IoT Design Methodology

Introduction

- Designing IoT systems can be a complex and challenging task as these systems involve interactions between various components such as IoT devices and network resources, web services, analytics components, application and database servers.

- IoT system designers often tend to design IoT systems keeping specific products/services in mind.

- So that designs are tied to specific product/service choices made. But it make updating the system design to add new features or replacing a particular product/service choice for a component becomes very complex, and in many cases may require complete re-design of the system.

Introduction

- Here we discuss a generic design methodology for IoT system design which is independent of specific product, service or programming language.

- IoT systems designed with the proposed methodology have reduced design, testing and maintenance time, better interoperability and reduced complexity.'

IoT Design Methodology

It includes:

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

Purpose & RequirementsSpecification

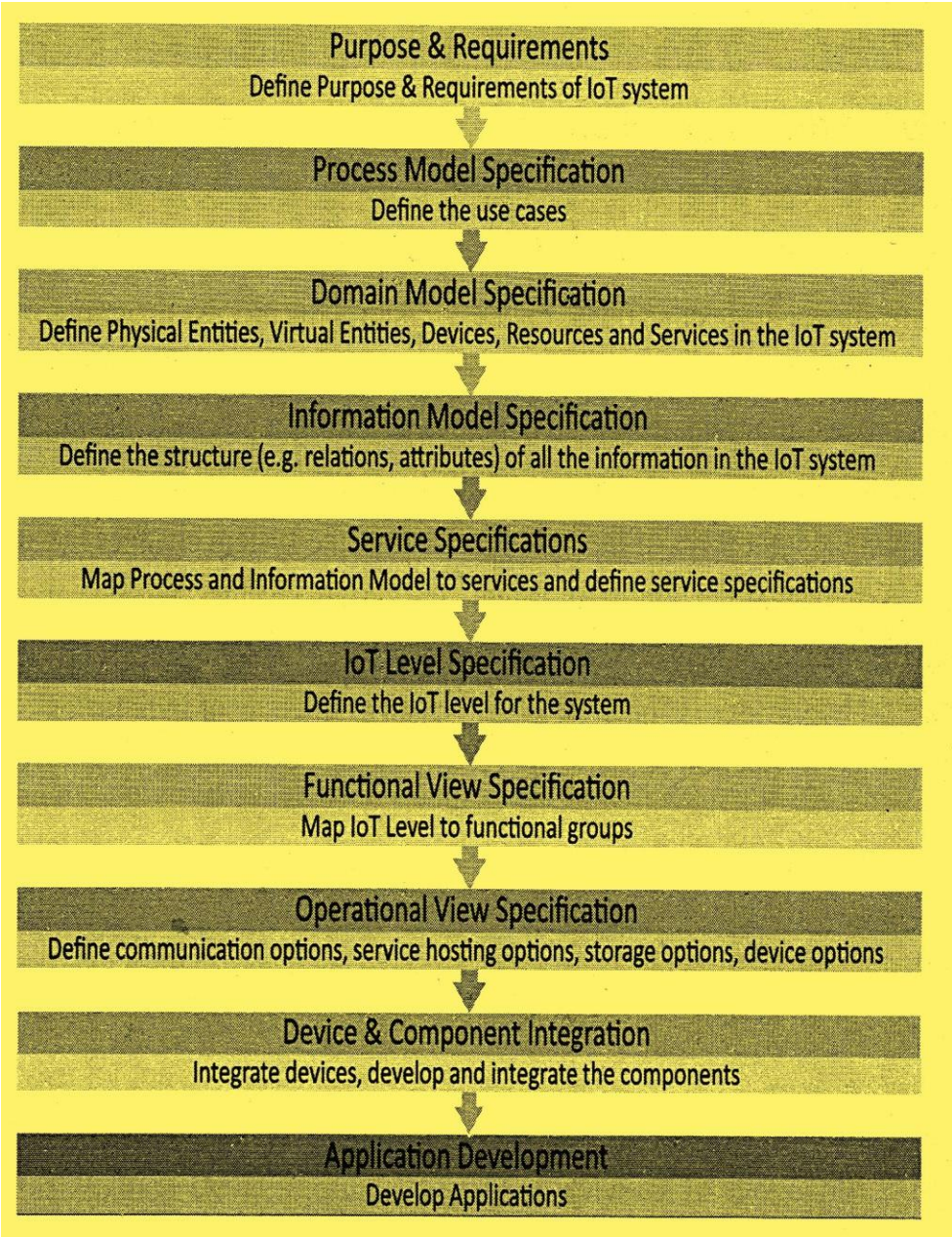
The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements are captured.

Purpose & Requirements Specification

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.



IoT DesignMethodology

It includes:

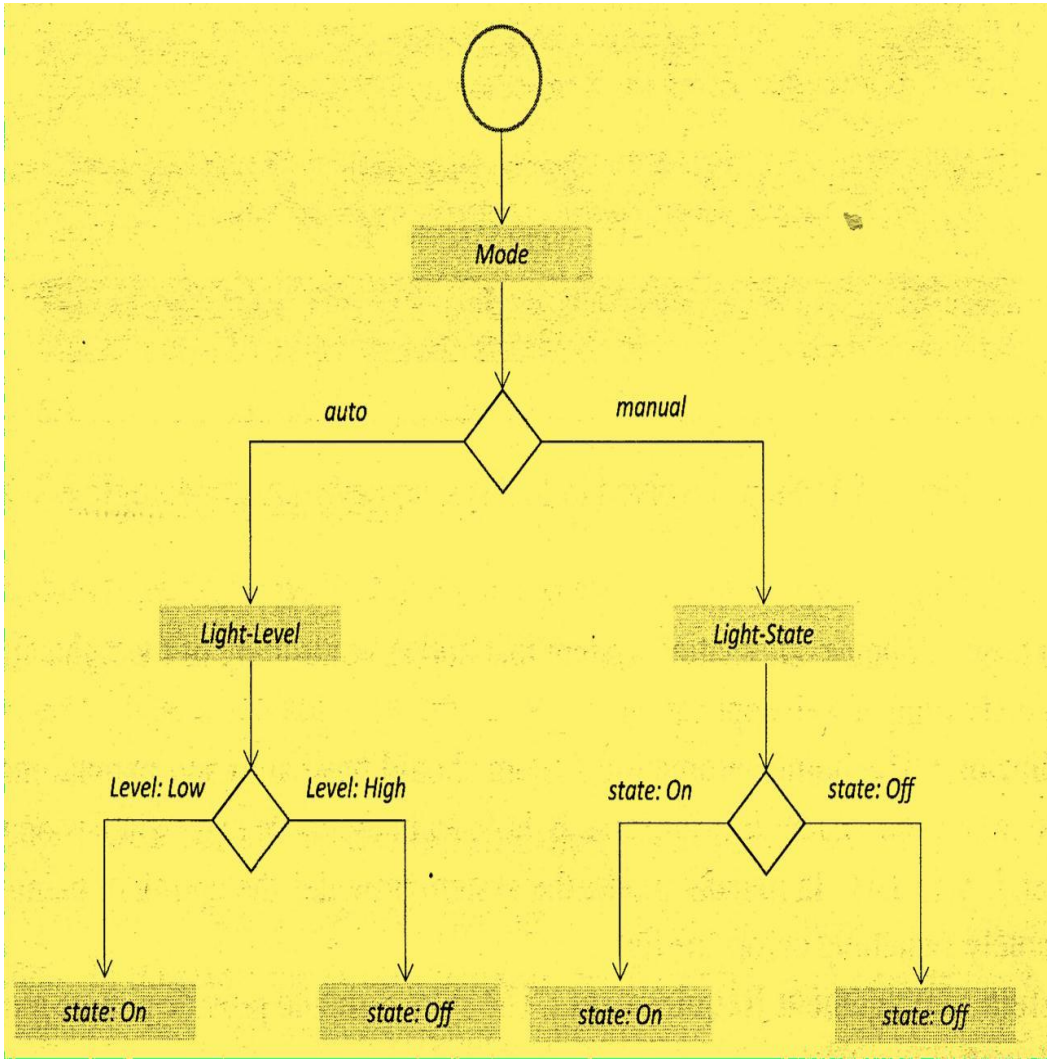
- Purpose & Requirements Specification
- Process Specification
- Domain Model Specification
- Information Model Specification
- Service Specification
- IoT Level Specifications
- Functional view Specification
- Operational View Specification
- Device & component Integration
- Application Development

2 Process Specification

- 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.

