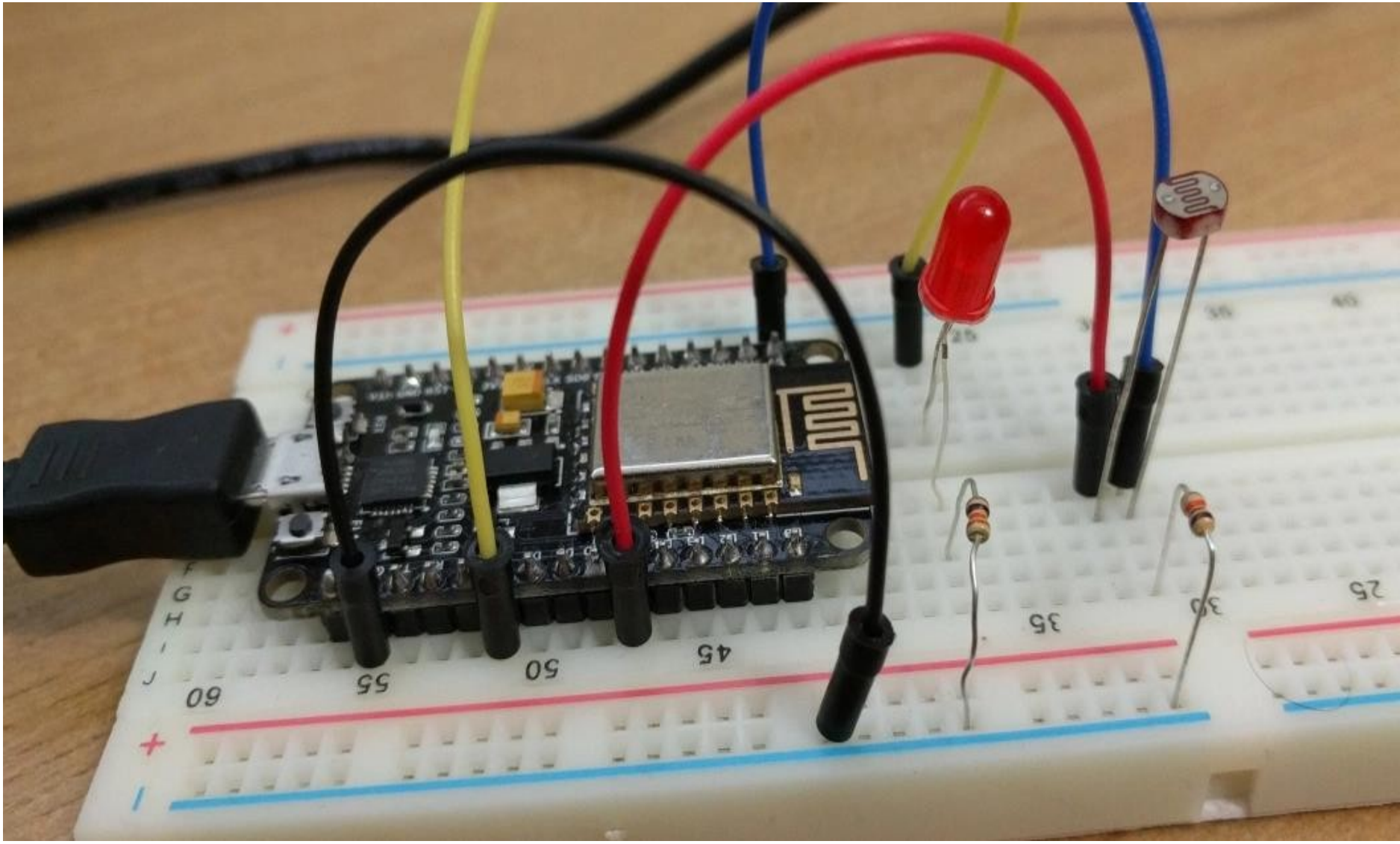
domain.The domain includes physical entities **for rooms.** The **devices in my system are nodemcu to which PIR and light dependent resistor is attached** and **raspberry pi to which camera and buzzer are attached to send alerts.**


Device
(Arduino)
attached to
attached to
Light Resistor
threshold_value=t
Light_level= value
PIR Sensor
Detect_Motion=Motion
interacts with
Actor
monitors
monitors
Physical Entity 1
Room1
Physical Entity 2
Room2
has motion
has motion
Device
(PI)
sends alerts to user
attached to
attached to
Camera
image: jpg
Buzzer
raise_alaram()

IoT level Specification

- implement my IOT system that is "**Home intrusion detection system**" in ***level 5***. The IOT level 5 system has multiple nodes and one coordinator node. Coordinator node collects the data from the end nodes and sends to the cloud.


Home Intrusion detection System- IOT Level5
Local
Cloud
Controller Service
Controller Service
App
Rest/WebSocket services
Resource
Resource
Database
End Node (Arduino)
Coordinator Node (Raspberry Pi)
End Point (Arduino)
Coordinatior Point (Raspberry Pi)
cloud Storage



May 2019

- Q.3 a) With the help of diagram list and briefly explain the steps involved in the IoTsystem design methodology [5M]
- Q.3 b) Explain the application development step of IoT design methodology, consider smart IoT-based home automation system as an example. 5M
- Q. 4 a) Explain the purpose and requirement specification in IoT design methodology, consider smart IoT-based home automation system as an example.[5M]
- Q.4 b) Explain the service specification step of IoT design methodology, consider smart IoT-based home automation system as an example. 5M

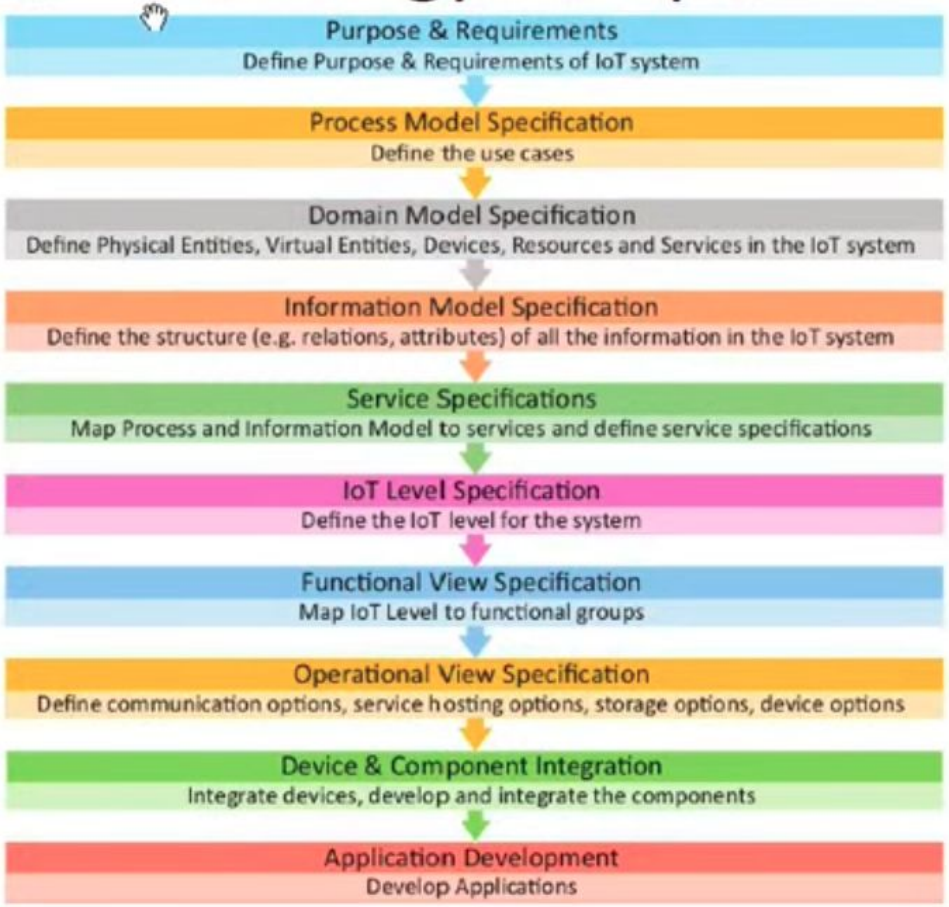
Mar 2018

- Q3) a) Explain purpose and requirements specifications step of IoT system design methodology, consider smart IoT - based home automation system as an example. [5]
- b) Explain information model specification step of IoT system design methodology, consider smart IoT - based home automation system as an example. [5]
- Q4) a) With the help of diagram list and briefly explain the steps involved in the IoT system design methodology. [5]
- b) Explain process model specification step of IoT system design methodology, consider smart IoT-based home automation system as an example. [5]

URL

- https://www.youtube.com/watch?v=5rJHy1WPaY0

IoT Design Methodology - Steps



Slide 63

Home Automation Case Study


SMART HOME
SH
+22°C
LOCK
LIVING ROOM
KITCHEN
BEDROOM
BATHROOM
NURSERY
NEXT

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.

Step:1 - Purpose & Requirements

- 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.

ESIOT2 IoT System Design Methodology, process speci domain model, functional view operational view

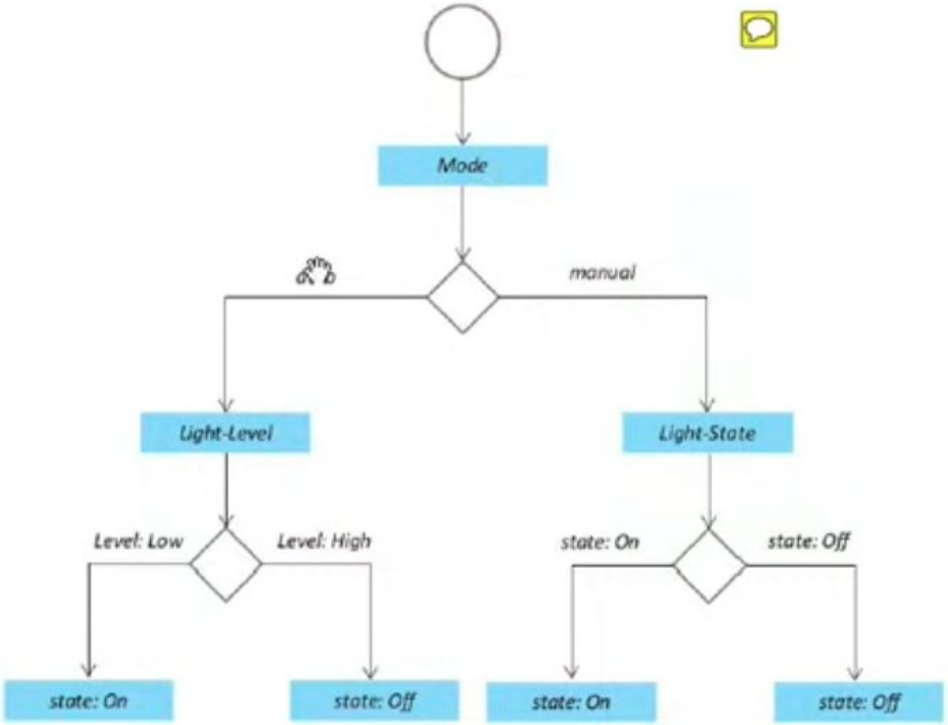
Step 2: Process Specification

- 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.

