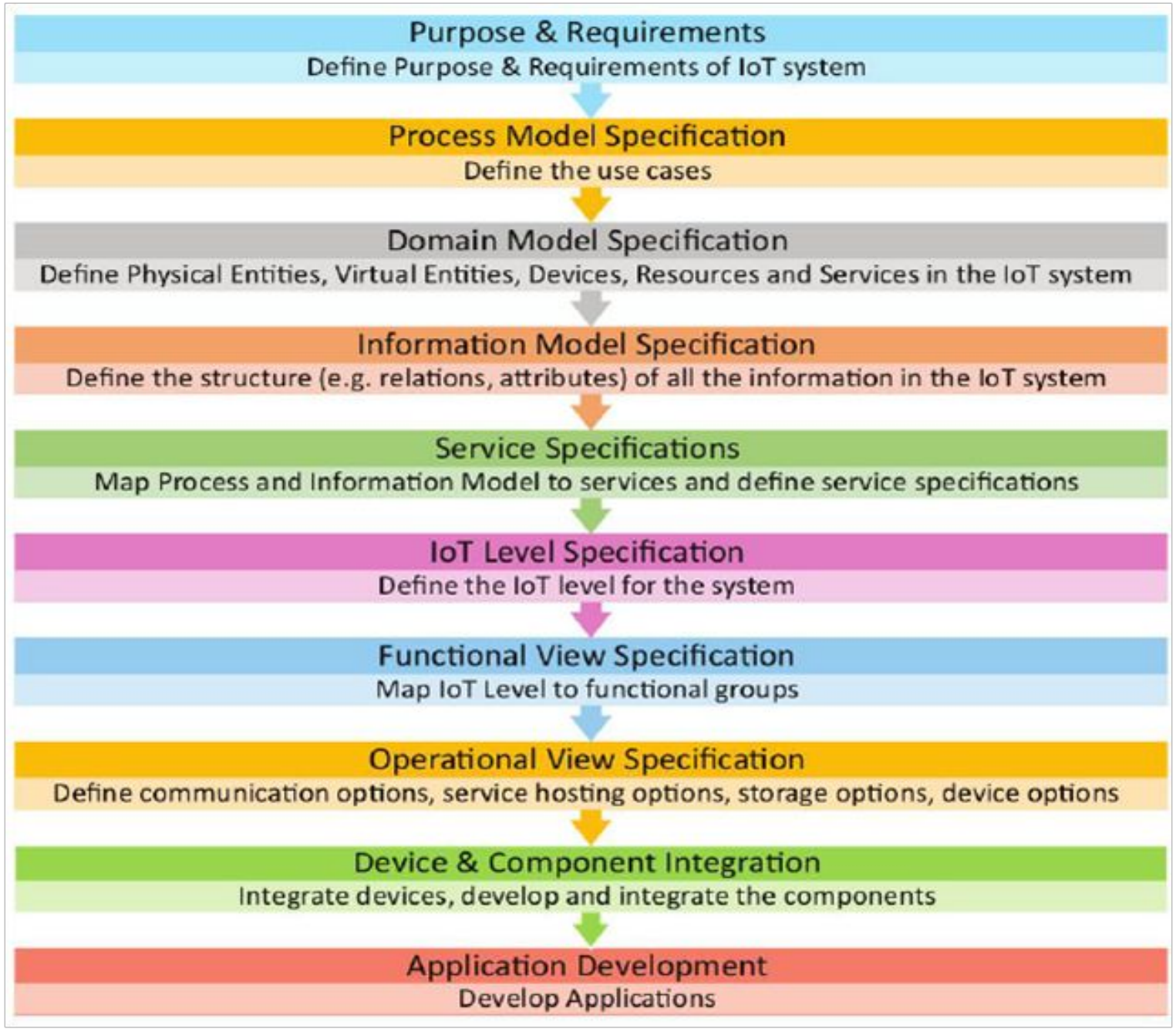
Internet of Things

Course Instructor: Akshaya K Pati

IoT Platforms Design Methodology

- Designing IoT systems can be a complex and challenging task as these systems involve interactions between various components.
- A generic design methodology which is independent of specific product, service or programming language.
- IoT systems designed with this methodology will have reduced design time, testing time, maintenance time, complexity and better interoperability.

The steps involved in the designing of an IoT system or application



Purpose and Requirements Specification

- In this step, the system purpose, behavior and requirements are captured.

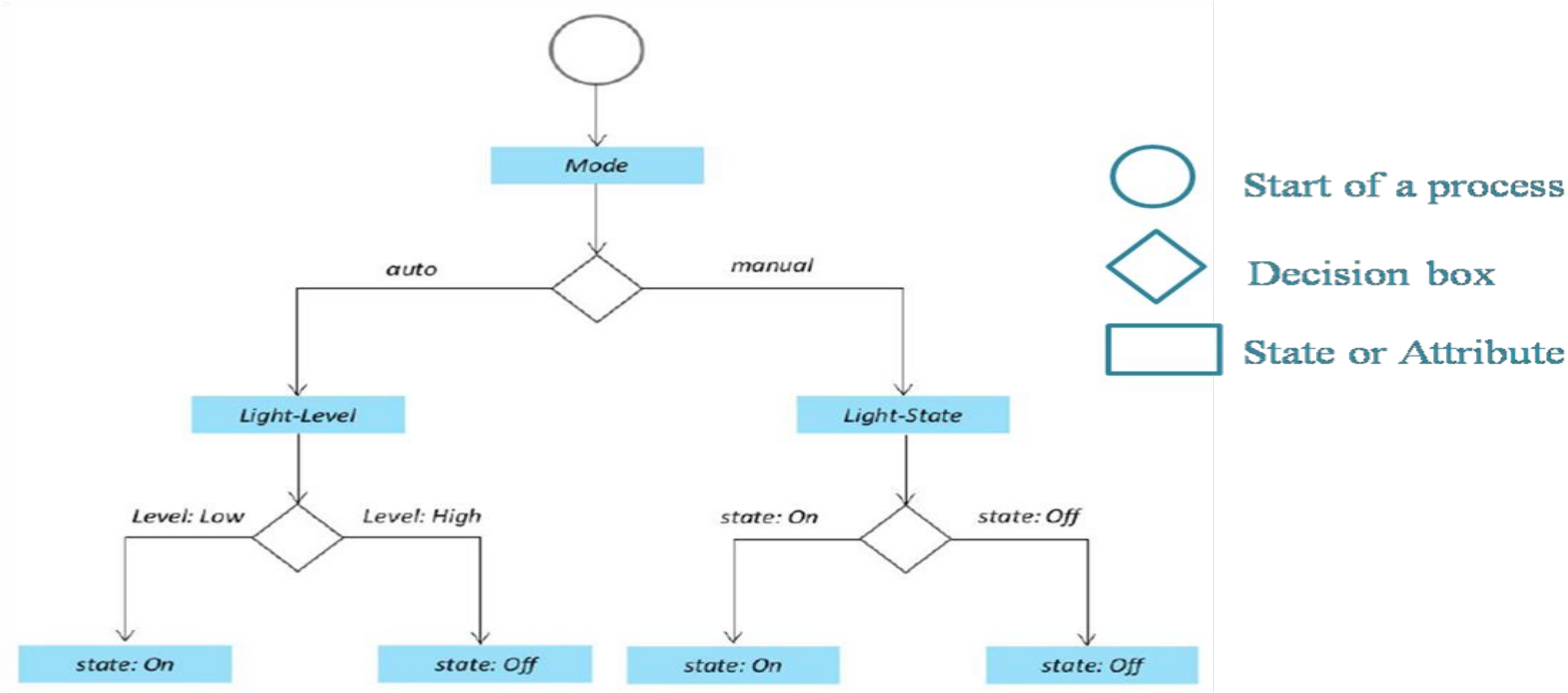
- Requirements can be:

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

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 use cases of the IoT system are formally described based on and derived from the purpose and requirements specifications.