Process Specification(second step)



In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

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.

Domain Model Specification

The entities, objects and concepts defined in the domain model include:

Physical Entity : Physical Entity is a discrete and identifiable entity in the physical environment (e.g. a room, a light, an appliance, a car, etc.).

Virtual Entity : Virtual Entity is a representation of the Physical Entity in the digital world.

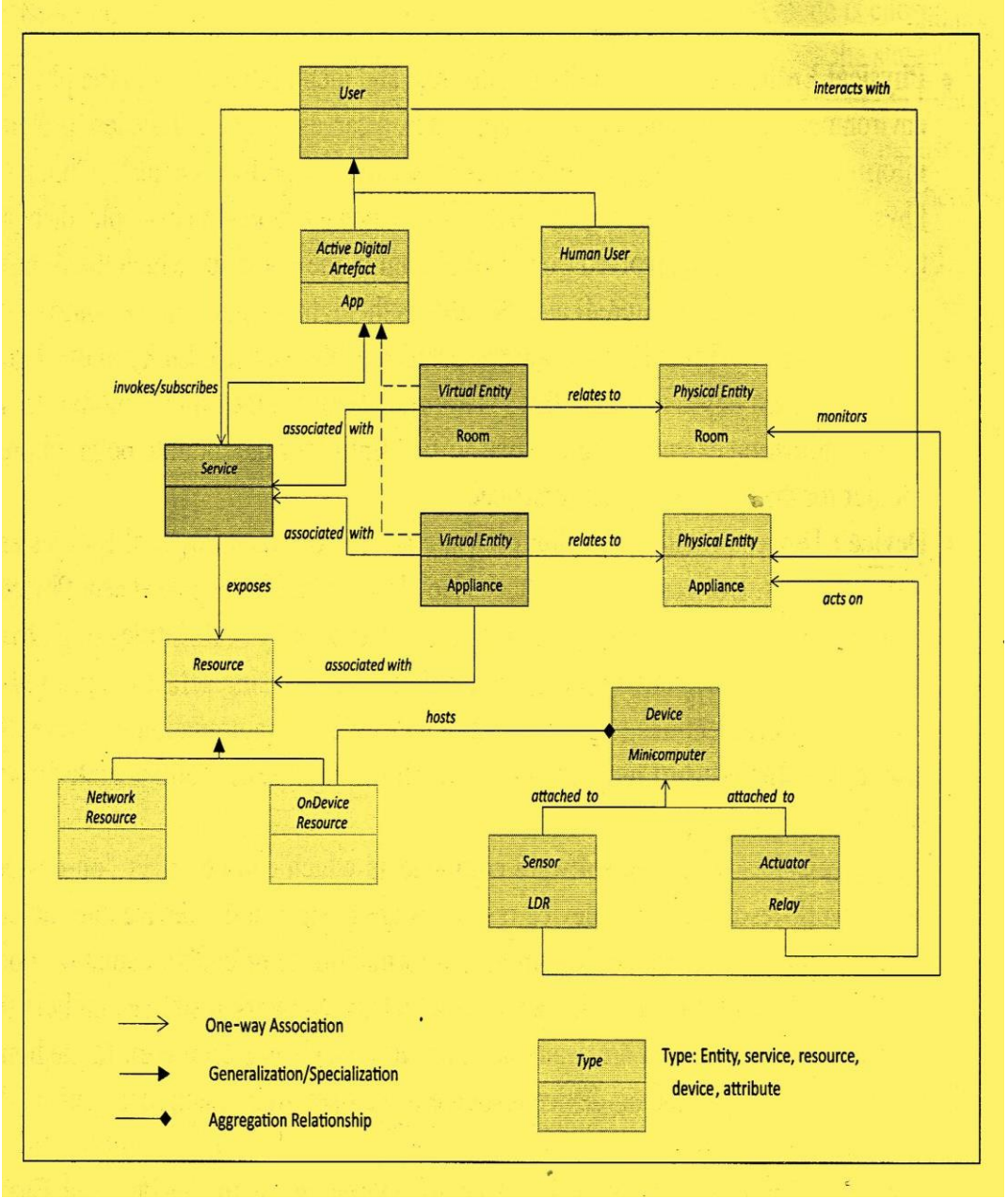
Device :provides a medium for interactions between Physical Entities and Virtual Entities. Devices are either attached to Physical Entities or placed near Physical Entities.

Domain Model Specification

Resource : Resources are software components which 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.

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.

Domain Model Specification



IoT DesignMethodology

It includes:

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

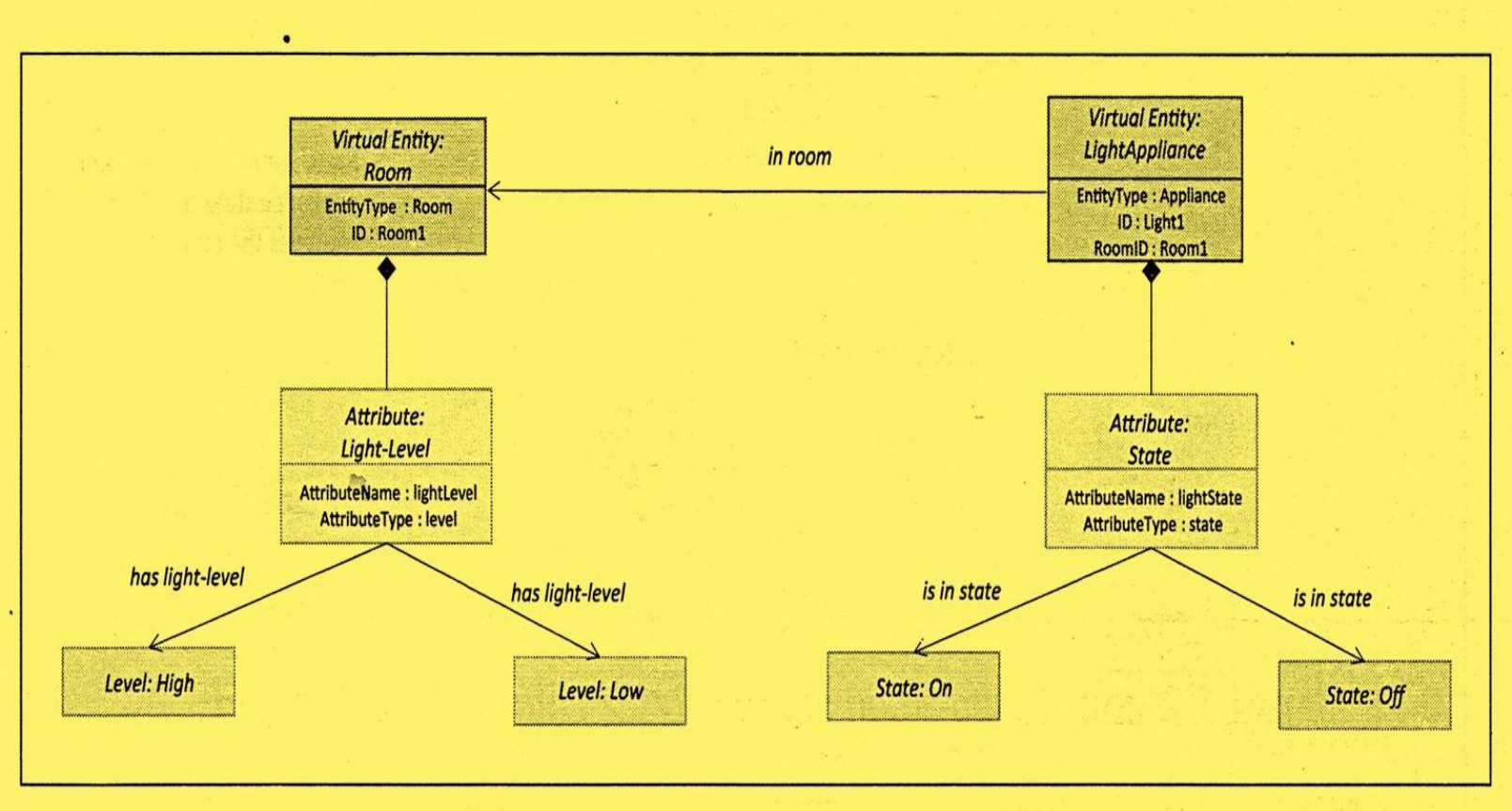
4 Information Model Specification

- 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

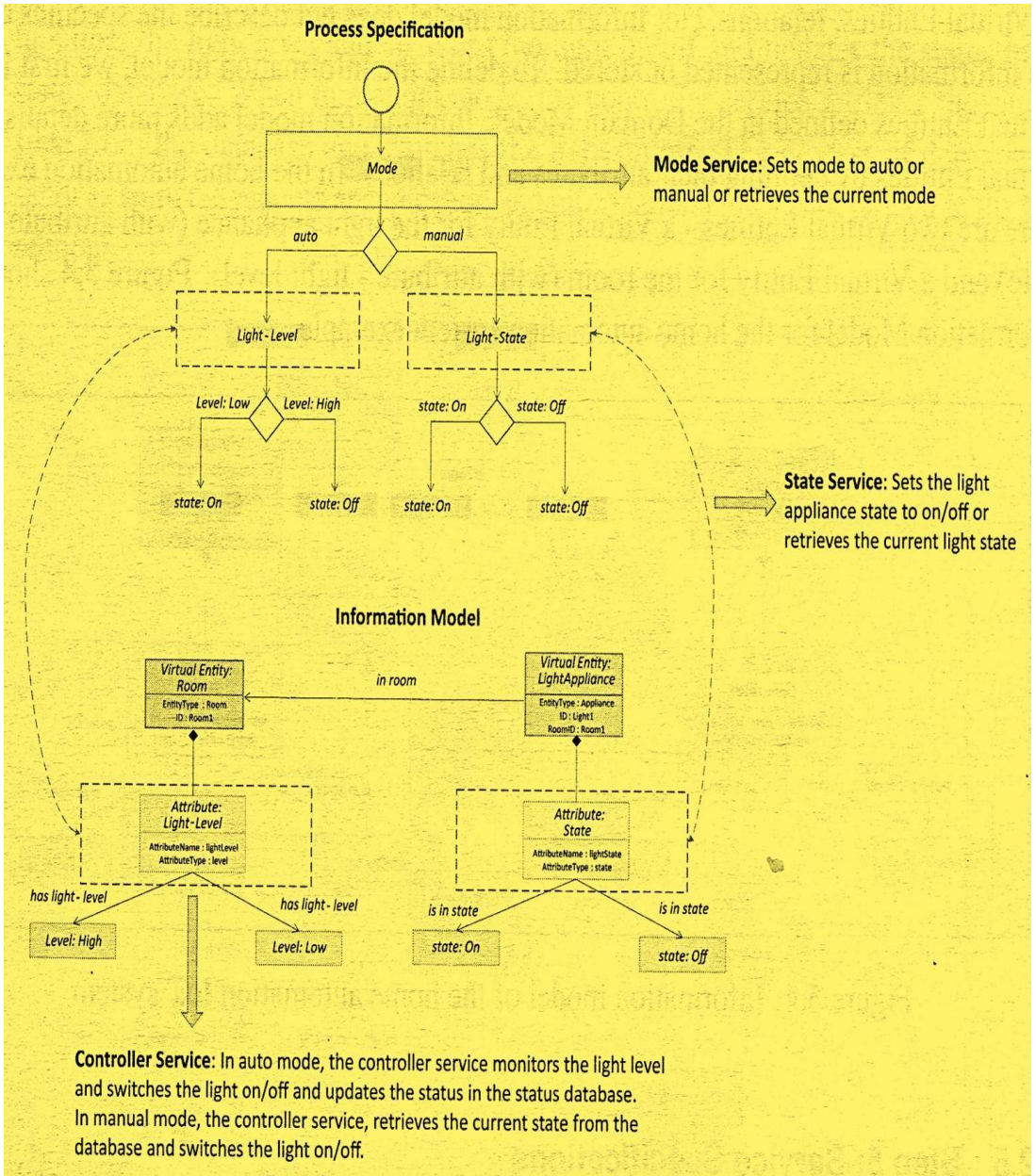
Information Model Specification



5 Service Specification

- 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.

Service Specification



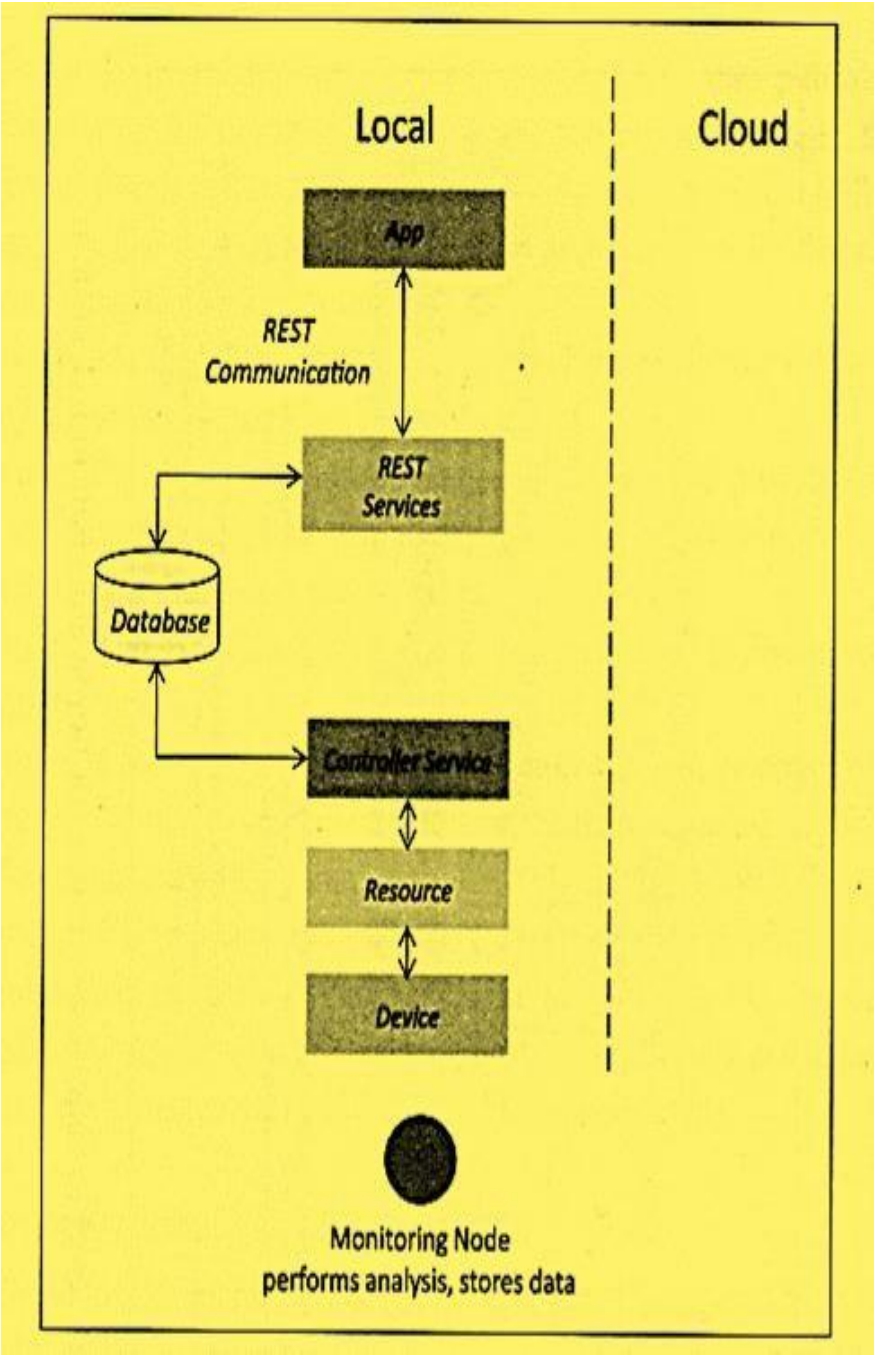
From the process specification and information model, we identify the states and attributes.