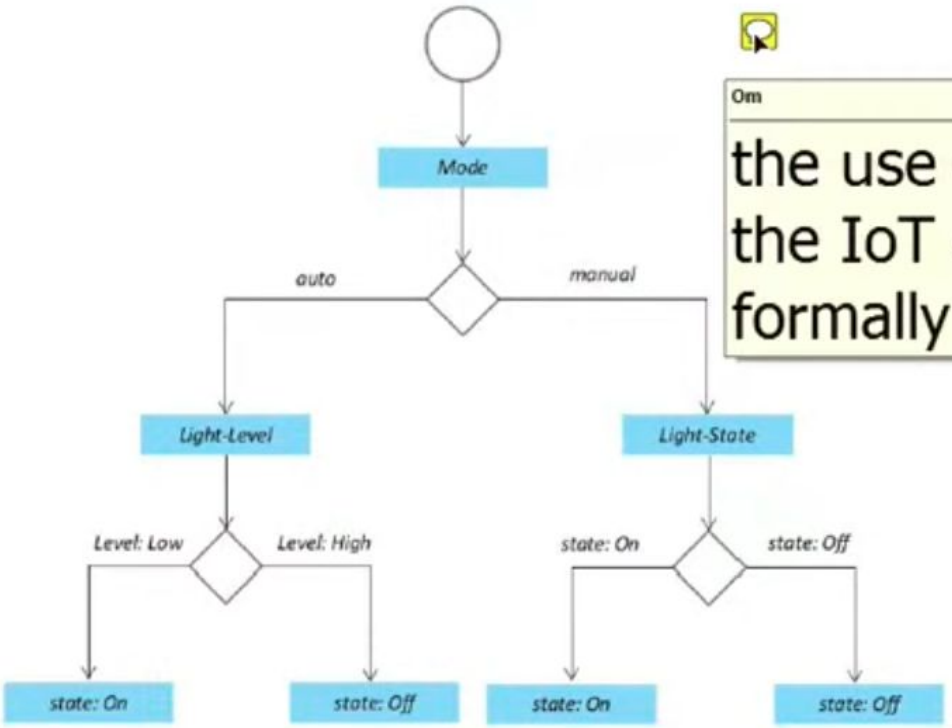
Slide 63

0:39 / 21:27

Step:2 - Process Specification



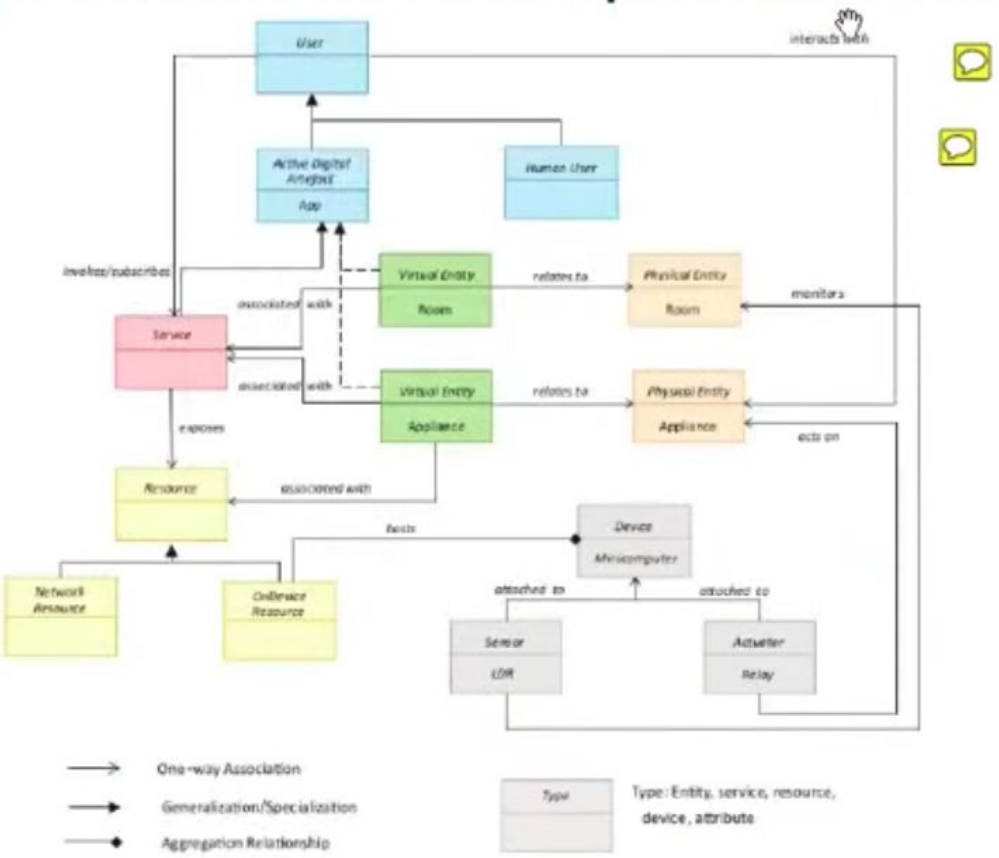
Step:2 - Process Specification



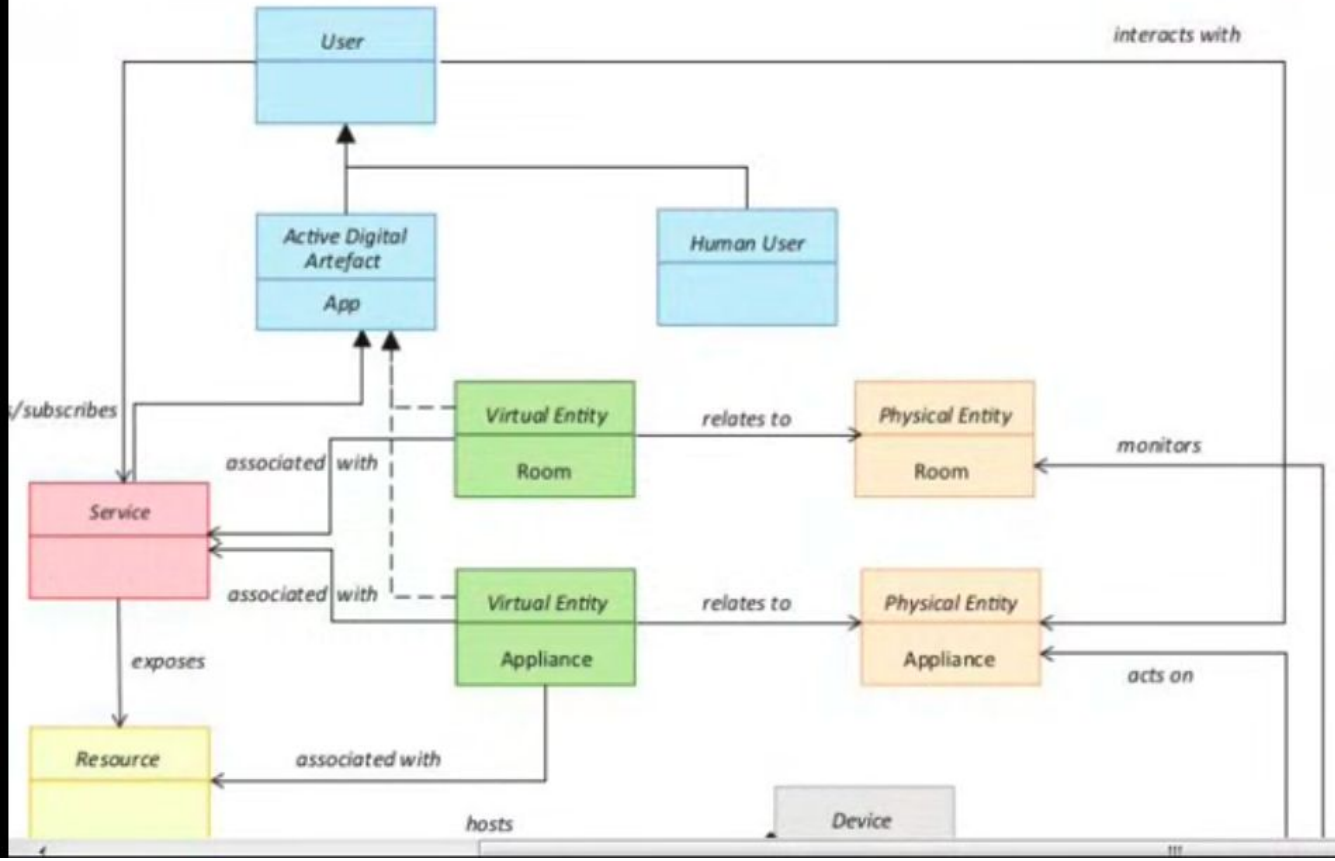
Step 3: Domain Model Specification

- 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 3: Domain Model Specification



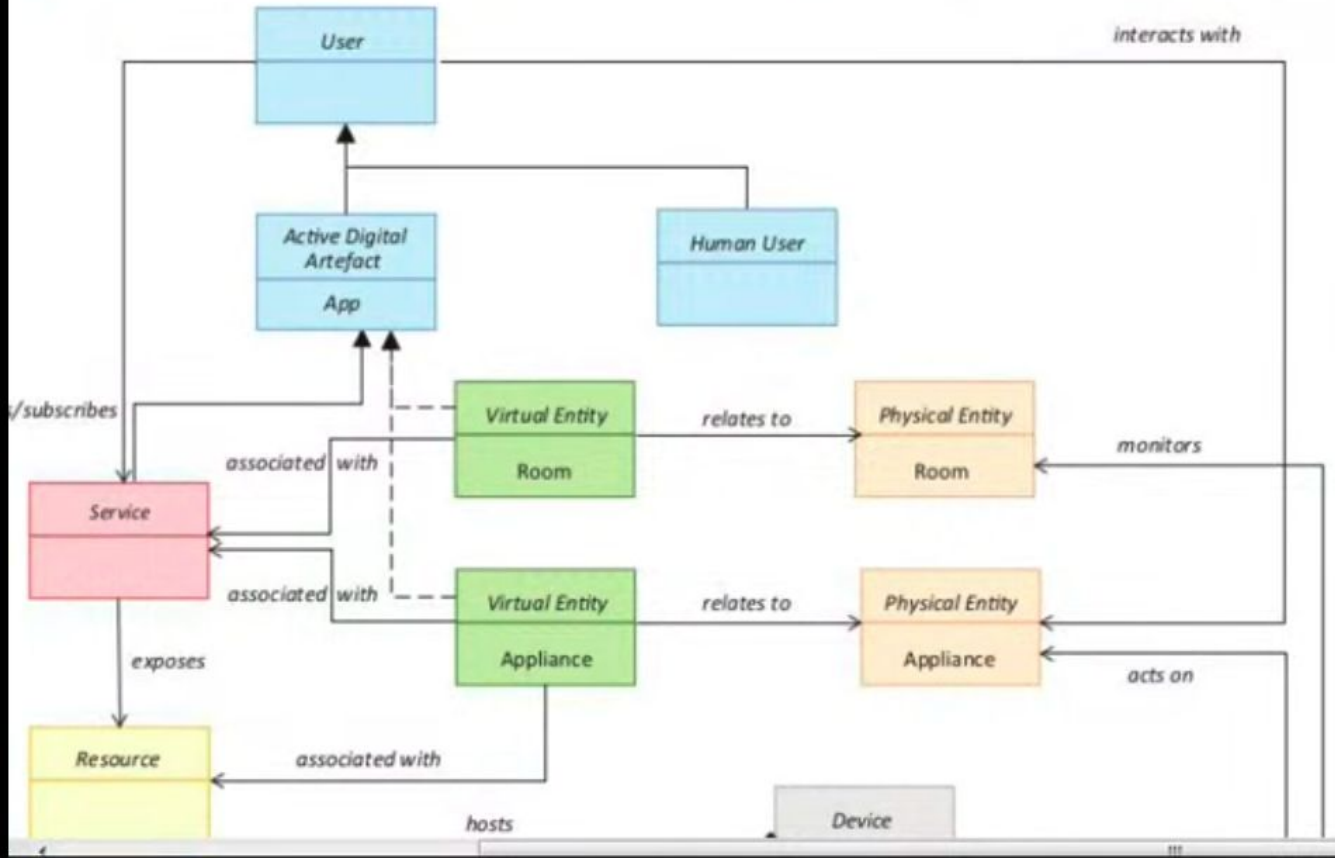
main Model Specification



Om

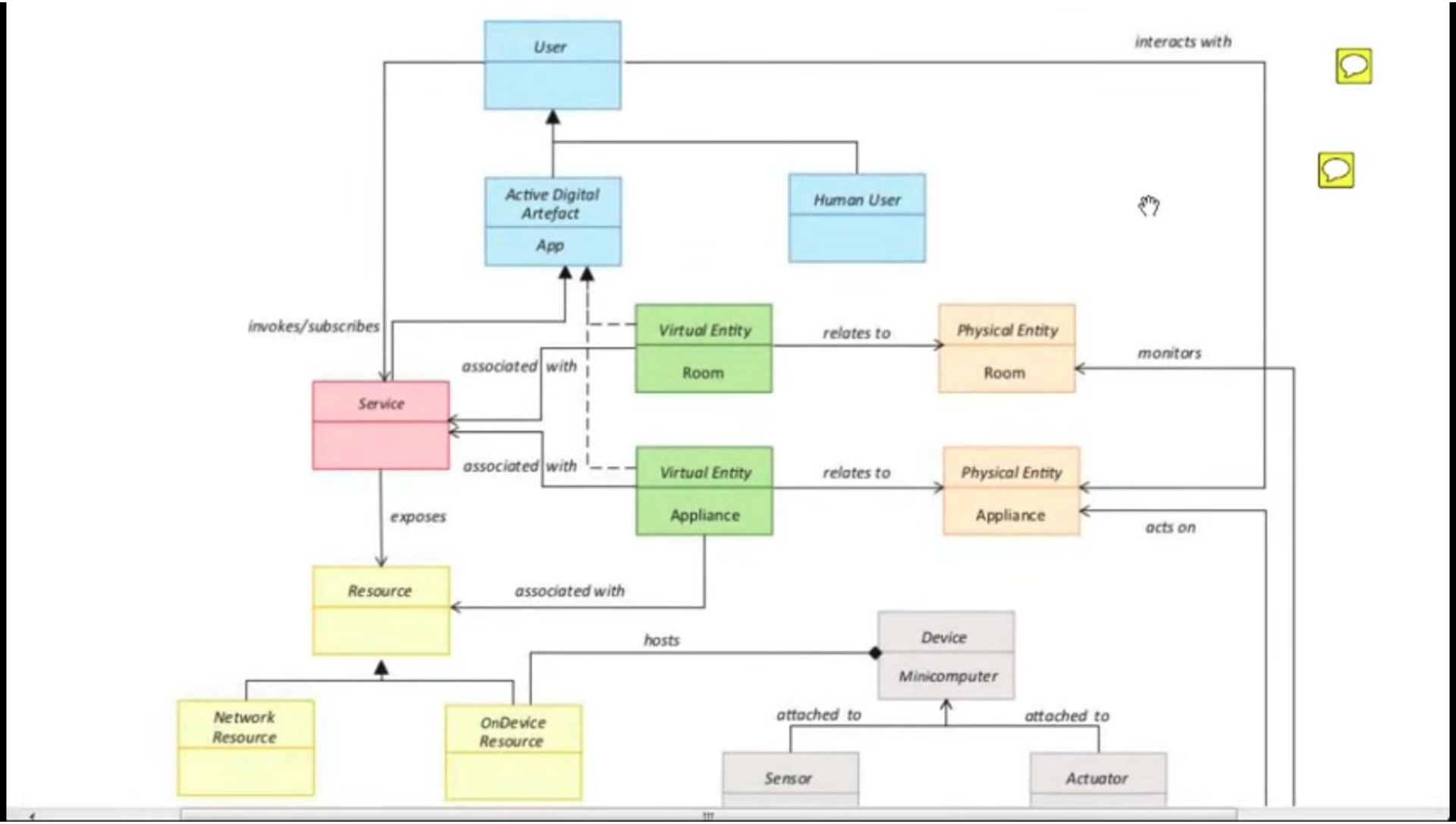
the main concepts, entities and objects in the domain of IoT system