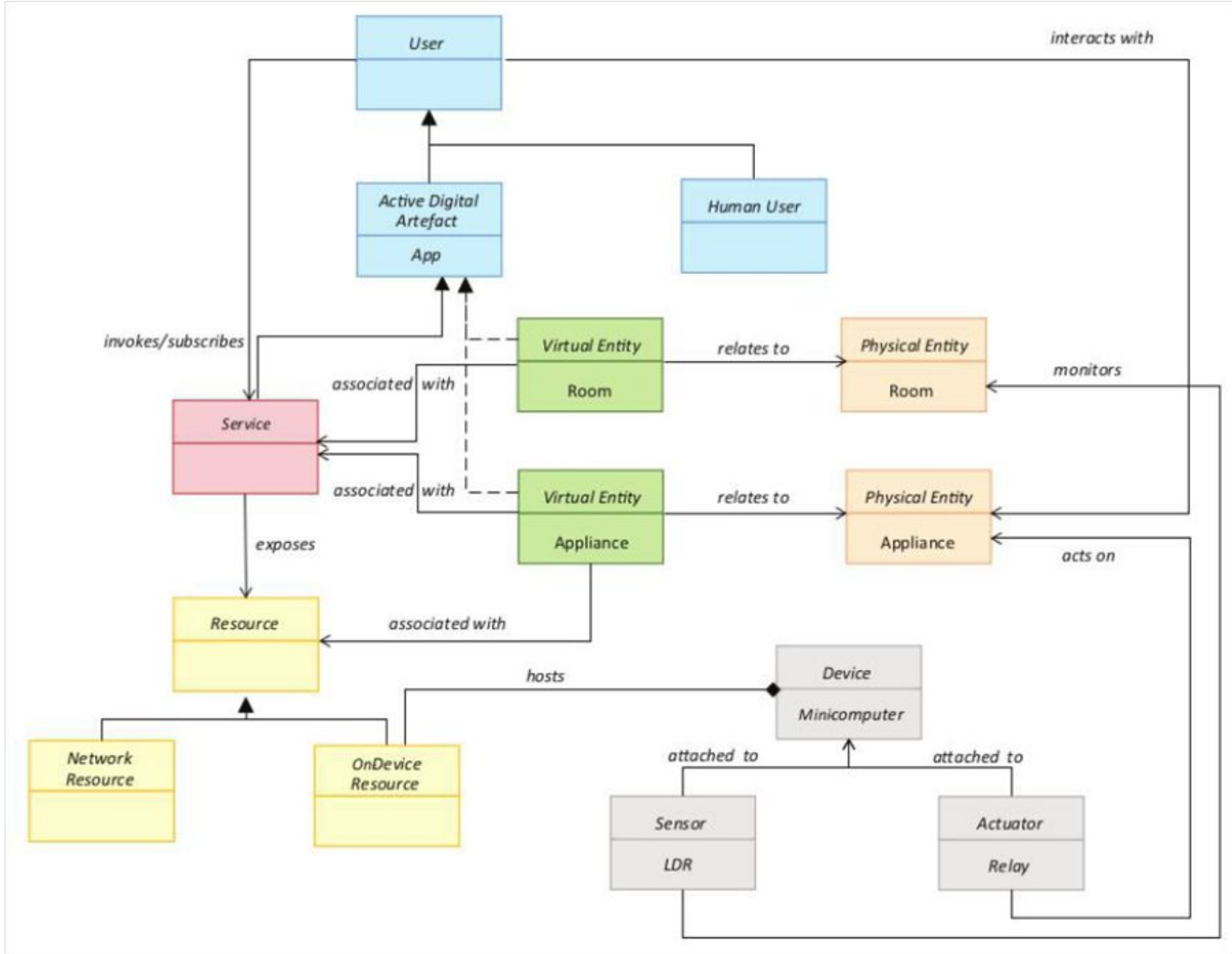


Step 3:Domain Model Specifications

- It defines the attributes of the object and reltionship between the objects
- It provides an abstract representaion of the cncept, objects, and enitites in the IoT domain, indepedent of any specific technology or platform.
- In this step, the designer can understandng of of the IoT domain for which the system is to be designed.
- This model includes the terms as:
- **Physical Entity**:
- A physical Entity is a discrete and identifiable entity in the physical environment (e.g. a room, a light, an appliance, a car, etc.).
- The IoT system provides information about the Physical Entity (using sensors) or performs actuation upon the Physical Entity (e.g., switching on a light).
- **Virtual Entity:**
- A virtual Entity is a representation of the Physical Entity in the digital world.

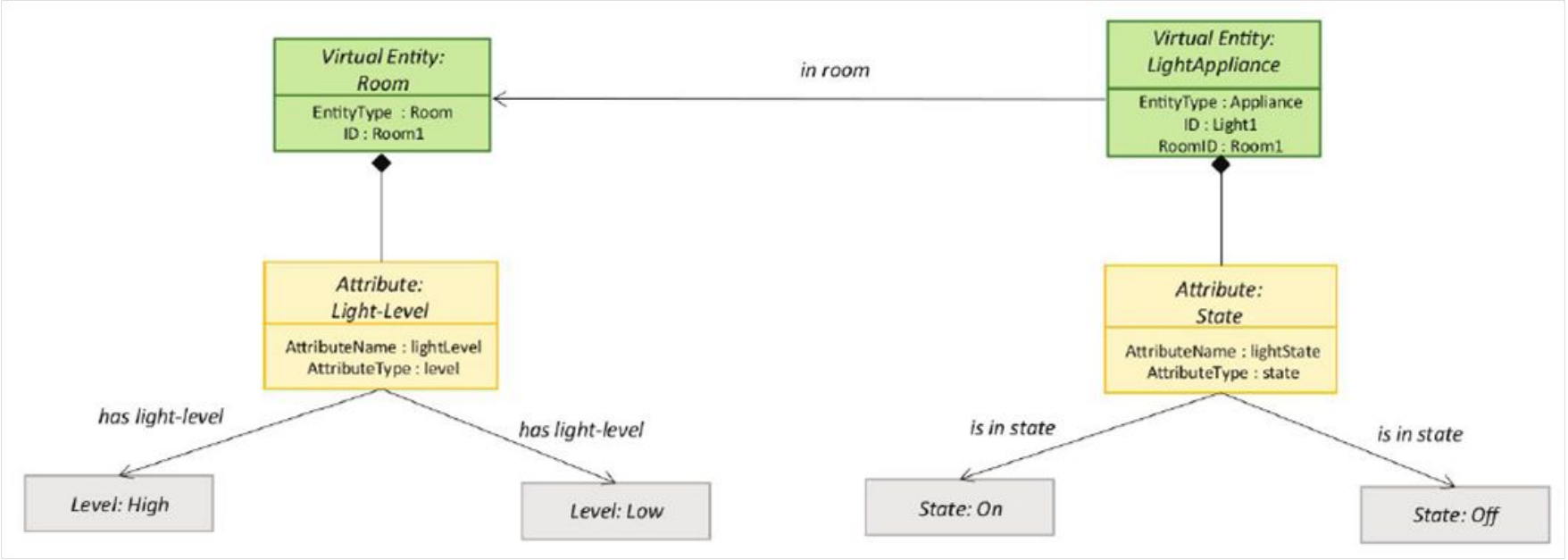
- **Device:**
- The device provides a medium for interactions between Physical Entities and Virtual Entities. Devices are either attached to Physical Entities or placed near Physical Entities.
- Devices are used to gather information about Physical Entities (e.g., from sensors), perform actuation upon Physical Entities (e.g. using actuators), or used to identify Physical Entities (e.g., using tags).
- **Resource:**
- Resources are software components that can be either "on-device" or "network resources".
- On-device resources are hosted on the device and include software components that either provide information on or enable actuation upon the Physical Entity to which the device is attached.
- Network resources include the software components that are available in the network (as a database).

- **Service**
- Services provide an interface for interacting with the Physical Entity
- Services access the resources hosted on the device or the network resources to obtain information about the Physical Entity or perform actuation upon the Physical Entity.