For each state and attribute we define a service.

These services either change the state or attribute values or retrieve the current values.

6 IoT Level Specifications

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



IoT Design Methodology

- Functional view Specification
- Operational View Specification

IoT DesignMethodology

It includes:

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

7 Functional view Specification (seventh step)

- 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.

Functional viewSpecification

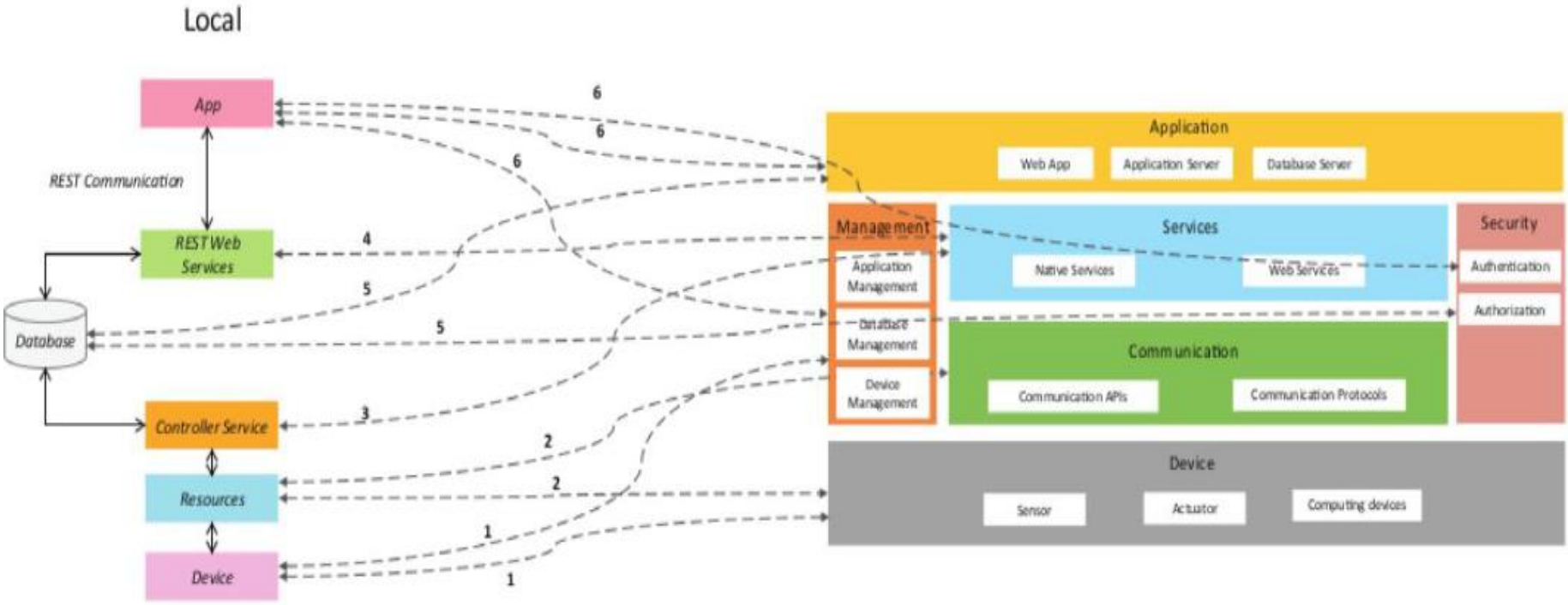
- The Functional Groups (FG) included in a Functional View include:

- **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 WebSocket) that are used by the services and applications to exchange data over the network.

Functional viewSpecification

- 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 IoT 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.



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)

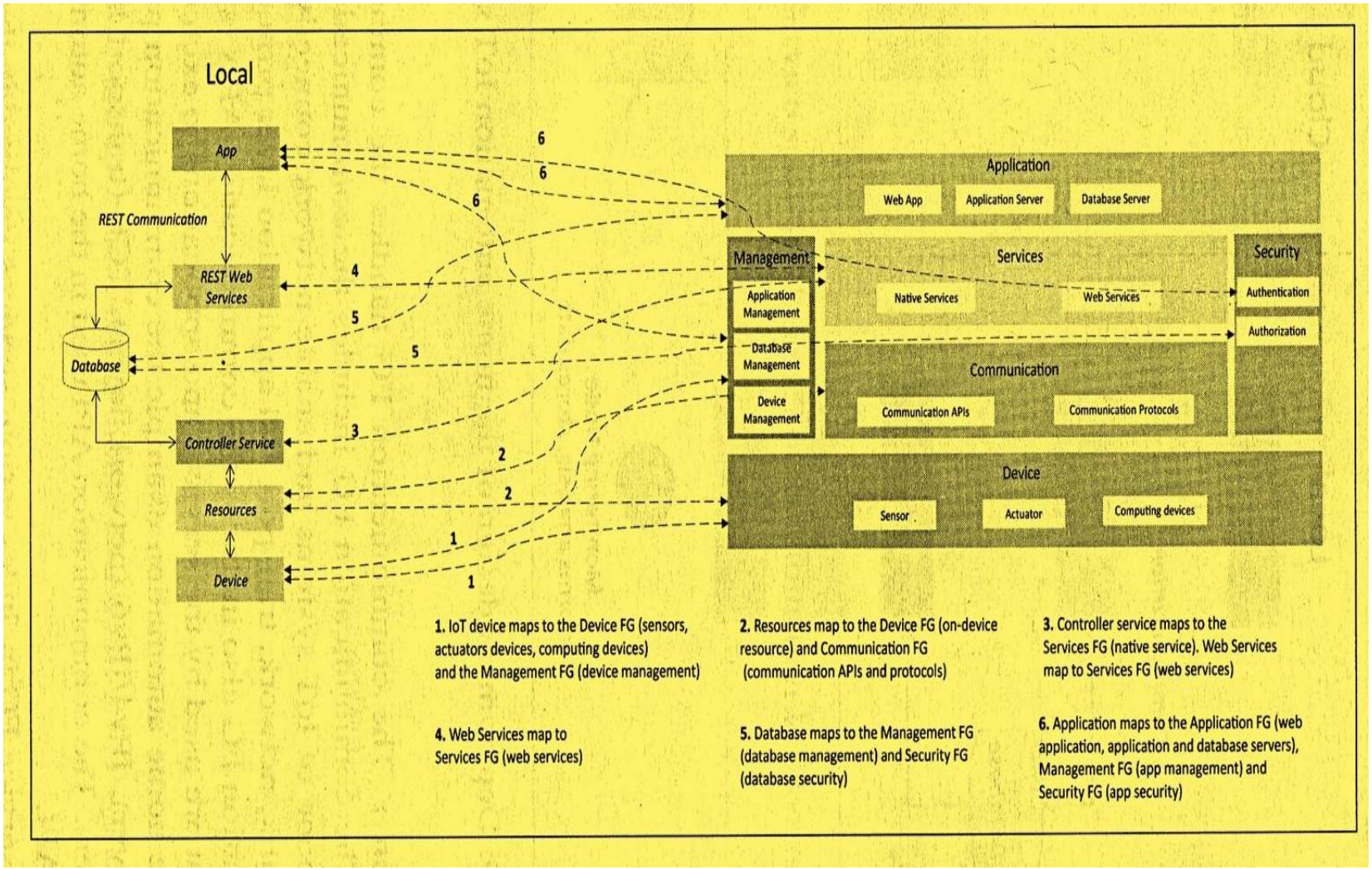
3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

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)

Functional view Specification



8 Operational View Specification

- In this step, various options pertaining to the loT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc.
- Operational View specifications for the home automation example are as follows:
- Devices: Computing device (Raspberry Pi), light dependent resistor (sensor), relay
- switch (actuator).
- Communication APls: REST APis
- Communication Protocol s: Link Layer - 802.11, Network Layer - 1Pv4/1Pv6,
- Transport TCP, Application - HTTP.