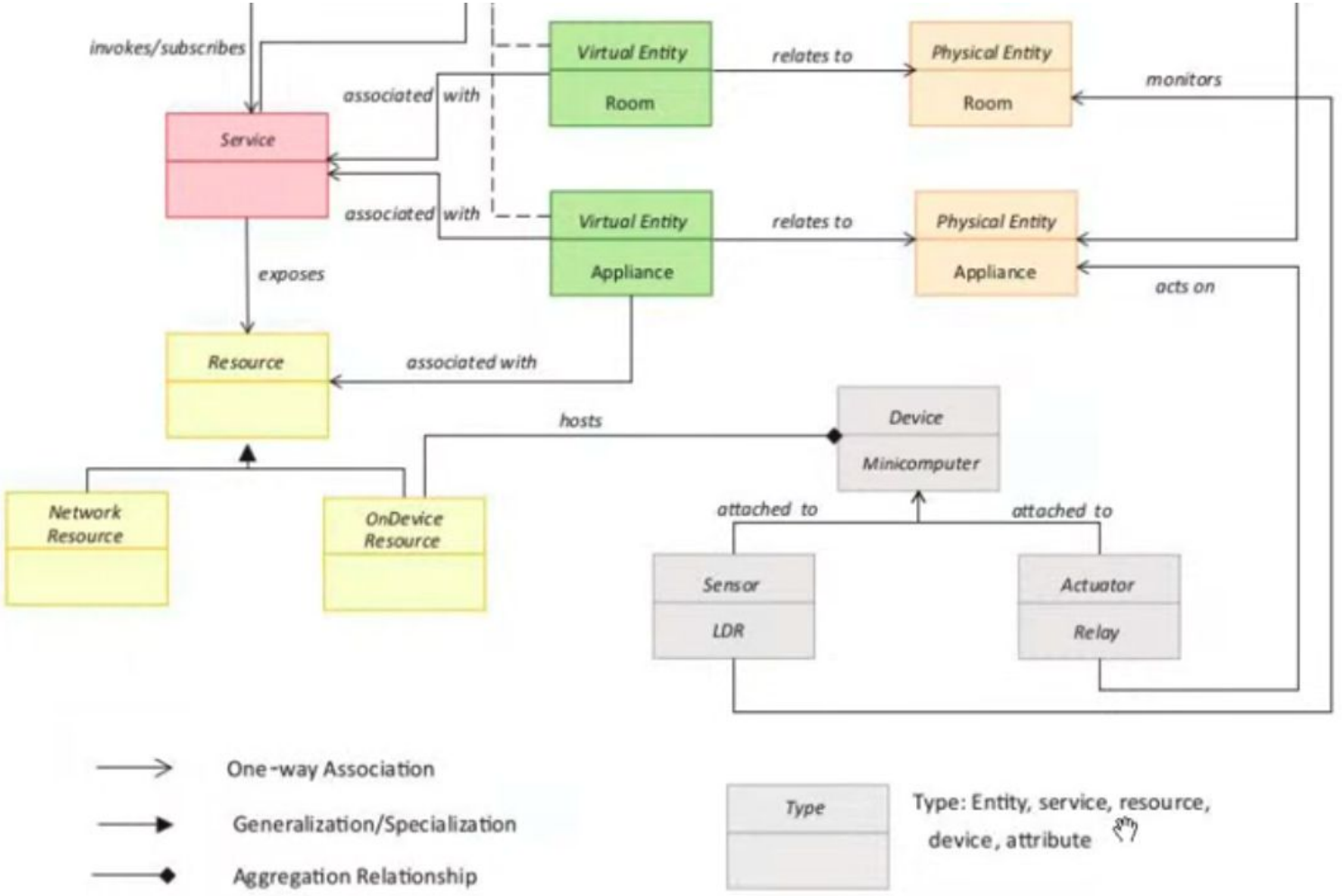
main Model Specification



Om

1. the objects and their relationships
2. provides an abstract representation

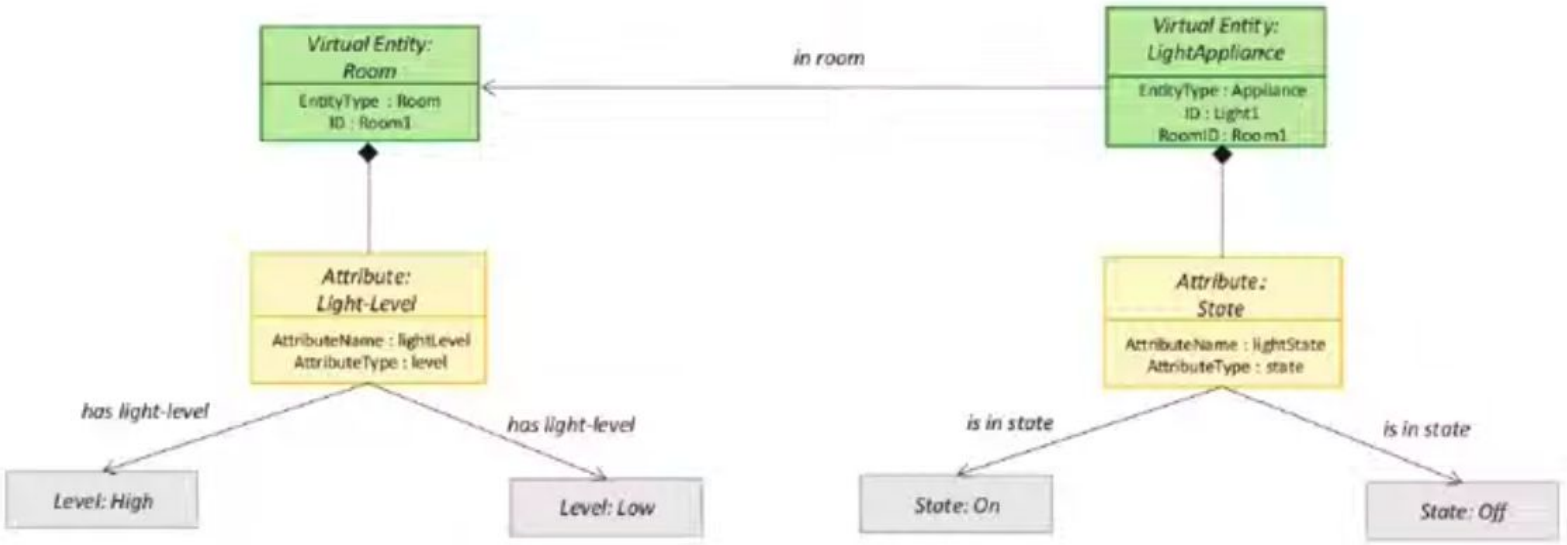

User
interacts with
Active Digital Artefact
App
Human User
invokes/subscribes
Virtual Entity
Room
relates to
Physical Entity
Room
monitors
associated with
Service
associated with
Virtual Entity
Appliance
relates to
Physical Entity
Appliance
acts on
exposes
Resource
associated with
hosts
Device
Minicomputer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


invokes/subscribes
Virtual Entity
Room
relates to
Physical Entity
Room
monitors
associated with
Service
associated with
Virtual Entity
Appliance
relates to
Physical Entity
Appliance
exposes
acts on
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

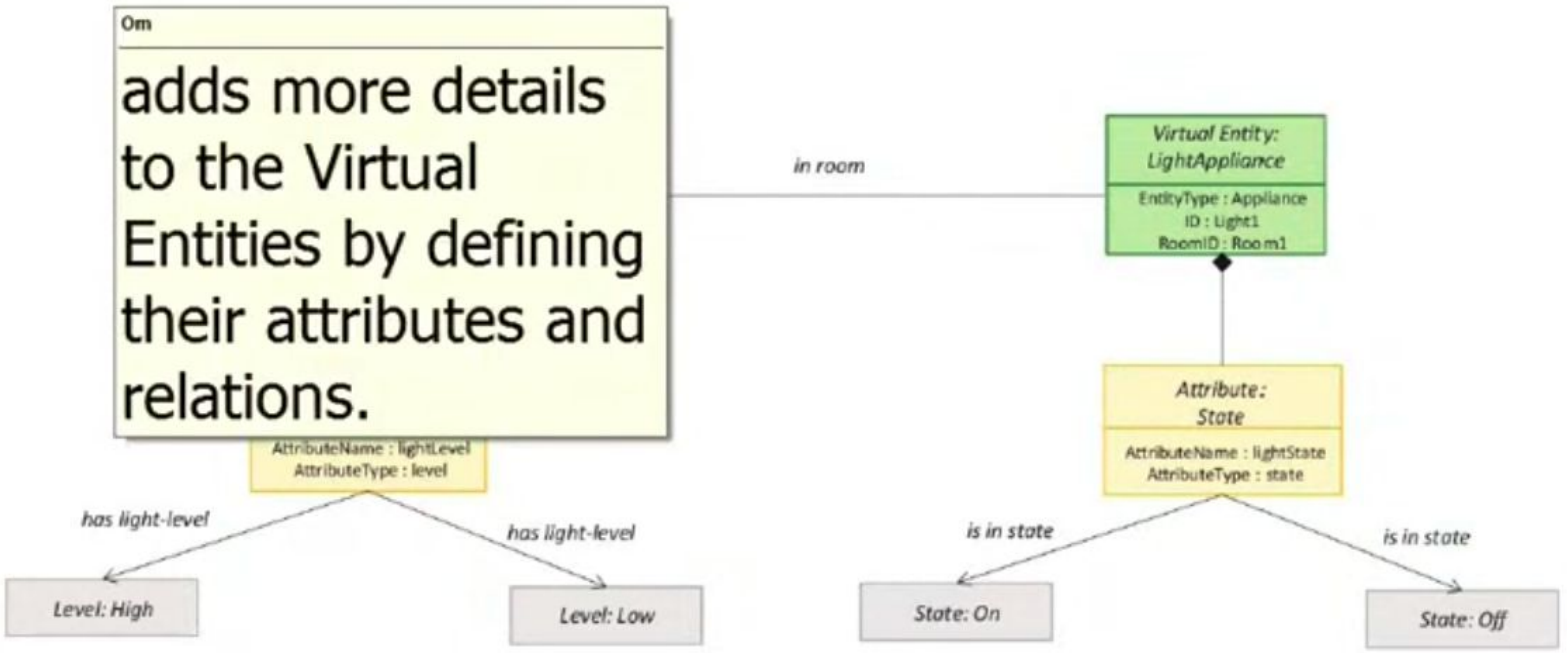
Step 4: Information Model Specification

- 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 4: Information Model Specification