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
LDR
Actuator
Relay

Step 4: Information Model Specific

- This step defines Information Model.
- The Information Model defines the structure of all the information in the IoT system. Example, attributes of Virtual Entities, relations, etc.
- The information model does not describe “how the information is represented or stored”.
- It adds more details to the virtual entities by defining their attributes and relations

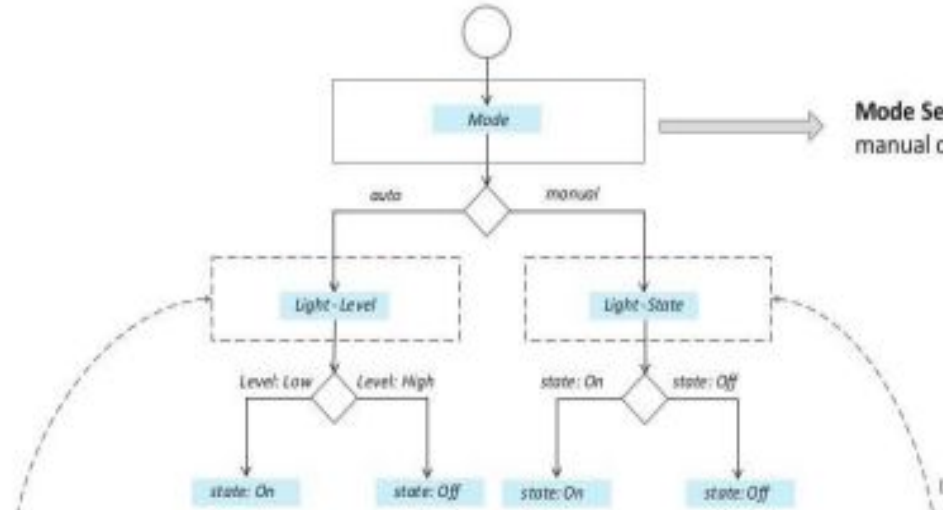


Step 5: Service Specifications

- Service specifications define the services in the IoT system such as service types, service inputs/output, service endpoints, service schedules, service preconditions, and service effects.
- These services either change the state or attribute values or retrieve the current values.
- Example: Home automation:
- Mode services set modes to auto or manual or retrive the current mode
- State services sets the light appliance state to on/off or retrieve the current ight state
- Controller service monitors the light level in auto mode and switches the light on/off and update the status in the status database

- The Mode service is a RESTful web service that sets the mode to auto or manual (PUT request), or retrieves the current mode (GET request).
- The mode is updated to/retrieved from the database.
- The State service is a RESTful web service that sets the light appliance state to on/off (PUT request) or retrieves the current light state (GET request).
- The state is updated to/retrieved from the status database.
- The Controller service runs as a native service on the device.

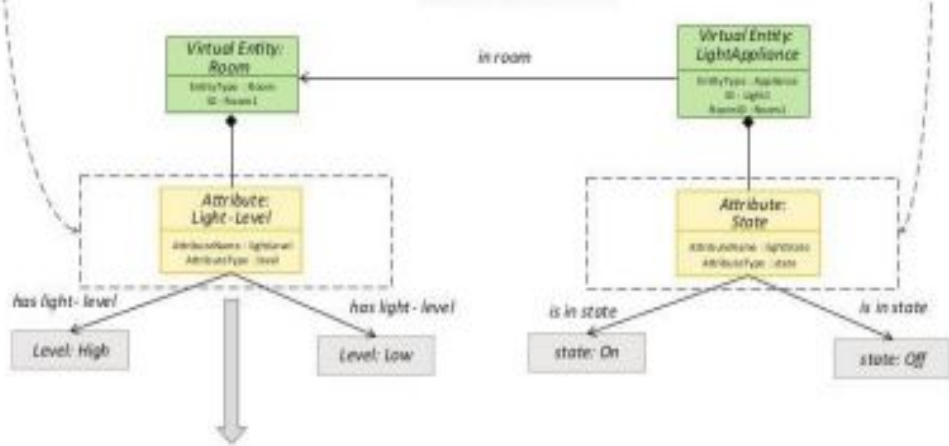
Process Specification



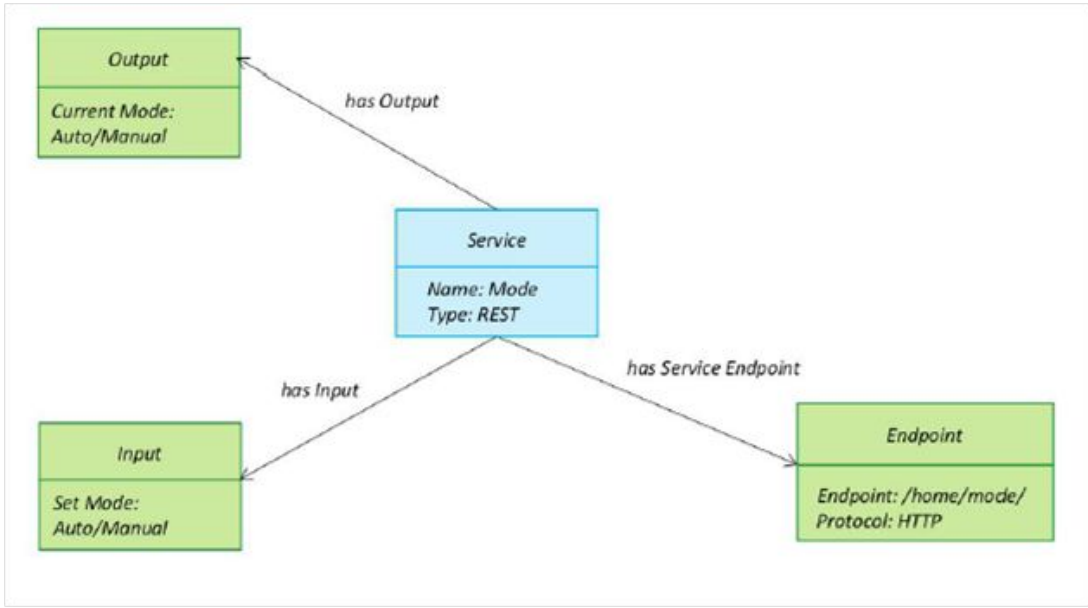
Mode Service: Sets mode to auto or manual or retrieves the current mode

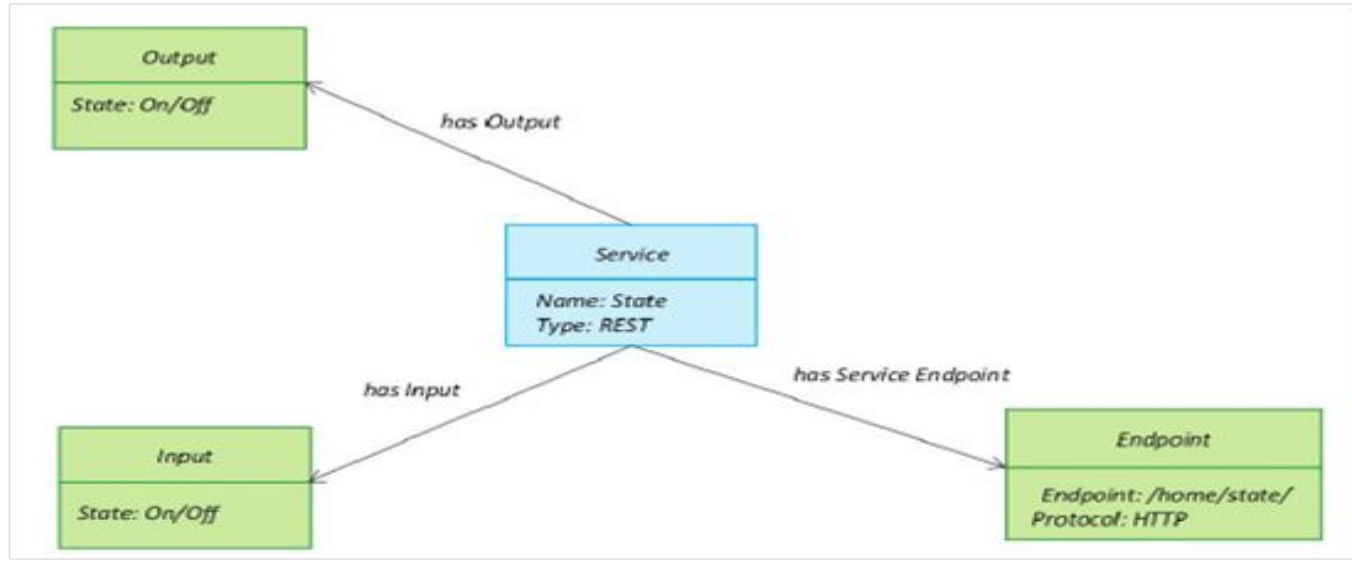
State Service: Sets the light appliance state to on/off or retrieves the current light state