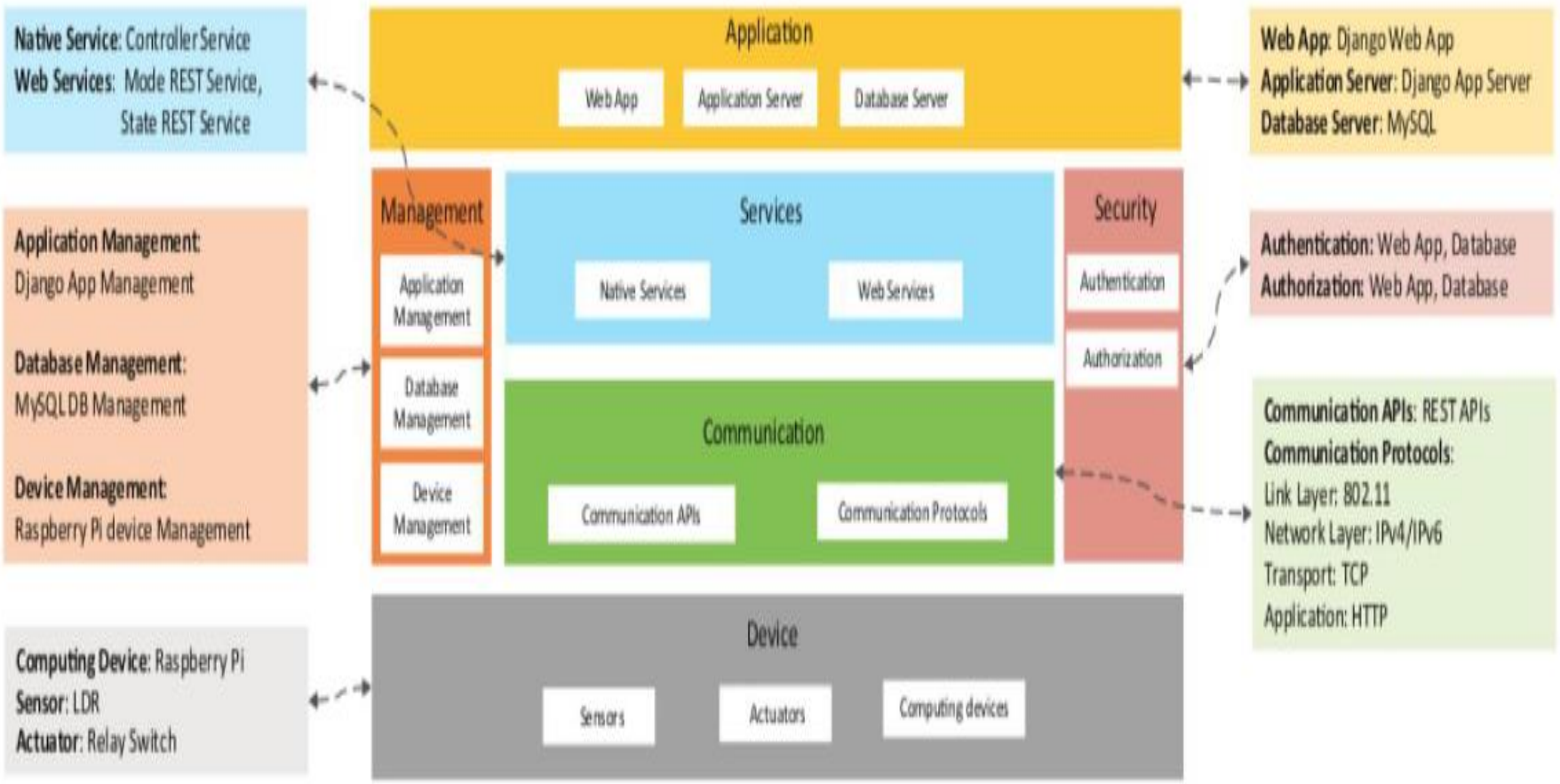
Operational View Specification

- Operational View specifications for the home automation example are as follows:
- Services:
  - Controller Service - Hosted on device, implemented in Python and run as a native service.
- Mode service - REST-fol web service, hosted on device, implemented with Django-REST Framework.
- 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.