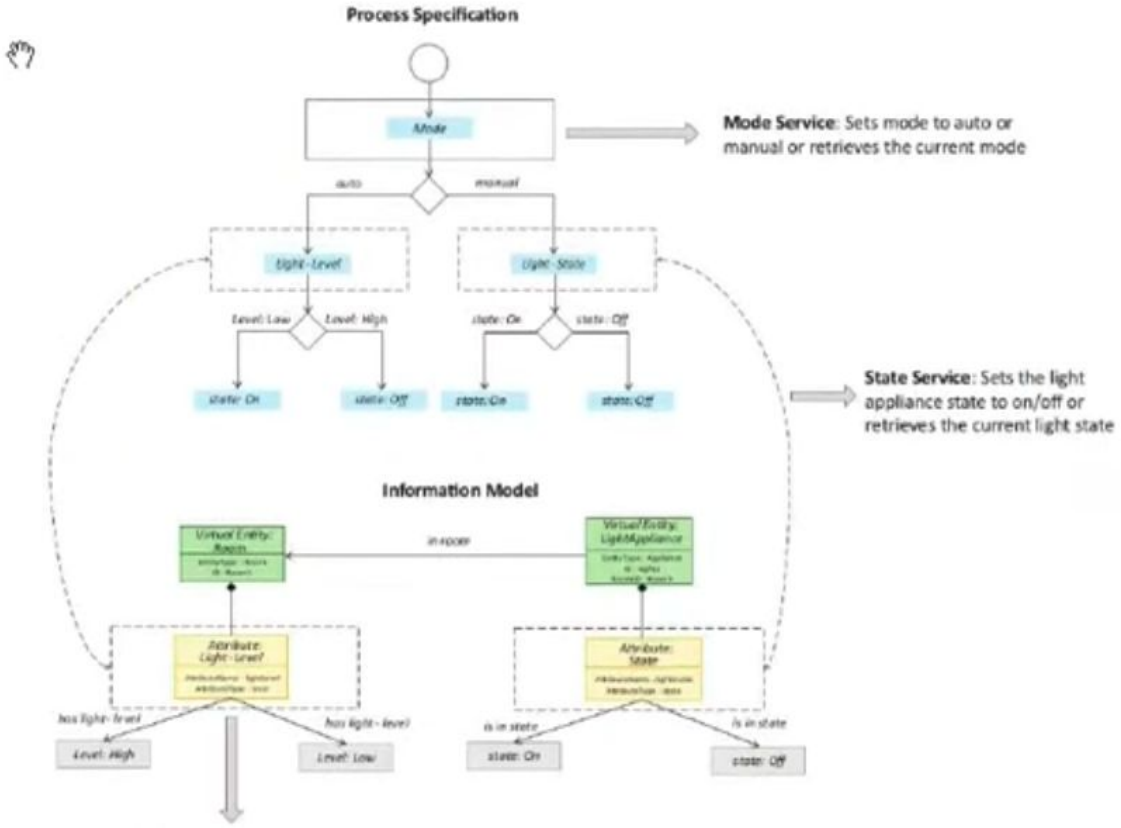
Step 4: Information Model Specification



Step 5: Service Specifications

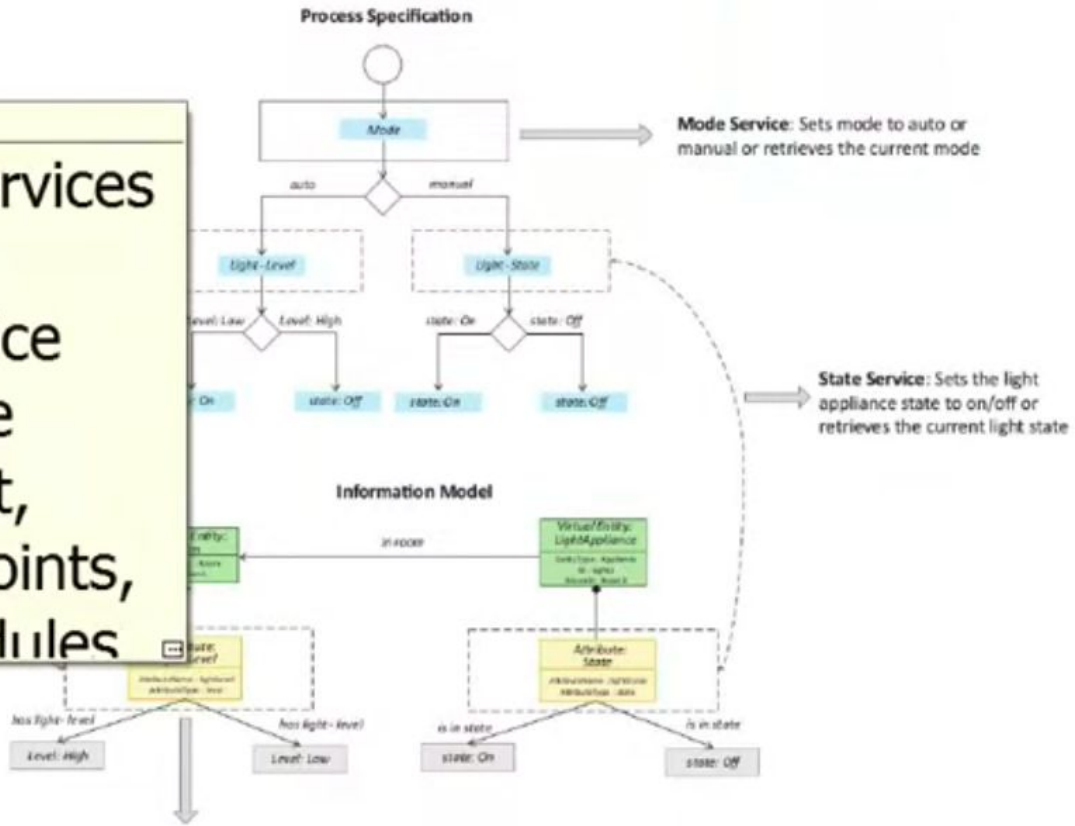
- The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 5: Service Specifications

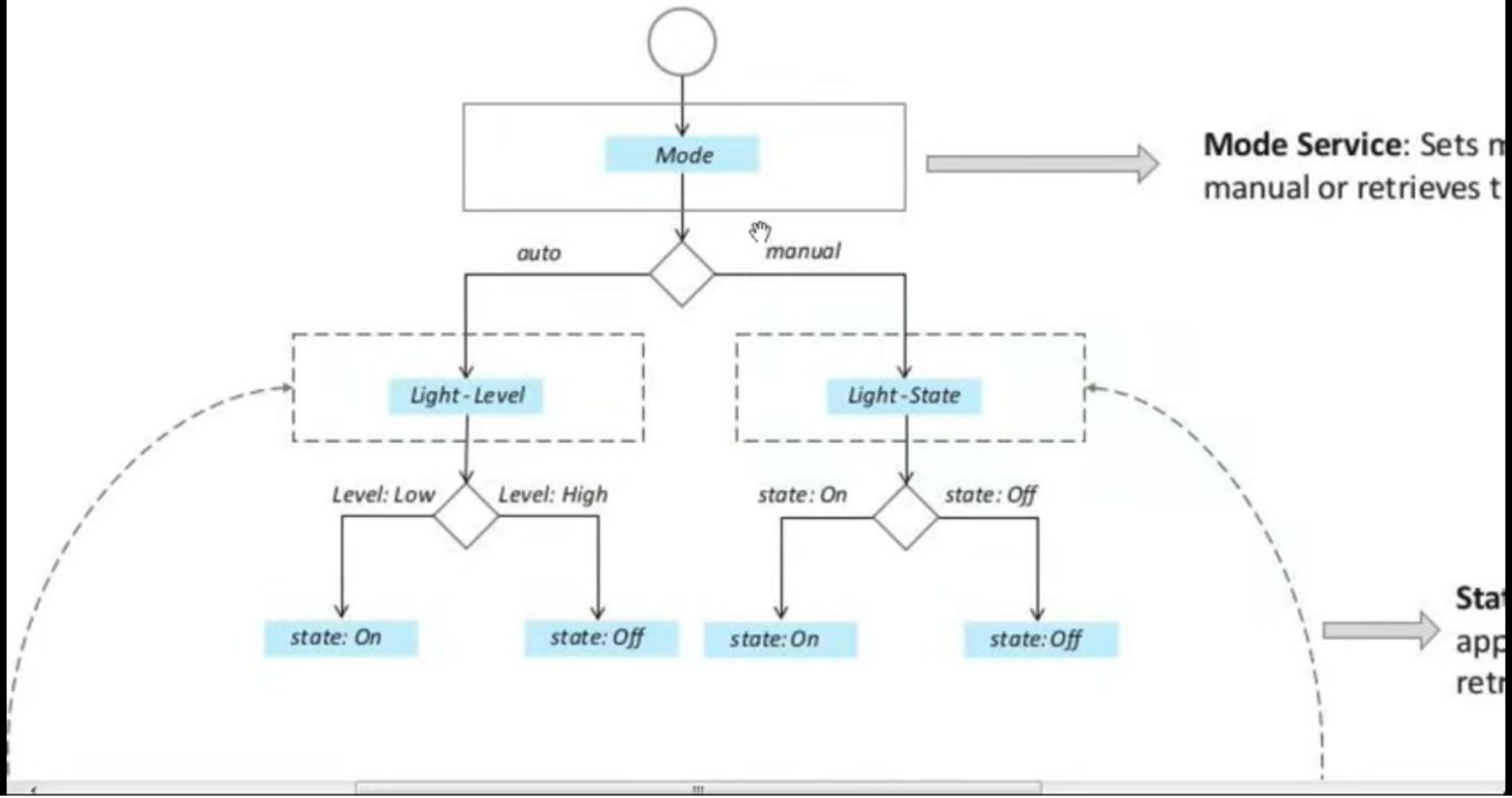


Step 5: Service Specifications

define the services
n the IoT
system, service
ypes, service
nputs/output,
service endpoints,
service schedules