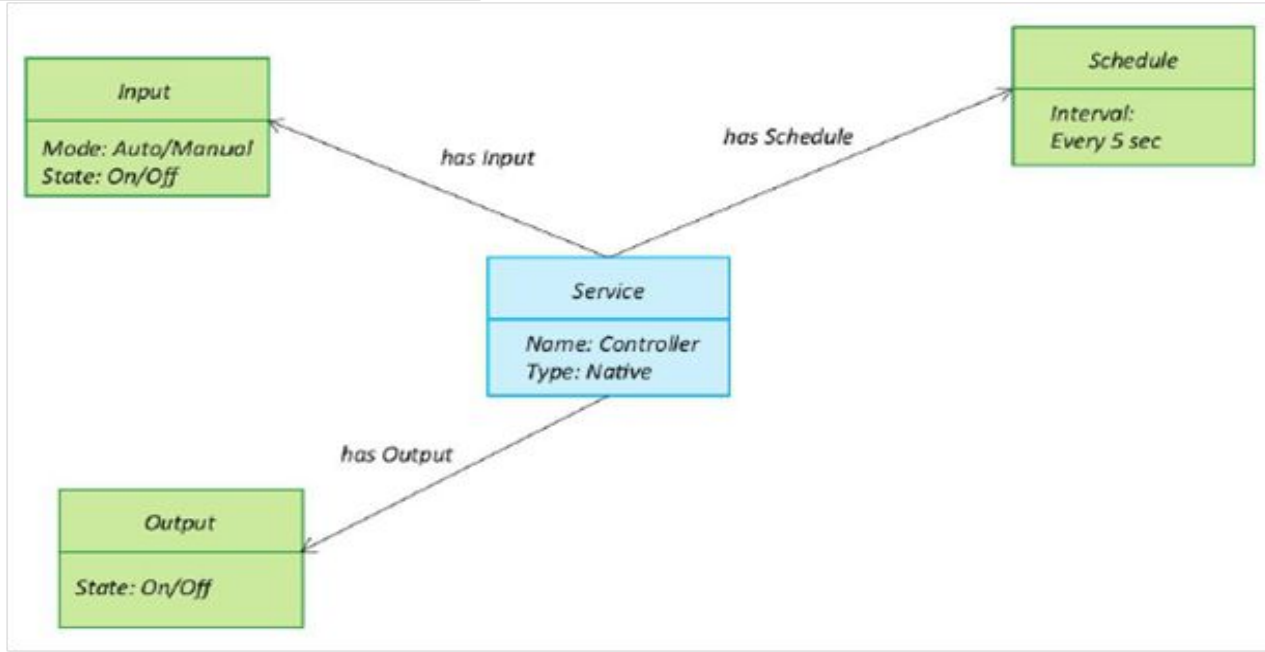
Information Model



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.


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


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


Input
Mode: Auto/Manual
State: On/Off
has Input
has Schedule
Schedule
Interval:
Every 5 sec
Service
Name: Controller
Type: Native
has Output
Output
State: On/Off

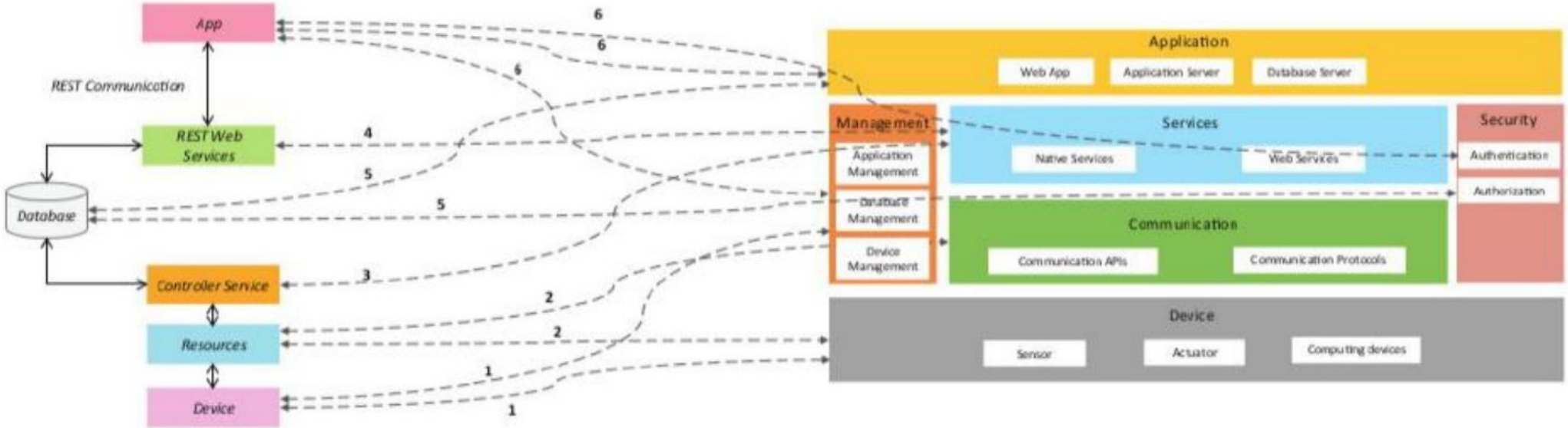
Step 6: IoT Level Specification

- IoT Level 1
- IoT Level 2
- IoT Level 3
- IoT Level 4
- IoT Level 5
- IoT Level 6

Step 7: Functional View Specification

- The Functional View (FV) defines the functions of the loT systems grouped into various Functional Groups (FGs).
- Each Functional Group either provides functionalities for interacting with instances of the Domain Model. It includes
- Device
- Communication
- Services
- Management
- Security
- Application

Local



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 8: Operational View Specification

- In this step define the Operational View Specifications. IoT system deployment and operation are defined, such as service hosting options, storage options, device options, application hosting options, etc

Devices:

- The 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 the device, implemented in Python, and run as a native service.
- 2. Mode service - REST-ful web service, hosted on a device, implemented with Django-REST Framework.
- 3. State service - REST-ful web service, hosted on a device, implemented with Django-REST Framework

Application:

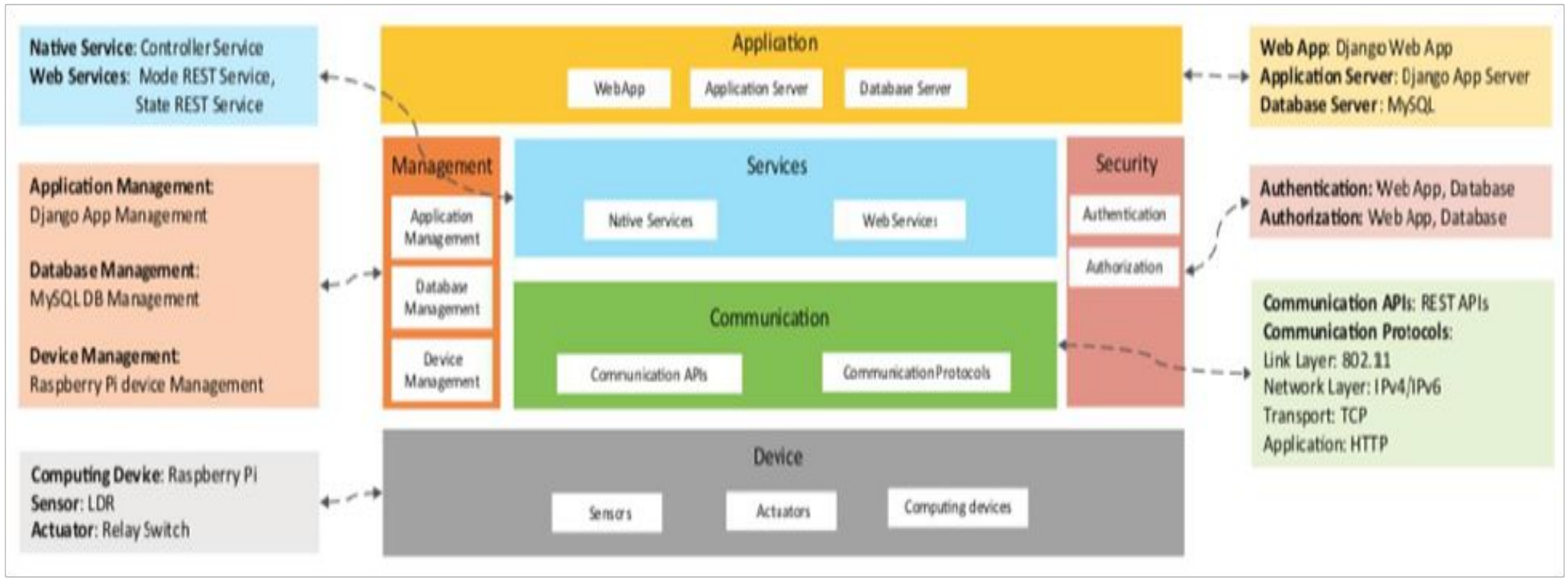
- Web Application - Django Web Application, Application Server - Django App Server, Database Server - MySQL.

Security:

- Authentication: Web App, Database

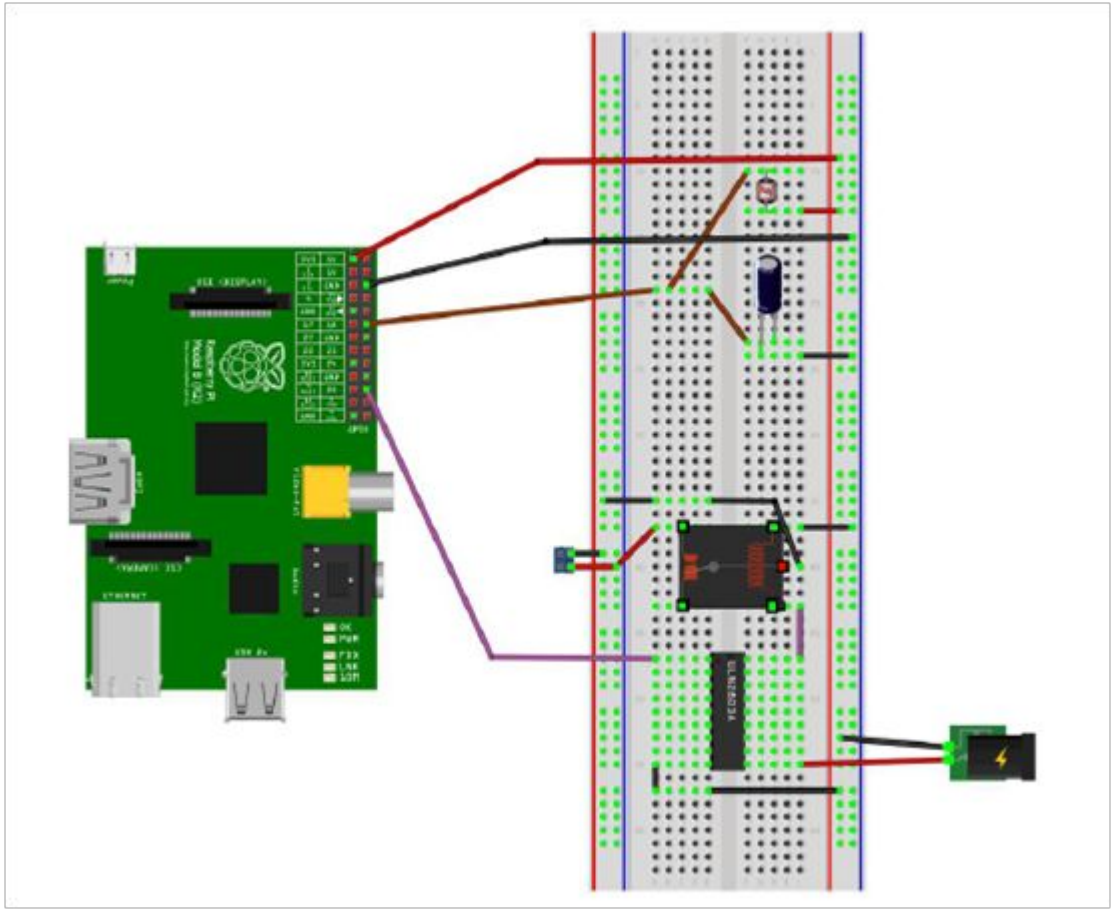
Management:

- Application Management - Django App Management
- Database Management - MySQL DB Management, Device Management - Raspberry Pi device Management.


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
WebApp
Application Server
Database Server
Management
Application Management
Database Management
Device Management
Services
Native Services
Web Services
Communication
Communication APIs
Communication Protocols
Security
Authentication
Authorization
Device
Sensors
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

In this step integration of the devices and components design such as minicomputer, LDR sensor, and relay switch actuator.



Step 10: Application Development

- The final step in the IoT design methodology.
- It is to develop the IoT application.
- The application has controls for the mode (auto-on or auto-off) and the light (on or off).

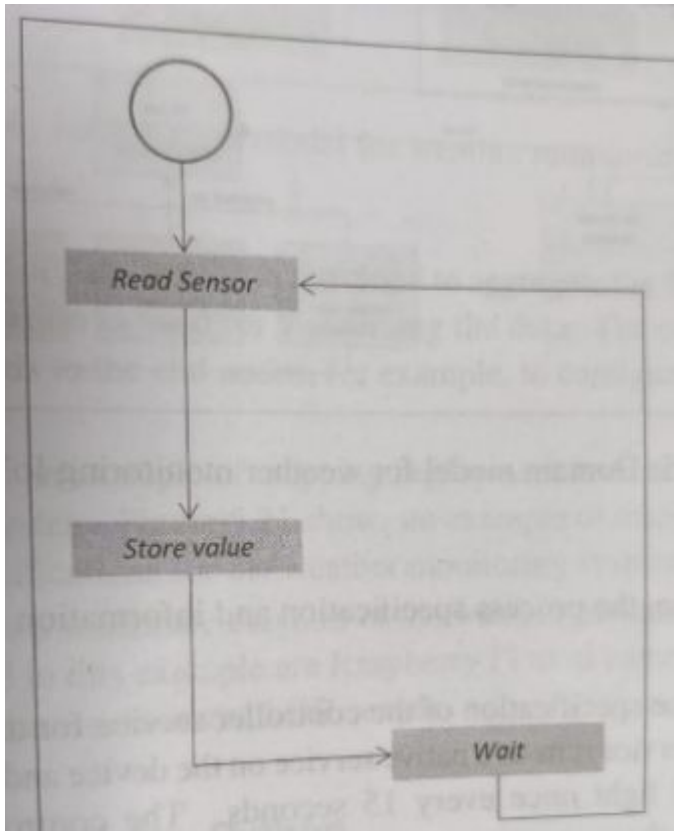
Design Methodology for Weather Monitoring System

Purpose:

- To collect data on environmental conditions such as temperature, pressure, humidity and light in an area using multiple end nodes.
- The end nodes send the data to the cloud where the data is aggregated and analyzed.