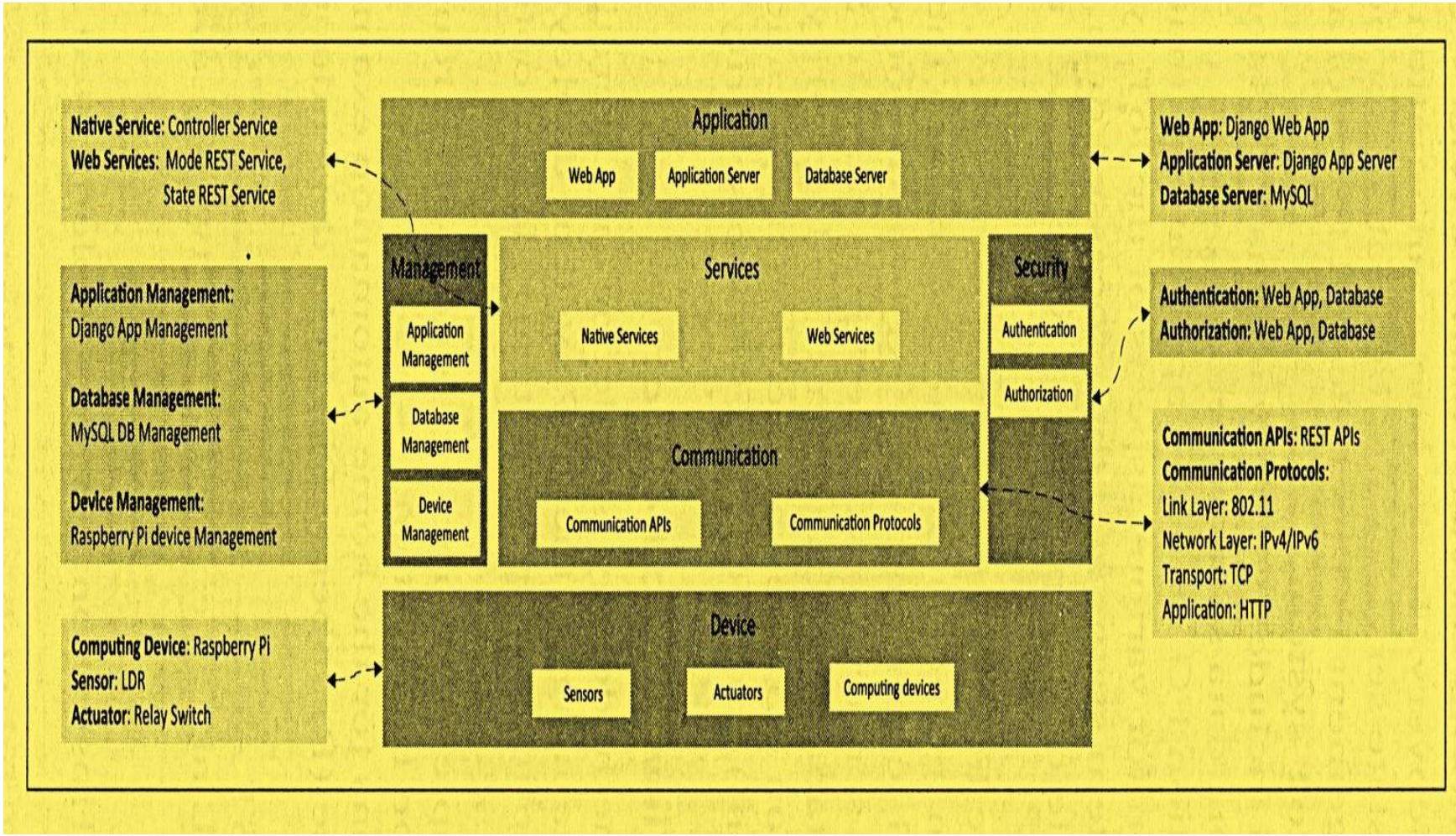
Operational View Specification

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

Operational View Specification



Operational View Specification

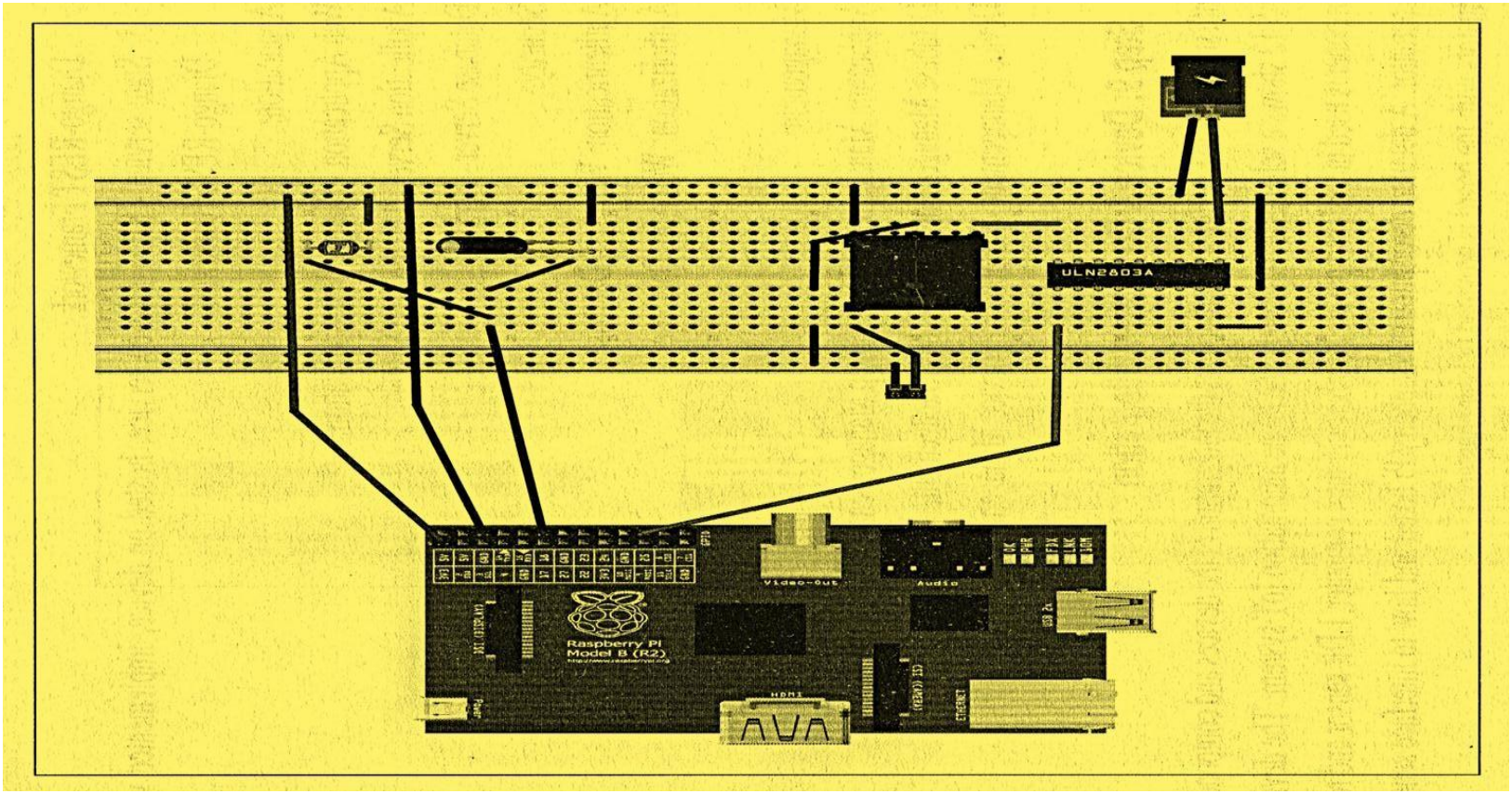


IoT Platforms DesignMethodology

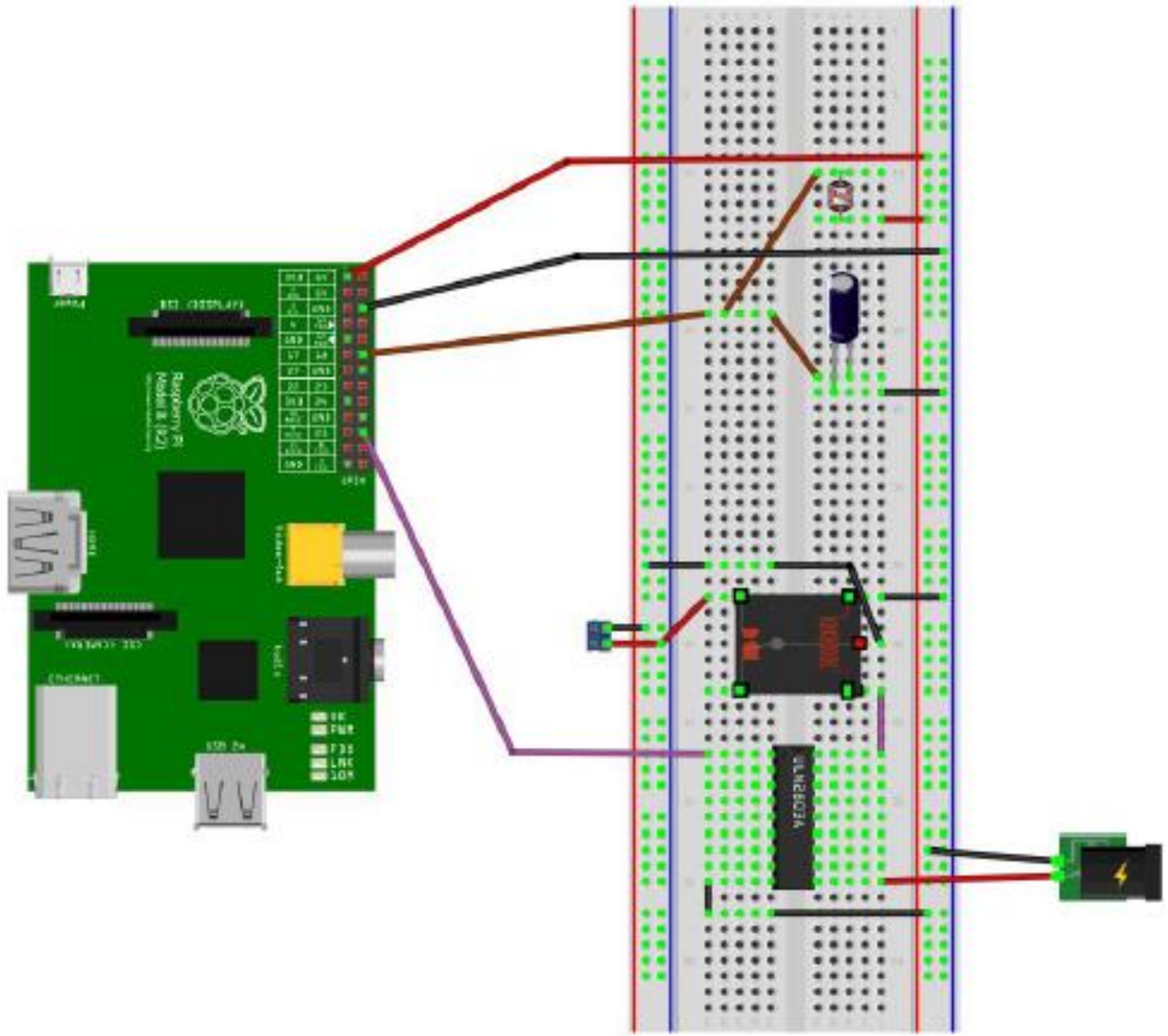
It includes:

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

Device & component Integration



The devices and components used in this example are Raspberry Pi mini computer, LDR sensor and relay switch actuator.