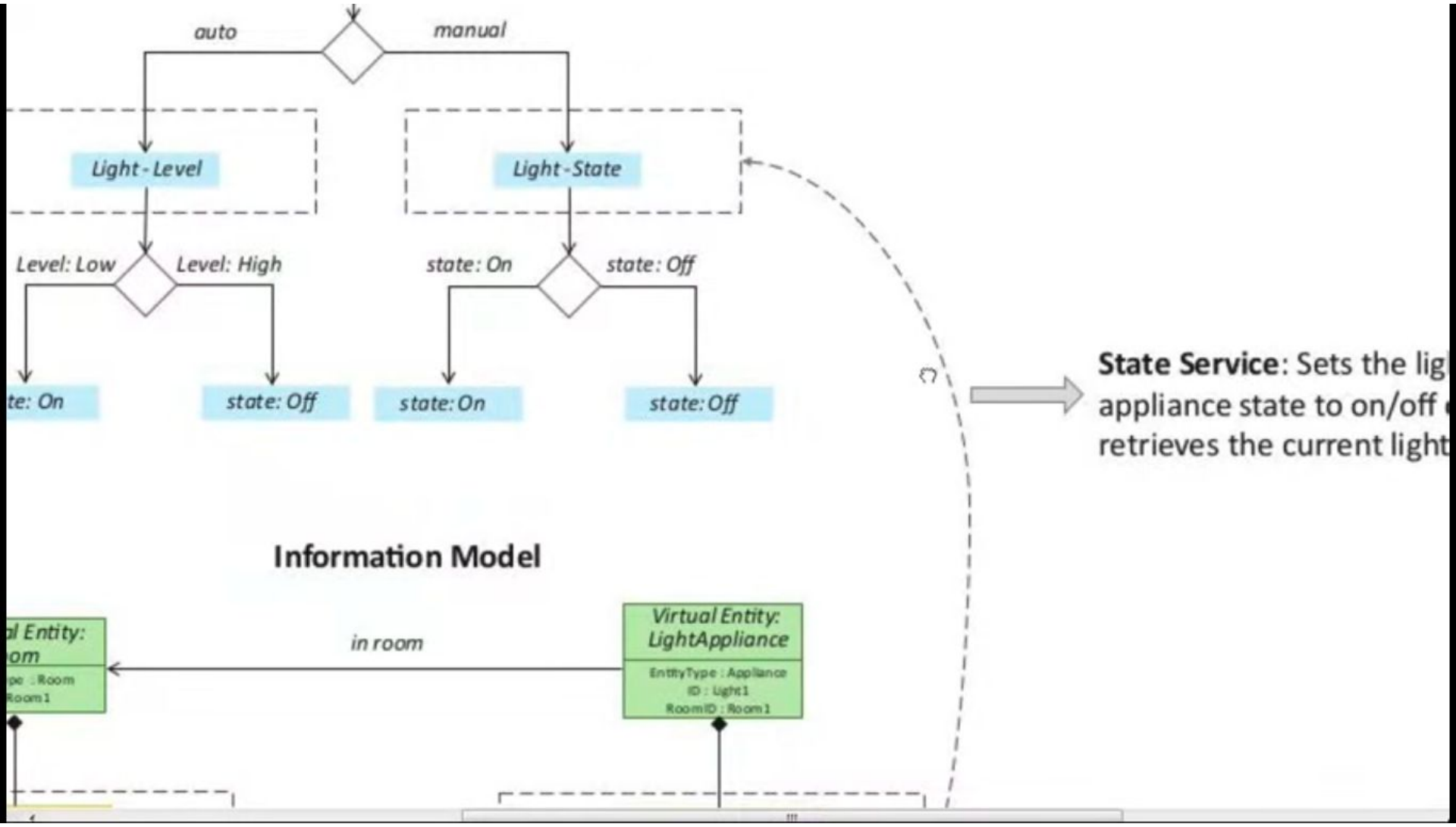


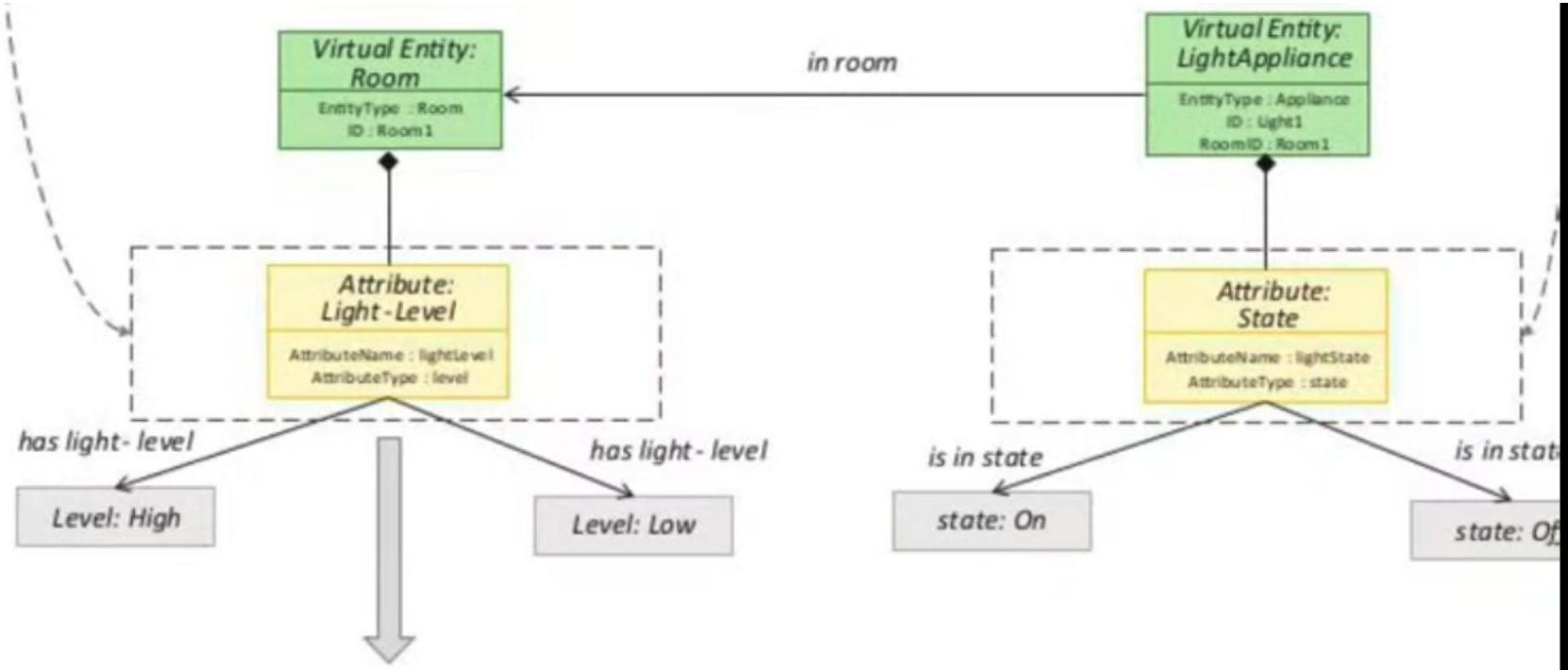
Process Specification



Mode Service: Sets m
manual or retrieves t

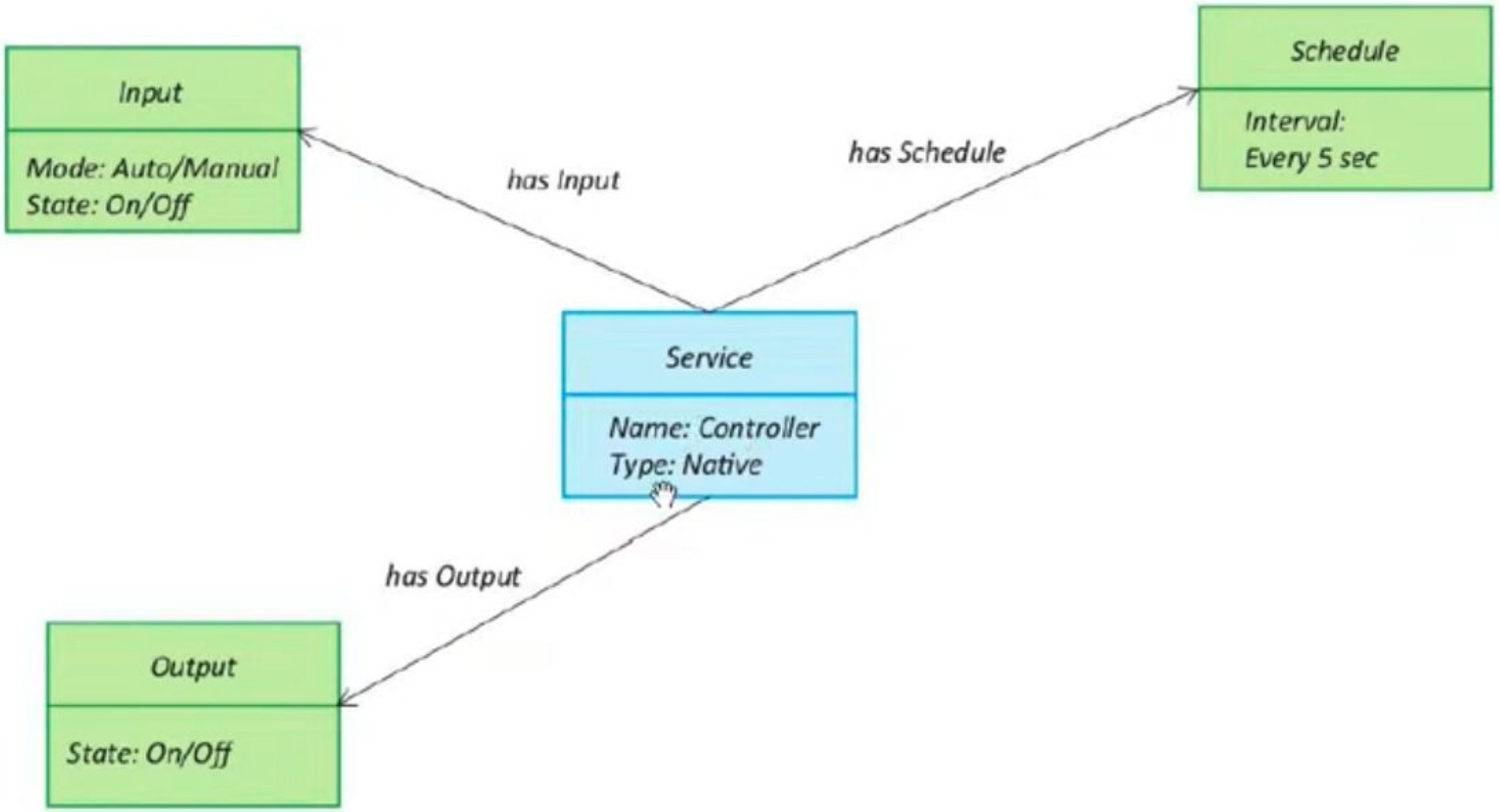
Stat
app
retr

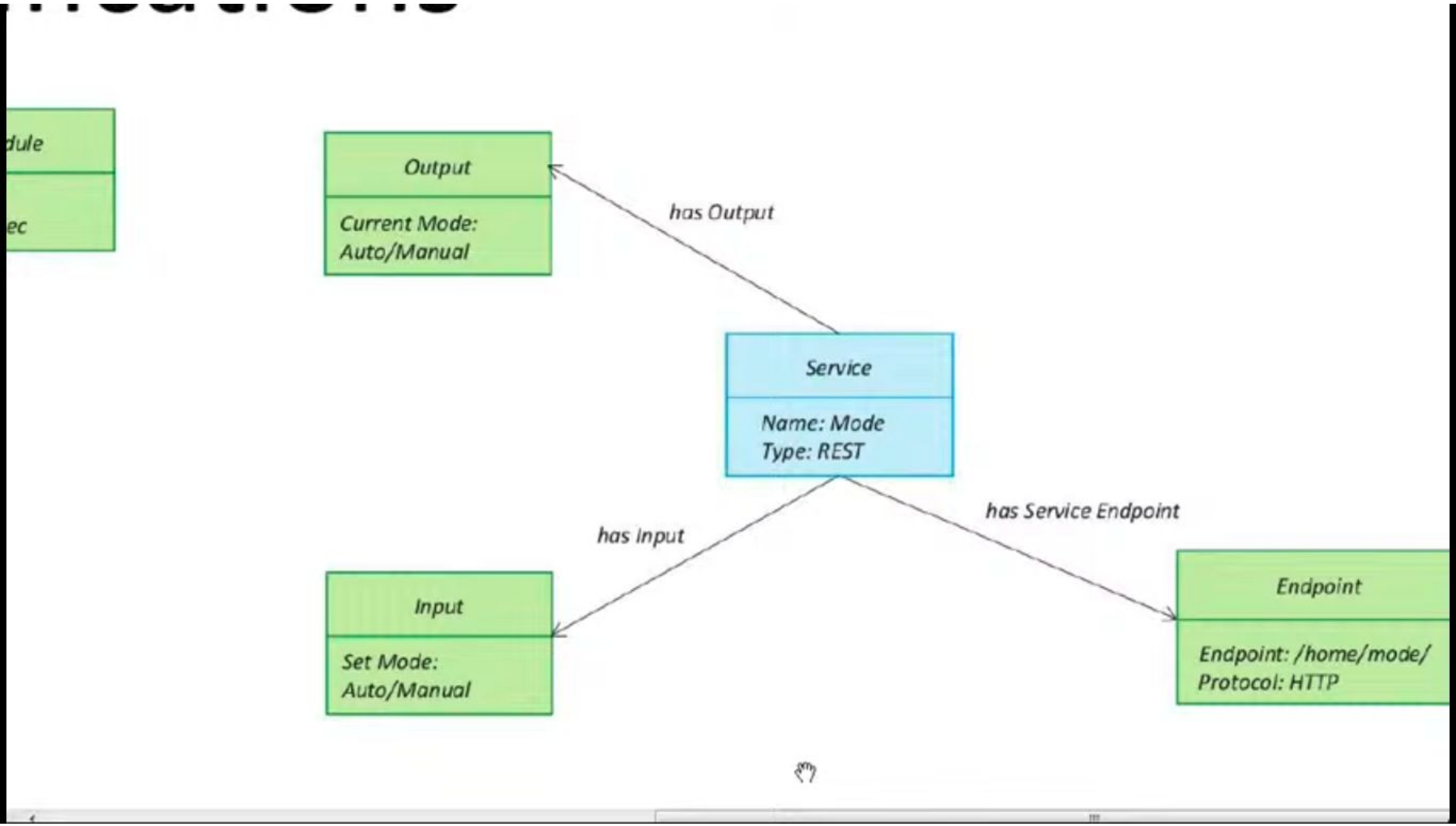

auto
manual
Light - Level
Light - State
Level: Low
Level: High
state: On
state: Off
te: On
state: Off
state: On
state: Off
State Service: Sets the lig
appliance state to on/off
retrieves the current light
Information Model
in room
Virtual Entity:
LightAppliance
EntityType : Appliance
ID : Light1
RoomID : Room1

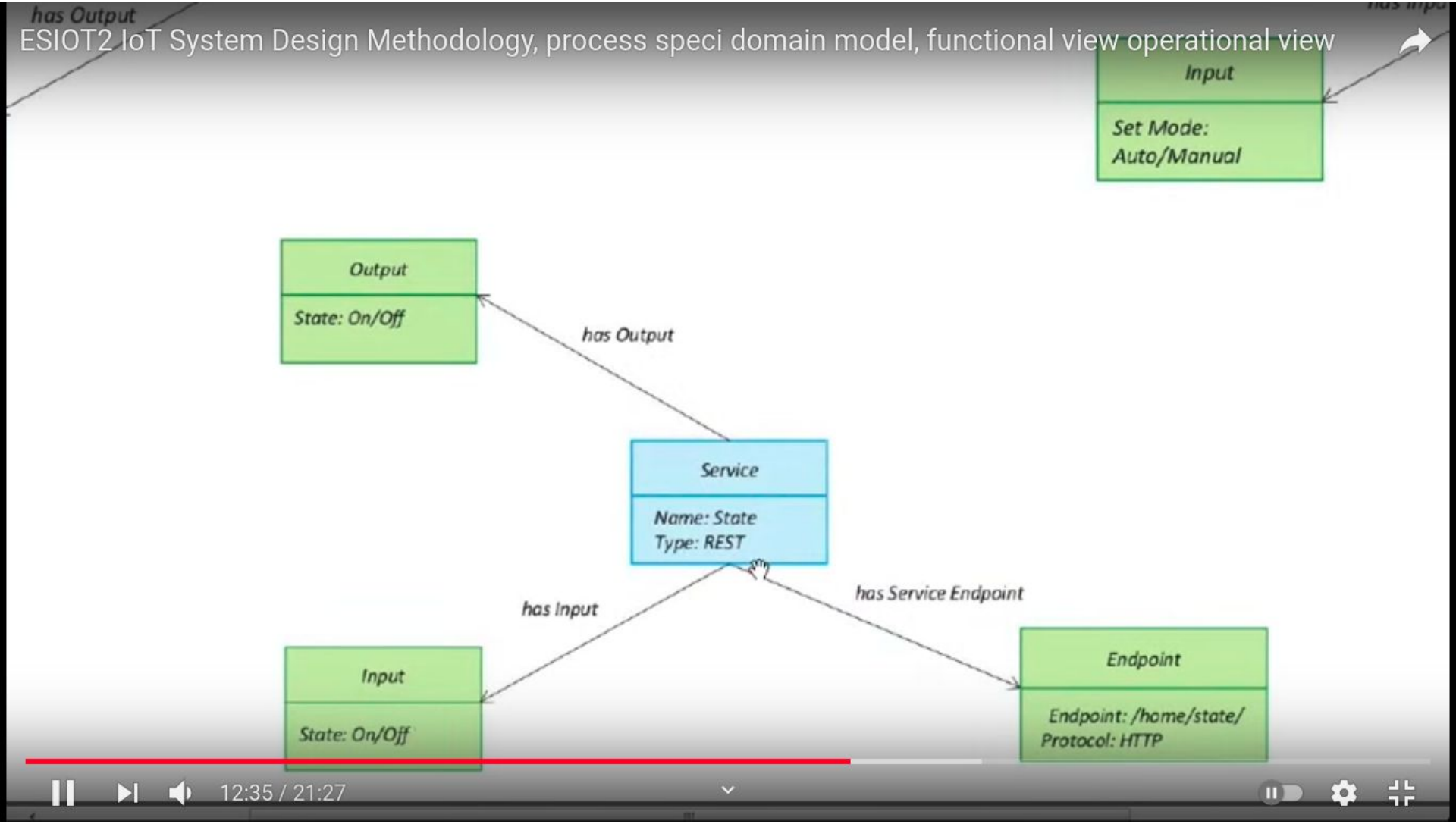


Controller Service: In auto mode, the controller service monitors the light level and switches the light on/off and updates the status in the status database. In manual mode, the controller service, retrieves the current state from the database and switches the light on/off.

Step 5: Service Specificati



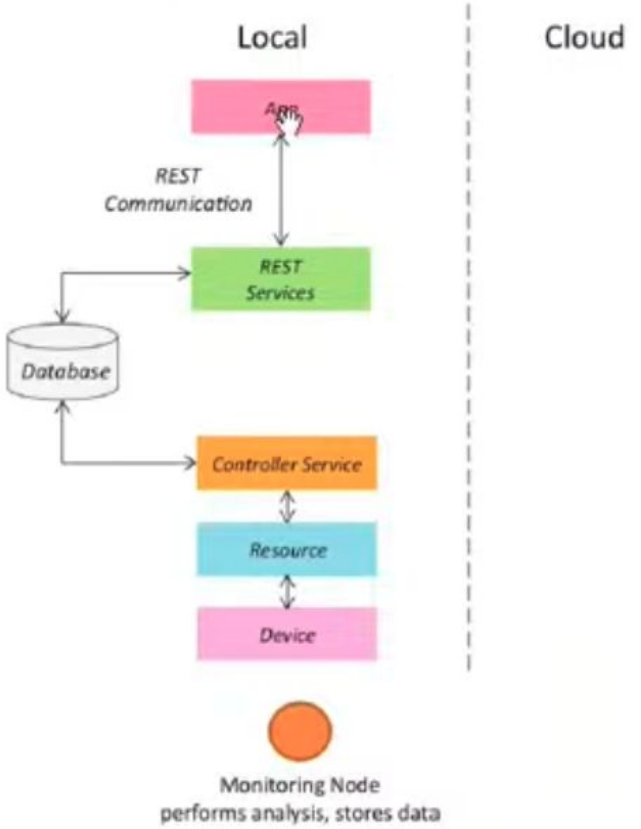

dule
ec
Output
Current Mode:
Auto/Manual
has Output
Service
Name: Mode
Type: REST
has Input
has Service Endpoint
Input
Set Mode:
Auto/Manual
Endpoint
Endpoint: /home/mode/
Protocol: HTTP


has Output
ESIOT2 IoT System Design Methodology, process speci domain model, functional view operational view
Input
Set Mode:
Auto/Manual
Output
State: On/Off
has Output
Service
Name: State
Type: REST
has Input
has Service Endpoint
Input
State: On/Off
Endpoint
Endpoint: /home/state/
Protocol: HTTP
12:35 / 21:27

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.