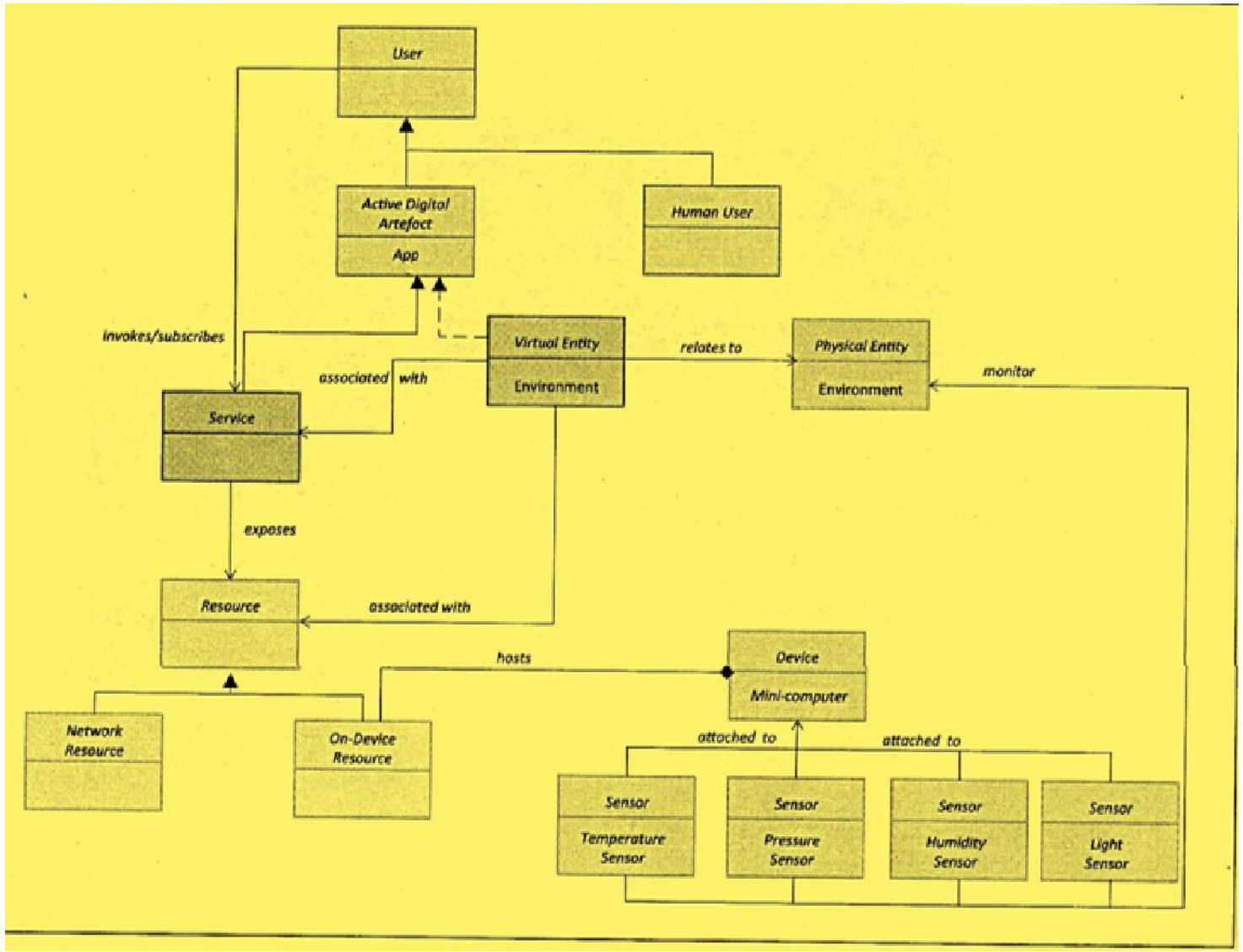
Process specification:

- Shows that the sensors are read after fixed intervals and the sensor measurements are stored.



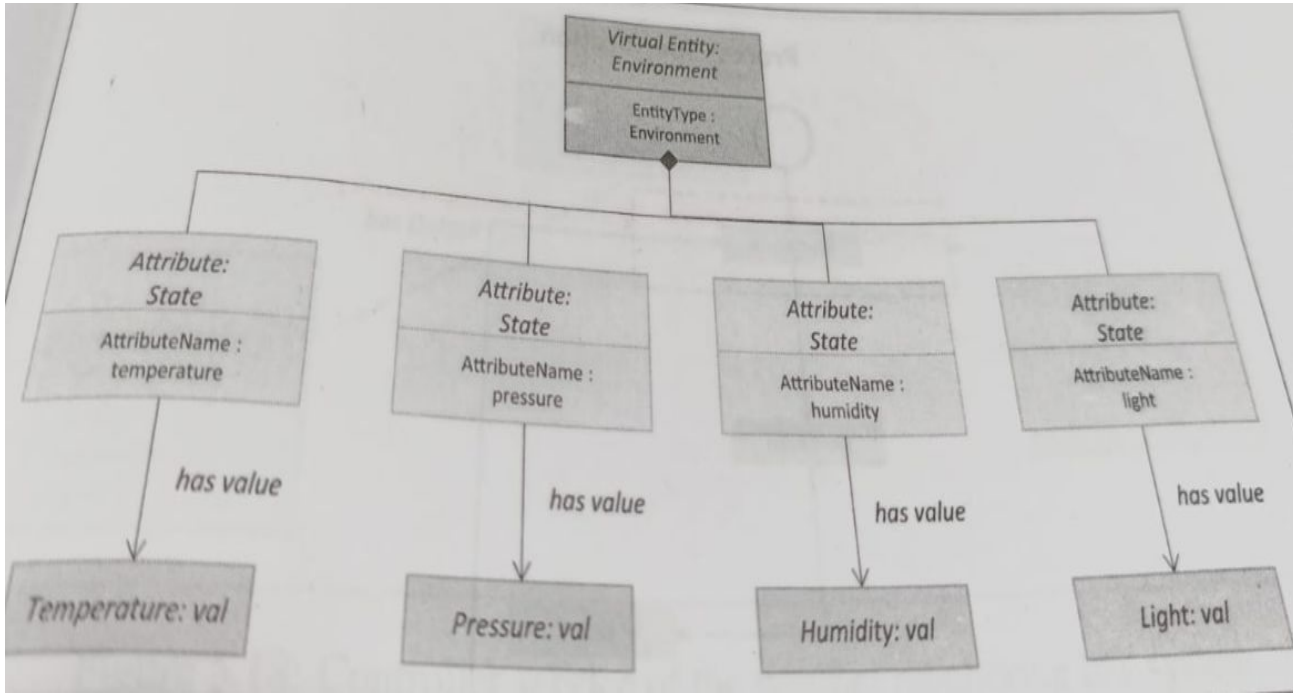
Domain Model:

- Physical entity is the environment which is being monitored
- Virtual entity for the environment.
- Devices include temperature sensor, pressure sensor, humidity sensor, light sensor and single-board mini computer
- Resources are software components which can be either on-device or network-resources.
- Services include the controller service that monitors the temperature , pressure, humidity, and light and sends reading to the cloud.