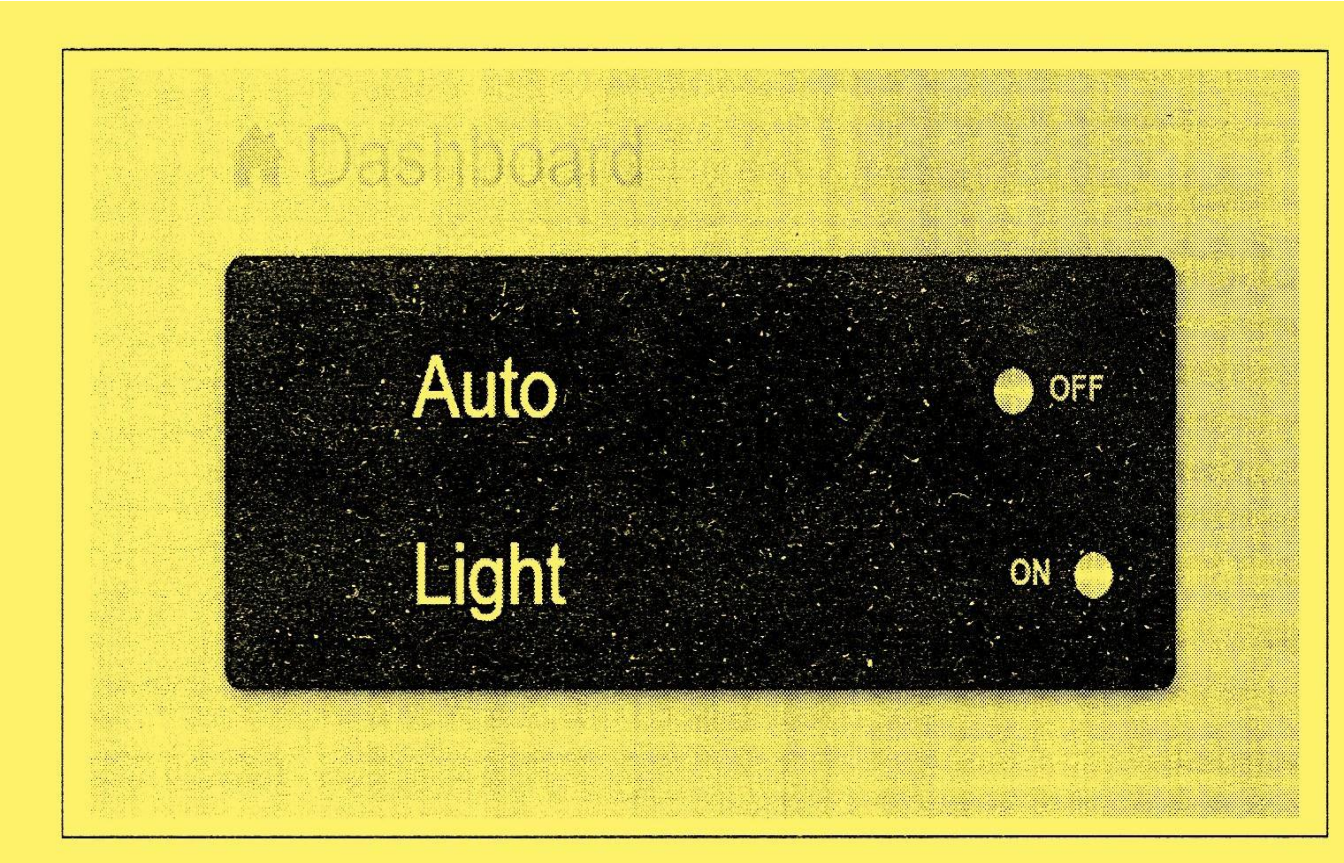


Application Development

- The application has controls for the mode (auto on or auto off) and the light (on or off).
- In the auto mode, the IoT system controls the light appliance automatically based on the lighting conditions in the room.
- When auto mode is enabled the light control in the application is disabled and it reflects the current state of the light.
- When the auto mode is disabled, the light control is enabled and it is used for manually controlling the light.

- **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.

Application Development



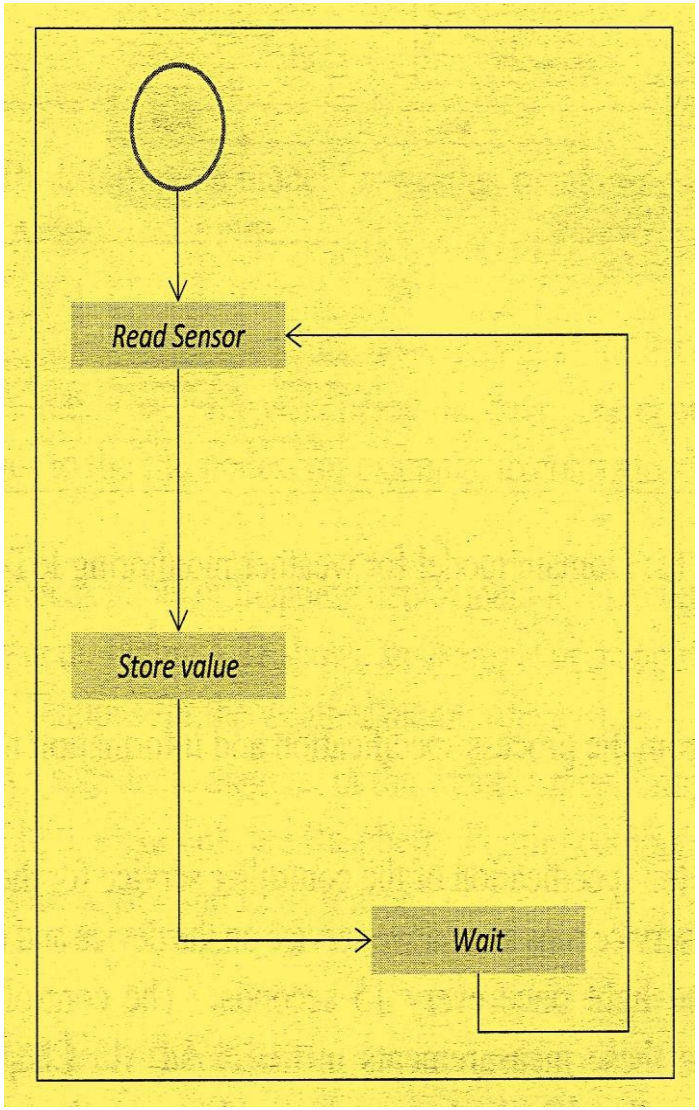
Case Study Weather MonitoringSystem

- The purpose of the weather monitoring system is 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.

Case Study Weather Monitoring System

- Figure shows the process specification for the weather monitoring system.
- The process specification shows that the sensors are read after fixed intervals and the sensor measurements are stored.