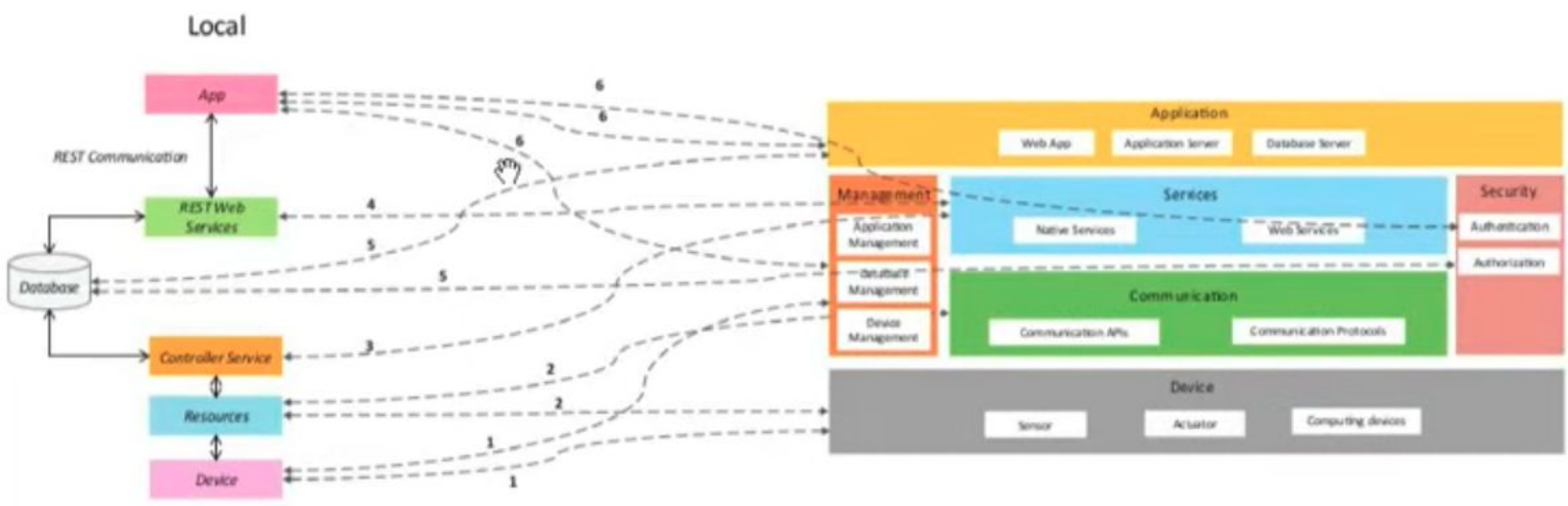
Step 6: IoT Level Specification



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 7: Functional View Specification



1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)

2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)

3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

4. Web Services map to Services FG (web services)

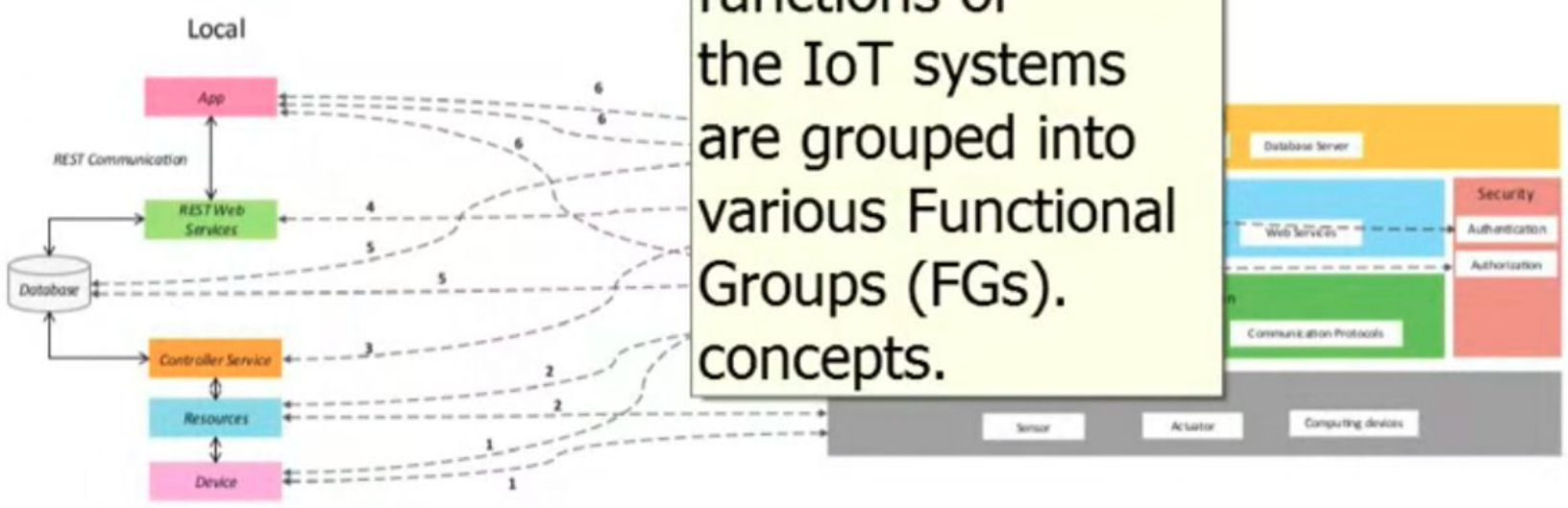
5. Database maps to the Management FG (database management) and Security FG (database security)

6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)

Step 7: Functional View Specification

Om

functions of
the IoT systems
are grouped into
various Functional
Groups (FGs).
concepts.



1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)

2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)

3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

4. Web Services map to Services FG (web services)

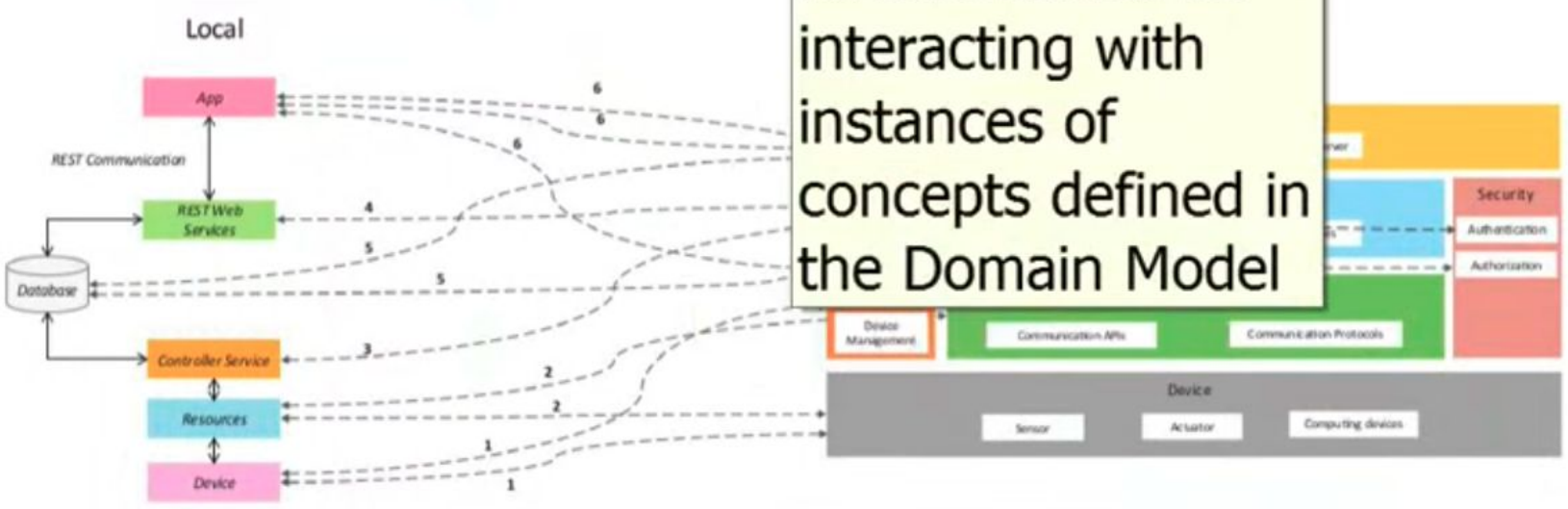
5. Database maps to the Management FG (database management) and Security FG (database security)

6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)

Step 7: Functional View Specification

Om

functionalities for interacting with instances of concepts defined in the Domain Model



1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)

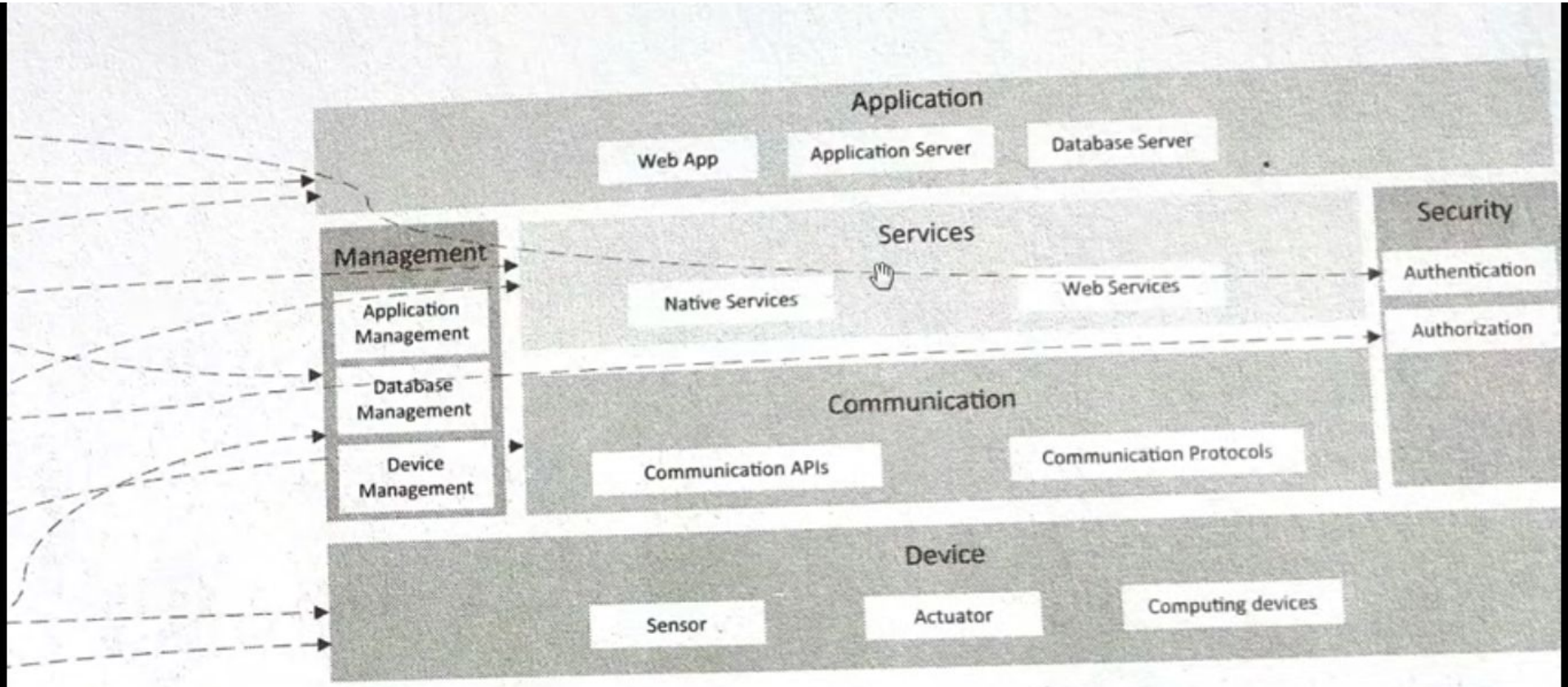
2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)