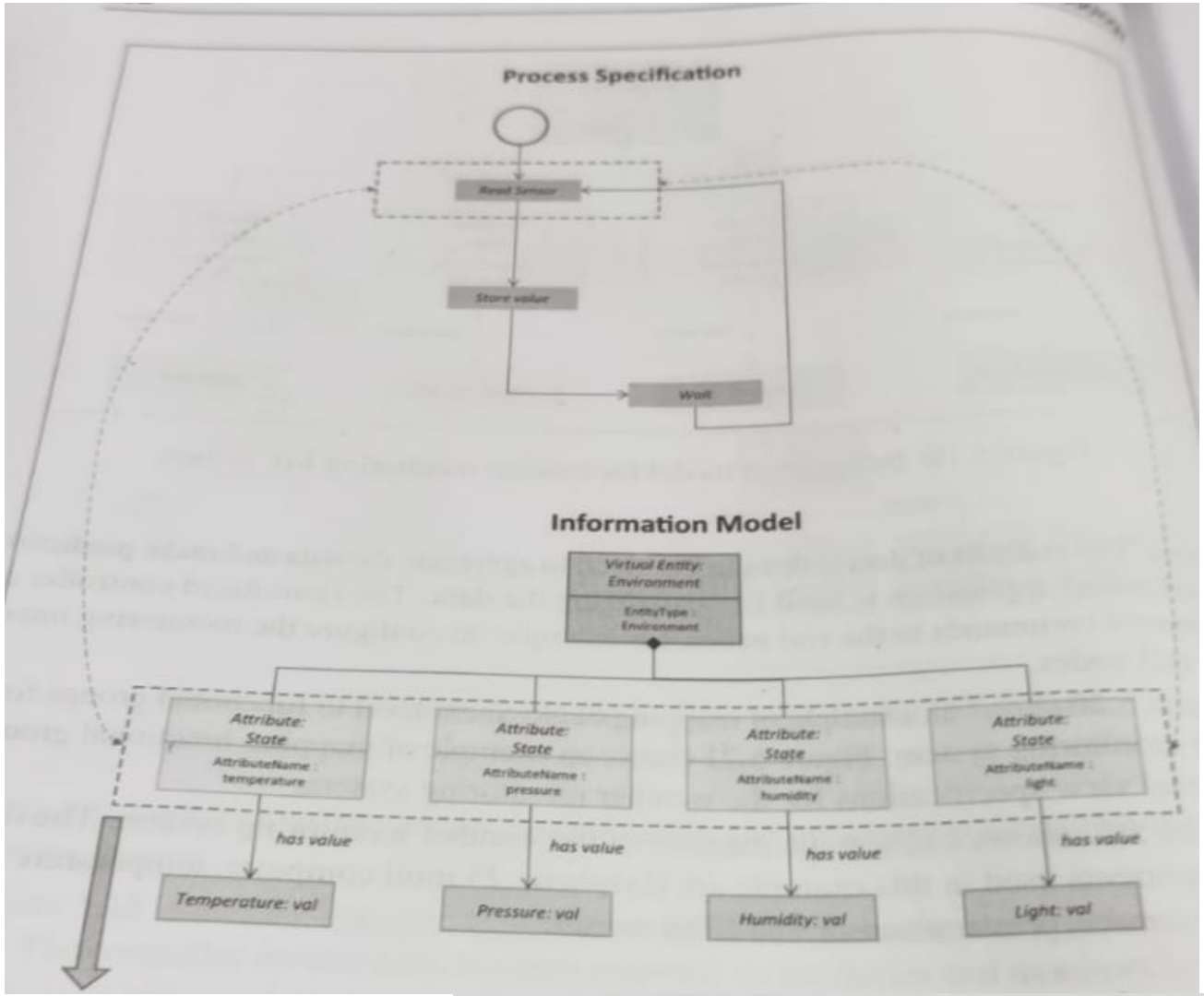

User
Active Digital Artefact
App
Human User
invokes/subscribes
Virtual Entity
Environment
relates to
Physical Entity
Environment
monitor
associated with
Service
exposes
Resource
associated with
hosts
Device
Mini-computer
Network Resource
On-Device Resource
attached to
attached to
Sensor
Temperature Sensor
Sensor
Pressure Sensor
Sensor
Humidity Sensor
Sensor
Light Sensor

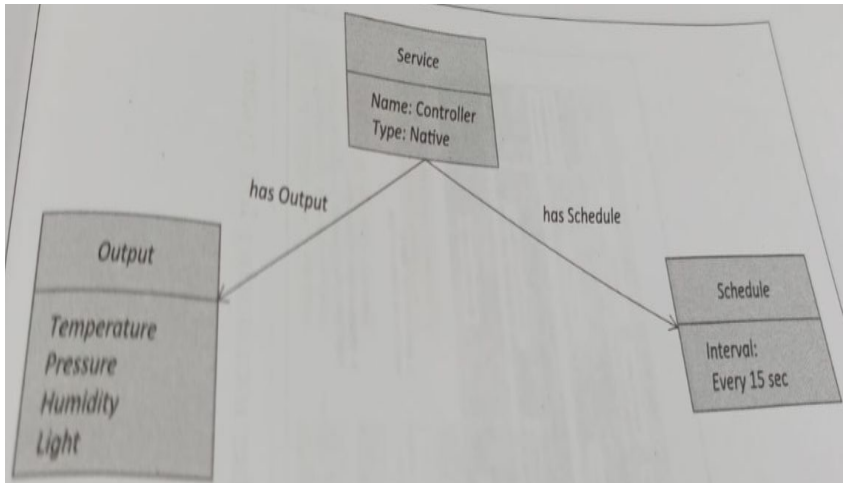
Information Model:

- There is one virtual entity the enivironment to be sensed
- The virtual entity has attributes as temperature, humidity, pressure, and light



- Service Model:

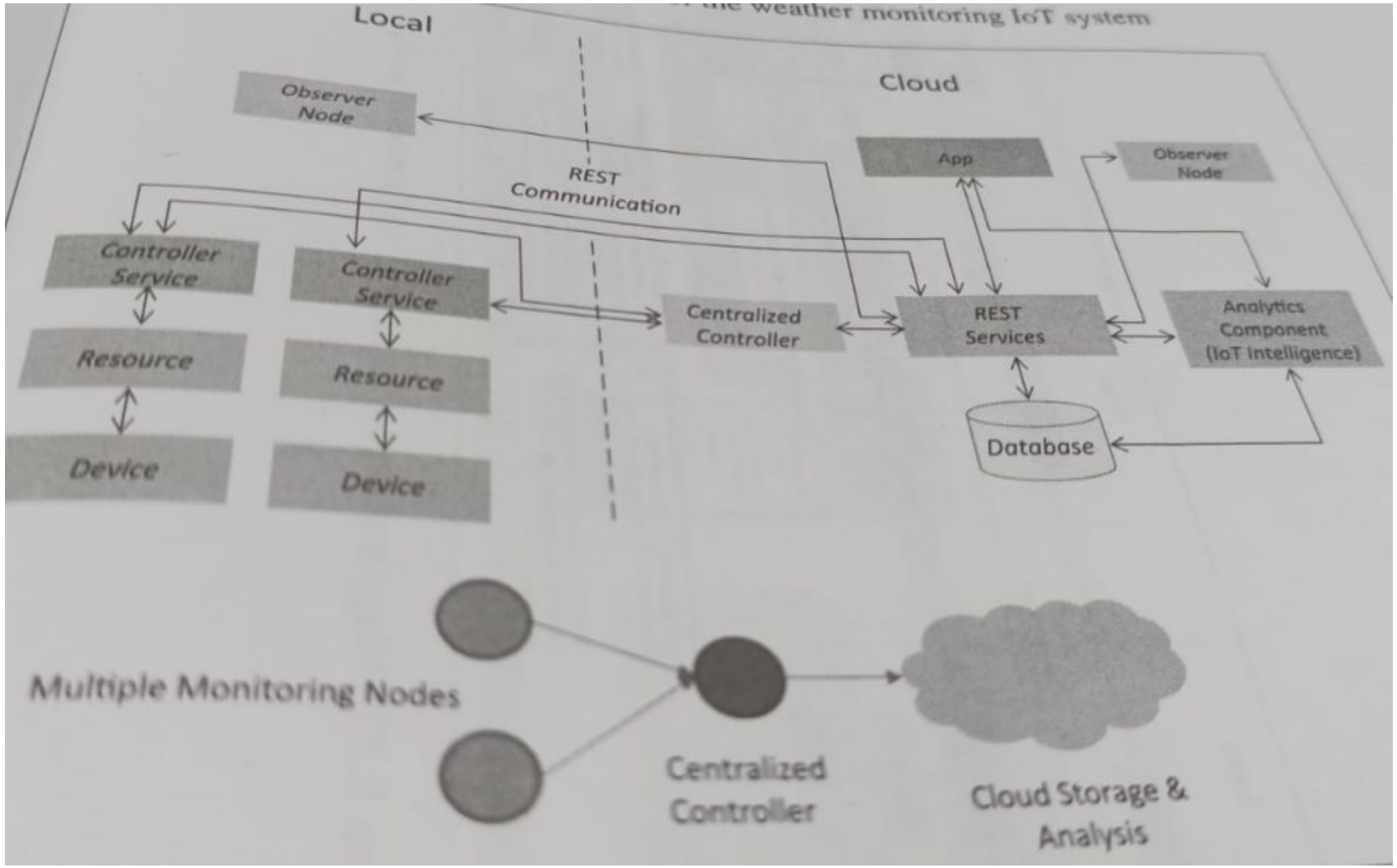




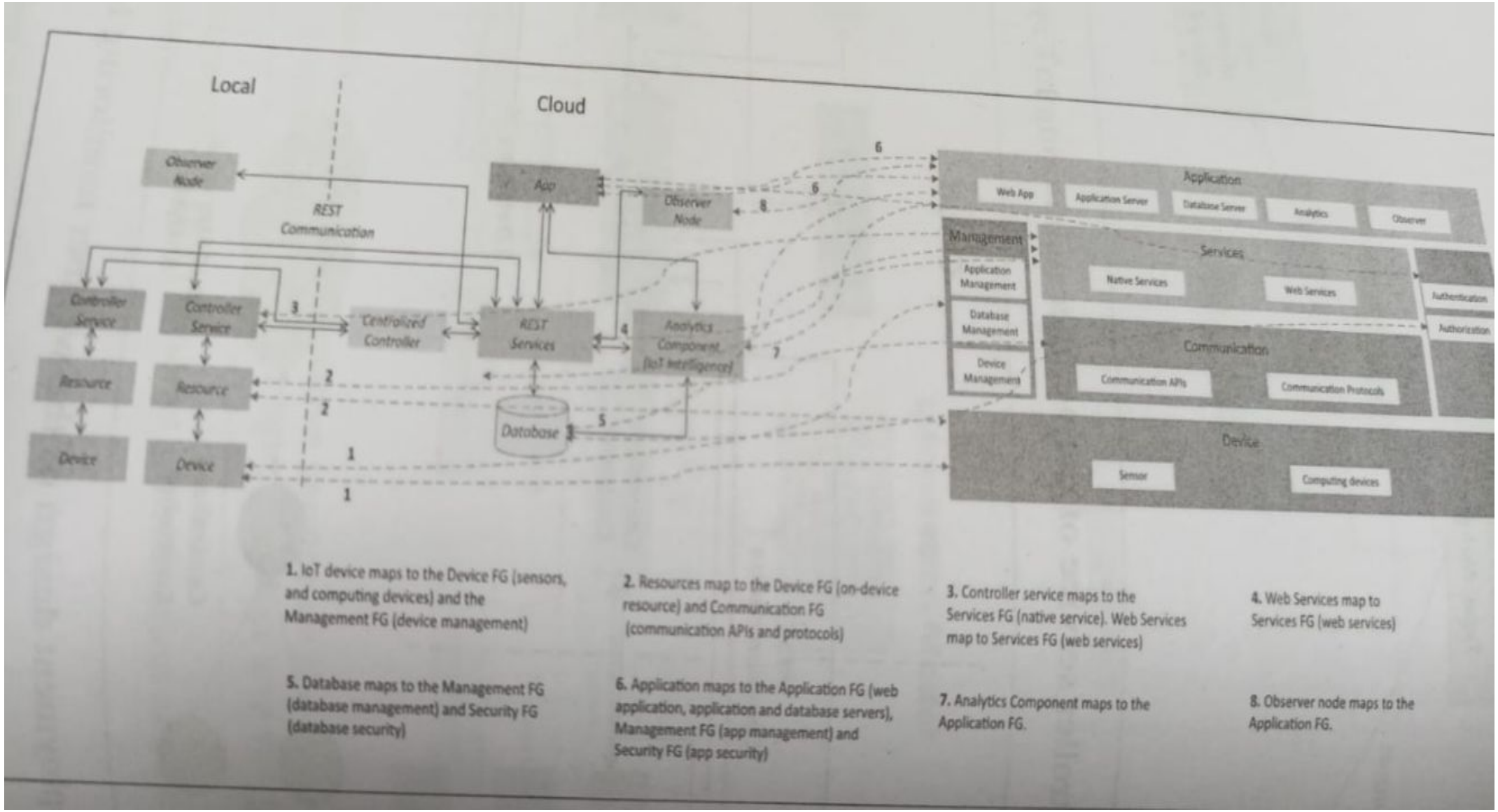
Controller Services

Controller Service: Runs as a native service on the device. Gets the current temperature, pressure, humidity and light readings and sends to the cloud database.

Deployement Level



Functional View Specification



1. IoT device maps to the Device FG (sensors, and 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)

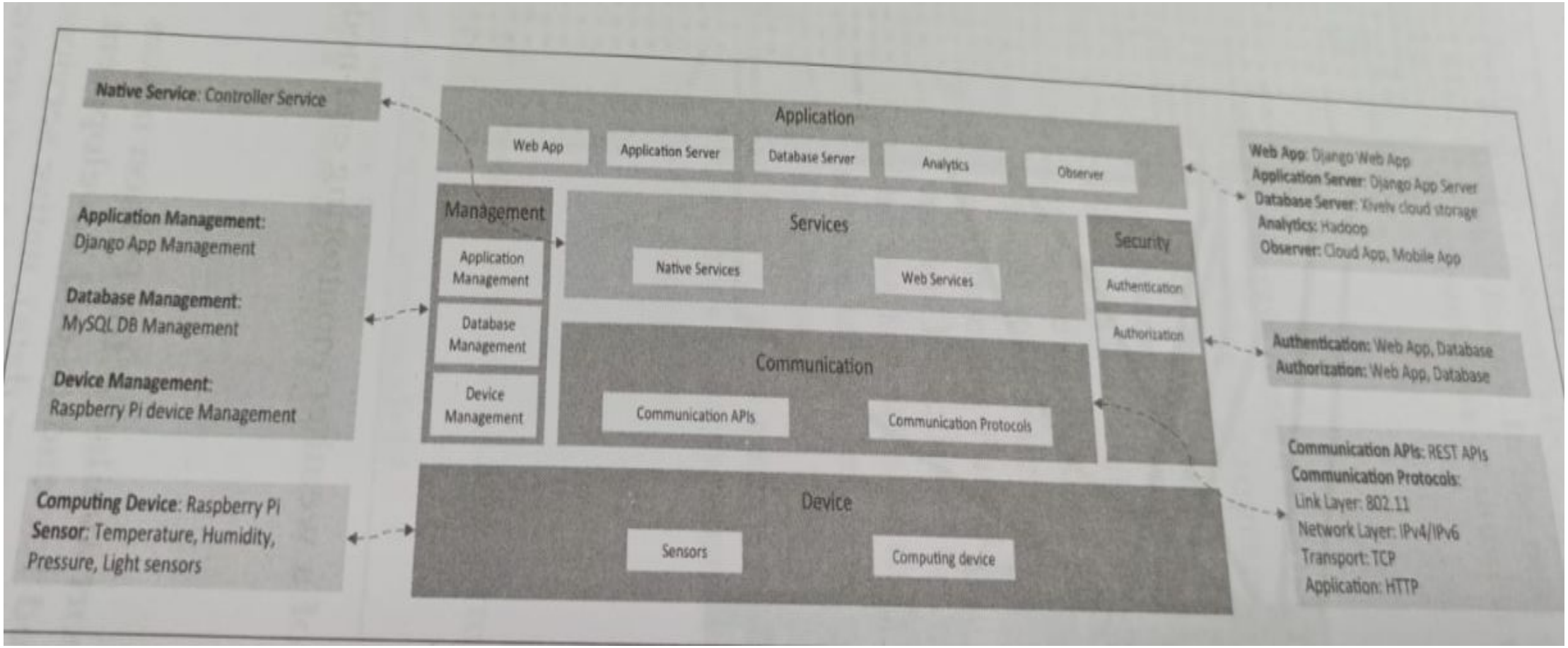
5. Database maps to the Management FG (database management) and Security FG (database security)

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

7. Analytics Component maps to the Application FG.

8. Observer node maps to the Application FG.

Operational View



Device and componenet intregration