3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

4. Web Services map to Services FG (web services)

5. Database maps to the Management FG (database management) and Security FG (database security)

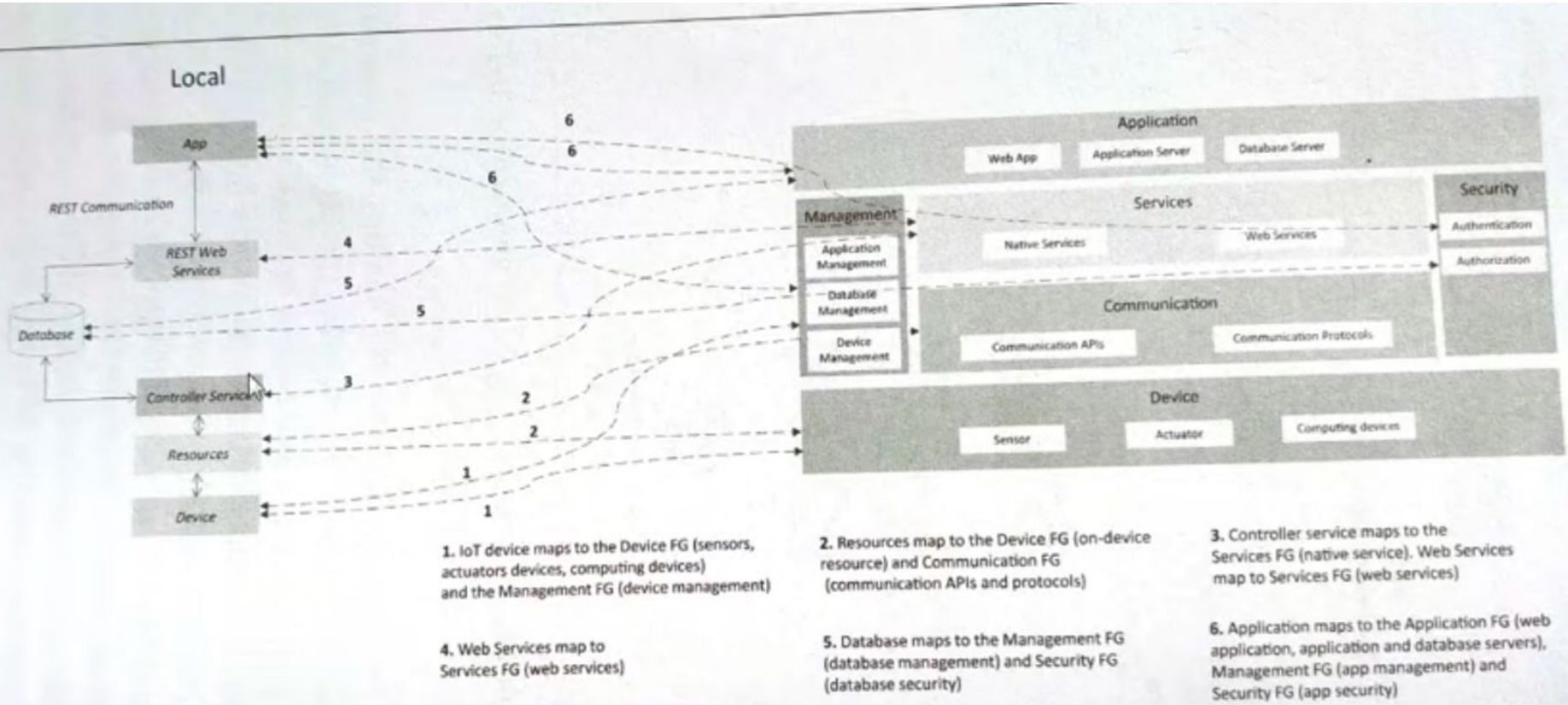
6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)



e Device FG (sensors,

2. Resources map to the Device FG (on-device

3. Controller service maps to the



1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)

2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)

3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

4. Web Services map to Services FG (web services)

5. Database maps to the Management FG (database management) and Security FG (database security)

6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)

Figure 5.10: Mapping deployment level to functional groups for home automation IoT system

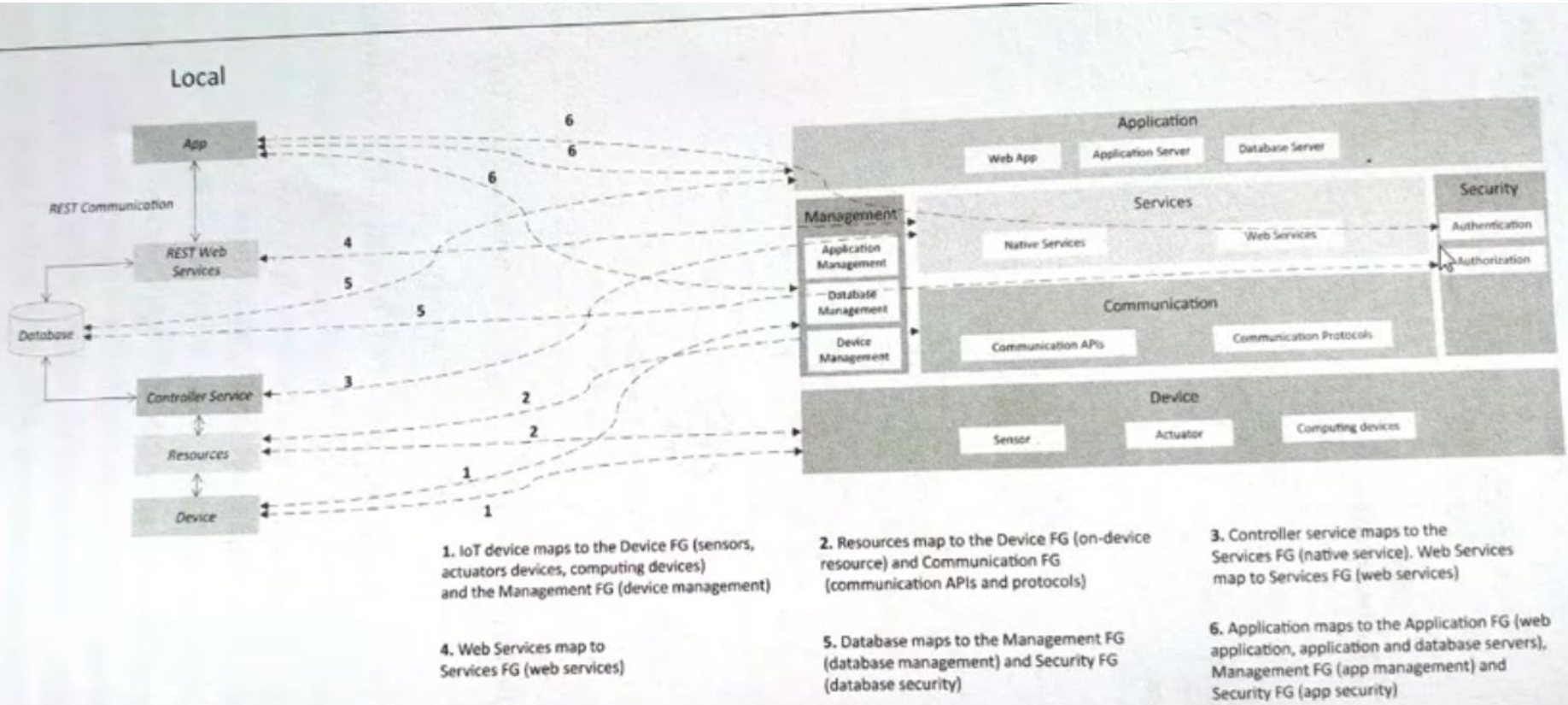
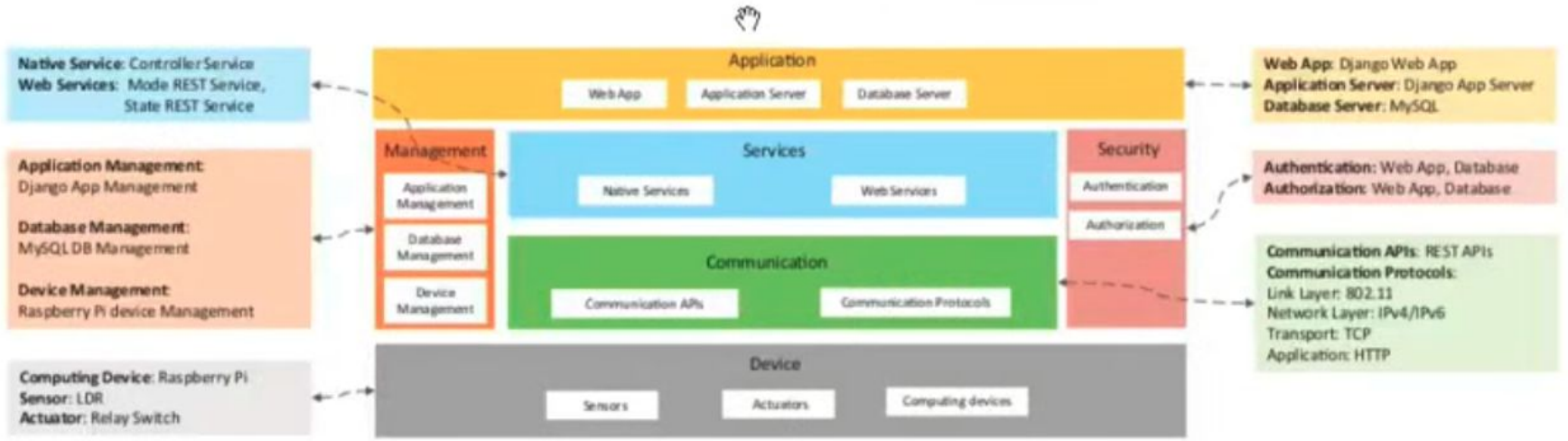


Figure 5.10: Mapping deployment level to functional groups for home automation IoT system

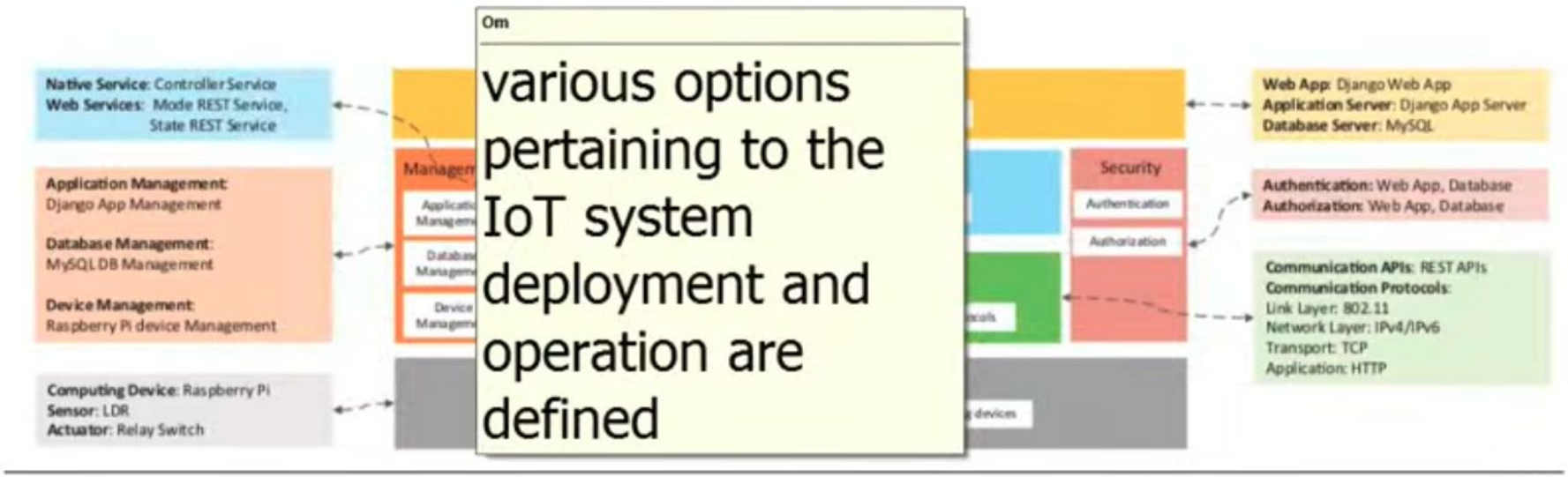
Step 8: Operational View Specification

- 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

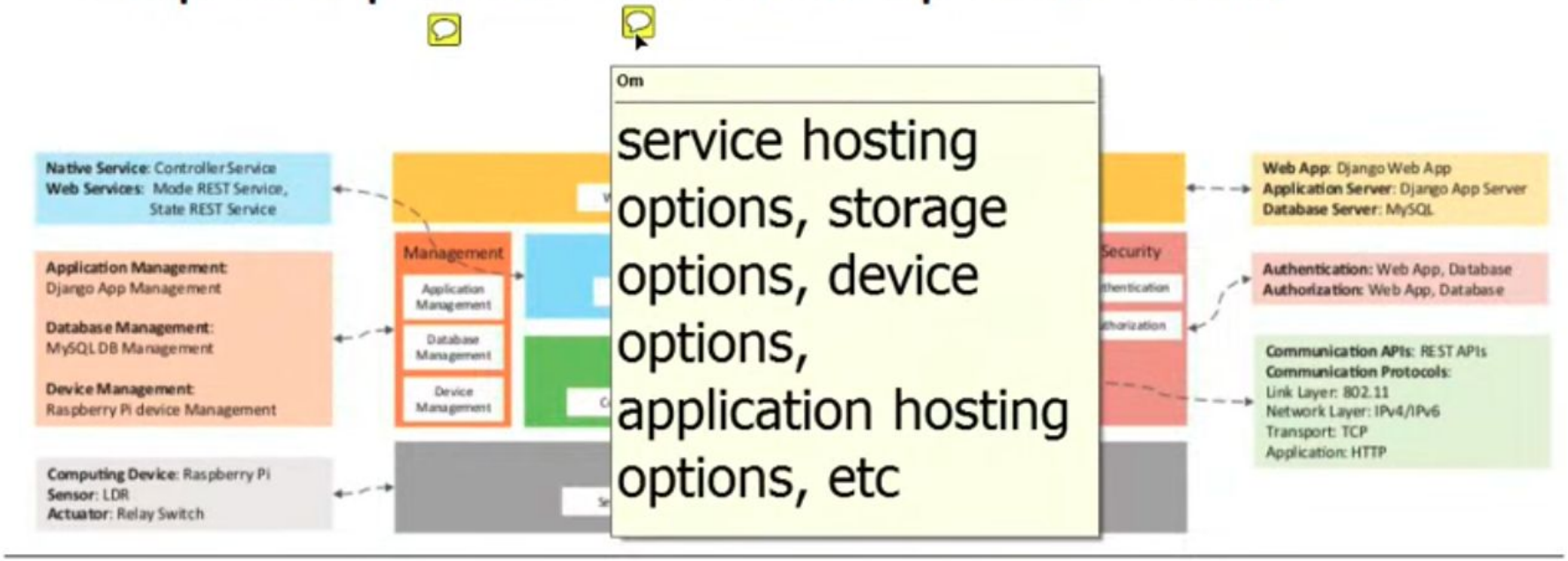
Step 8: Operational View Specification

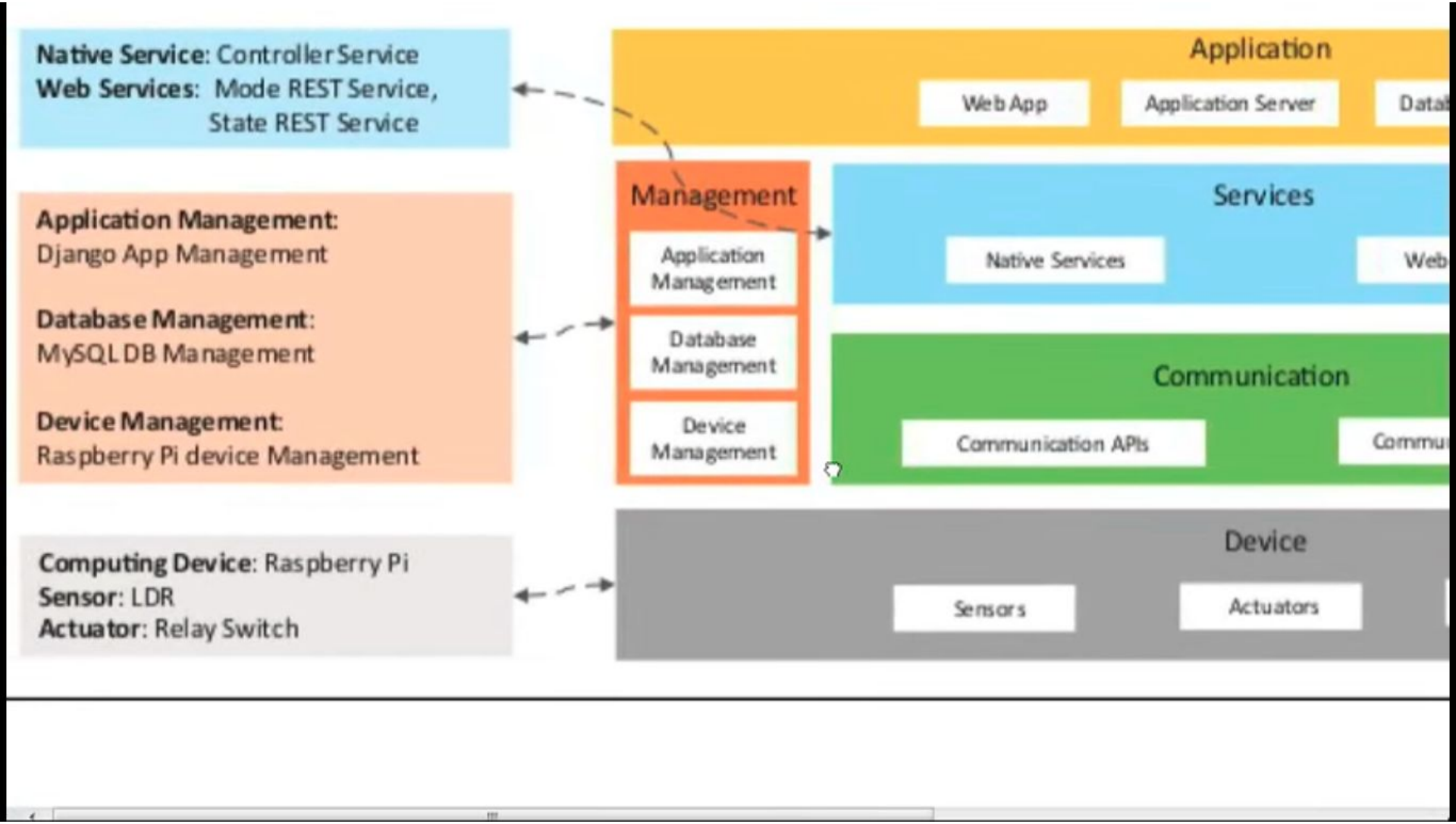


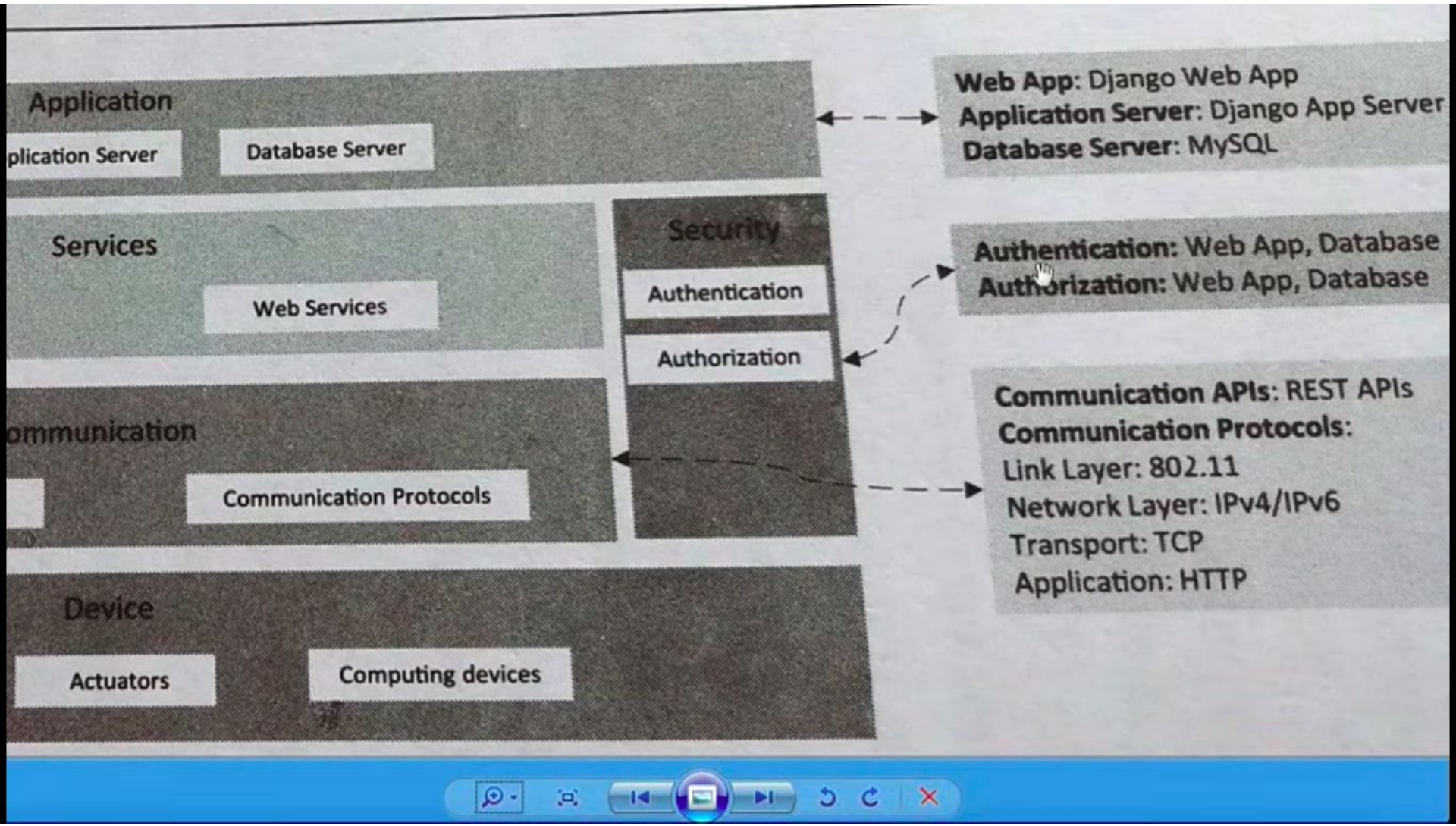
Step 8: Operational View Specification



Step 8: Operational View Specification



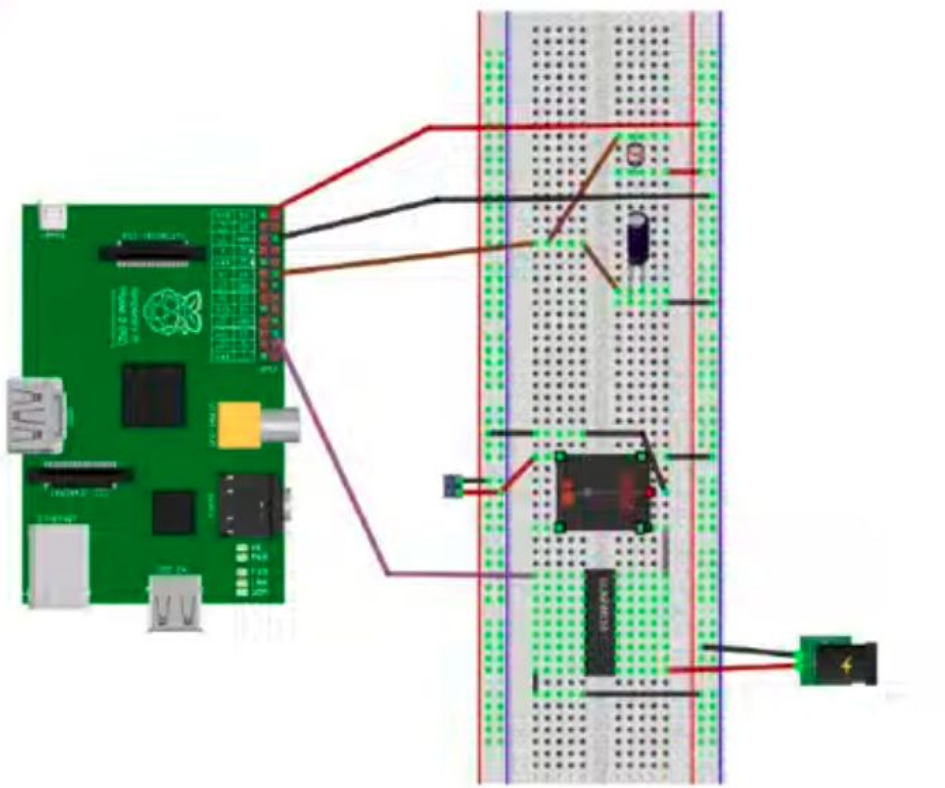

Native Service: Controller Service
Web Services: Mode REST Service,
State REST Service
Application Management:
Django App Management
Database Management:
MySQL DB Management
Device Management:
Raspberry Pi device Management
Computing Device: Raspberry Pi
Sensor: LDR
Actuator: Relay Switch
Application
Web App
Application Server
Management
Application Management
Database Management
Device Management
Services
Native Services
Communication
Communication APIs
Device
Sensors
Actuators


Application
plication Server
Database Server
Services
Web Services
Security
Authentication
Authorization
ommunication
Communication Protocols
Device
Actuators
Computing devices
Web App: Django Web App
Application Server: Django App Server
Database Server: MySQL
Authentication: Web App, Database
Authorization: Web App, Database
Communication APIs: REST APIs
Communication Protocols:
Link Layer: 802.11
Network Layer: IPv4/IPv6
Transport: TCP
Application: HTTP

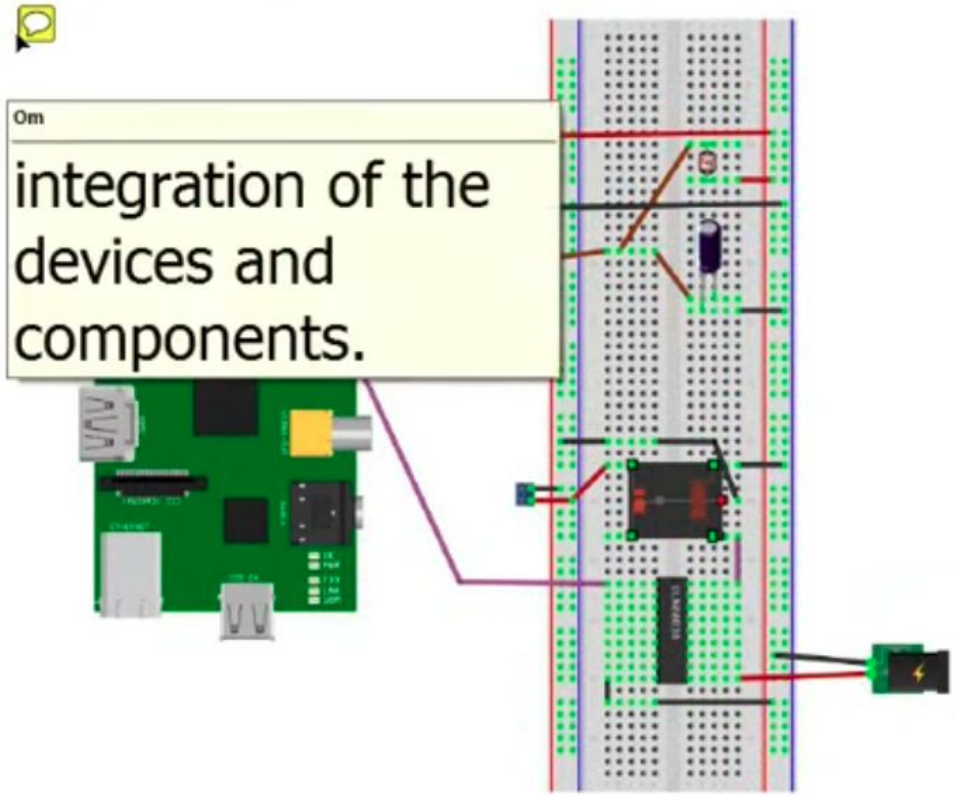
Step 9: Device & Component Integration

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

Step 9: Device & Component Integration



Step 9: Device & Component Integration



Step 10: Application Development

- The final step in the IoT design methodology is to develop the IoT application.

Slide 63

Step 10: Application Development

- 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.