Case Study Weather MonitoringSystem

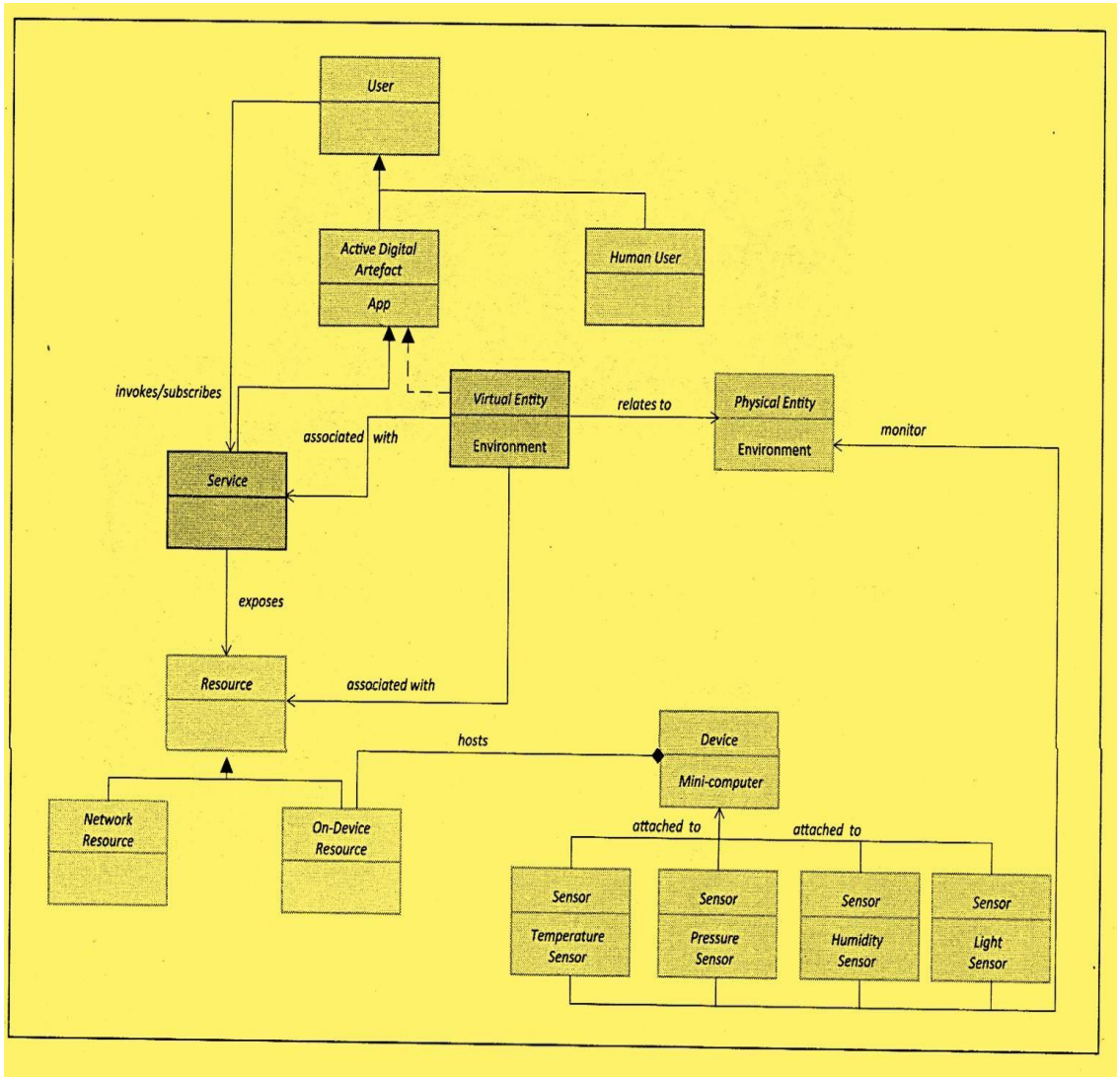
- In this domain model the physical entity is the environment which is being monitored .

•There is a 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.

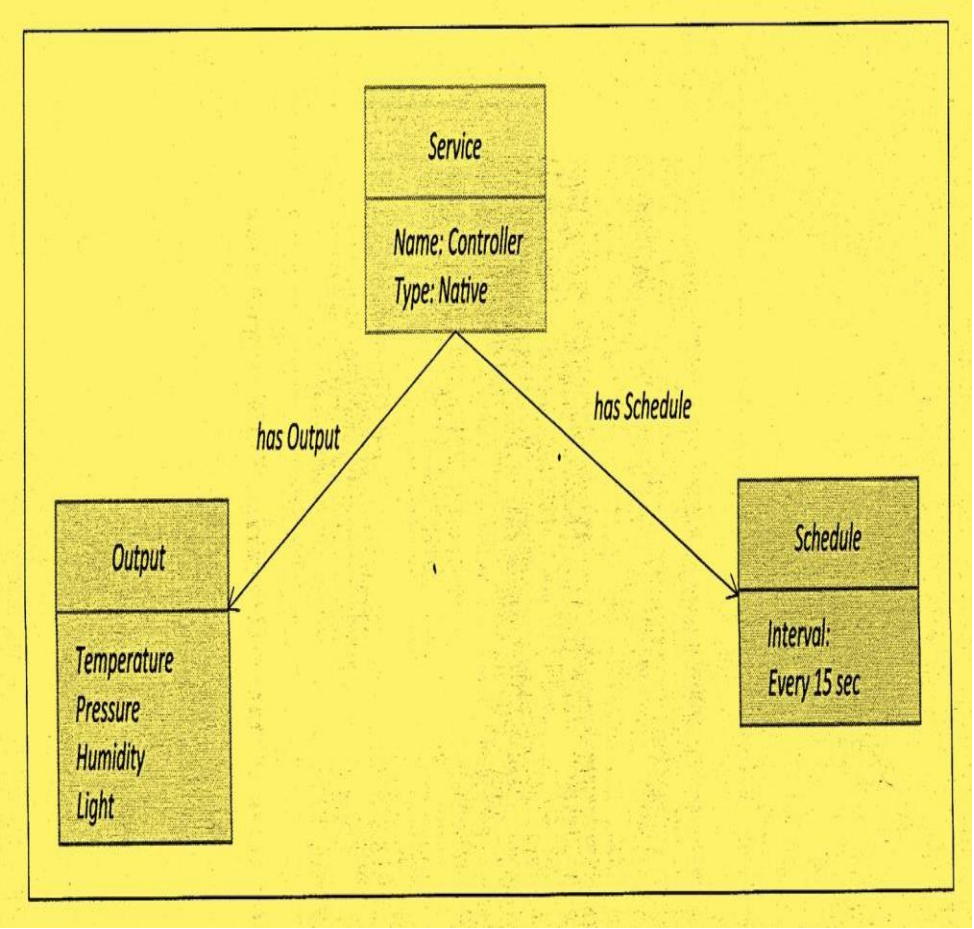
- Services include the controller service that monitors the temperature , pressure deriving the services from the process specification and information model for the weather monitoring system, humidity and light and sends the readings to the deriving the services from the process specification and information model for the weather monitoring system.

Case Study Weather Monitoring System (domain model)



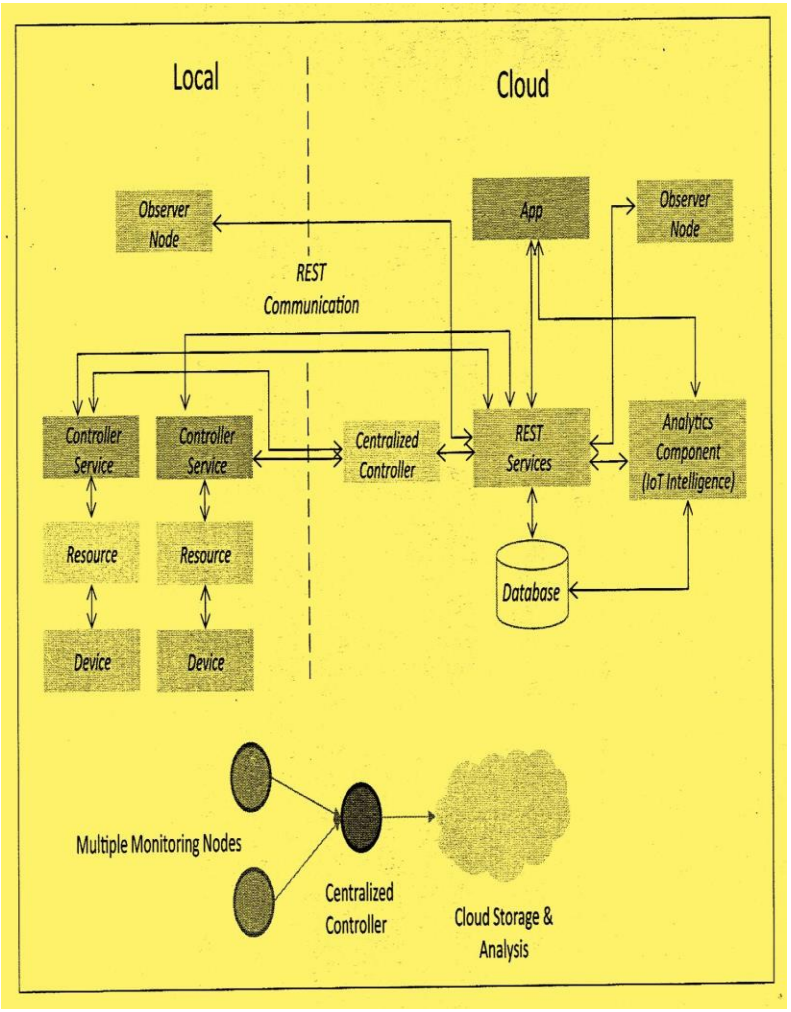
Case Study Weather Monitoring System (controller service)

- The controller service runs as a native service on the device and monitors temperature, pressure, humidity and light once every 15 seconds.
- The controller service calls the REST service to store these measurements in the cloud.

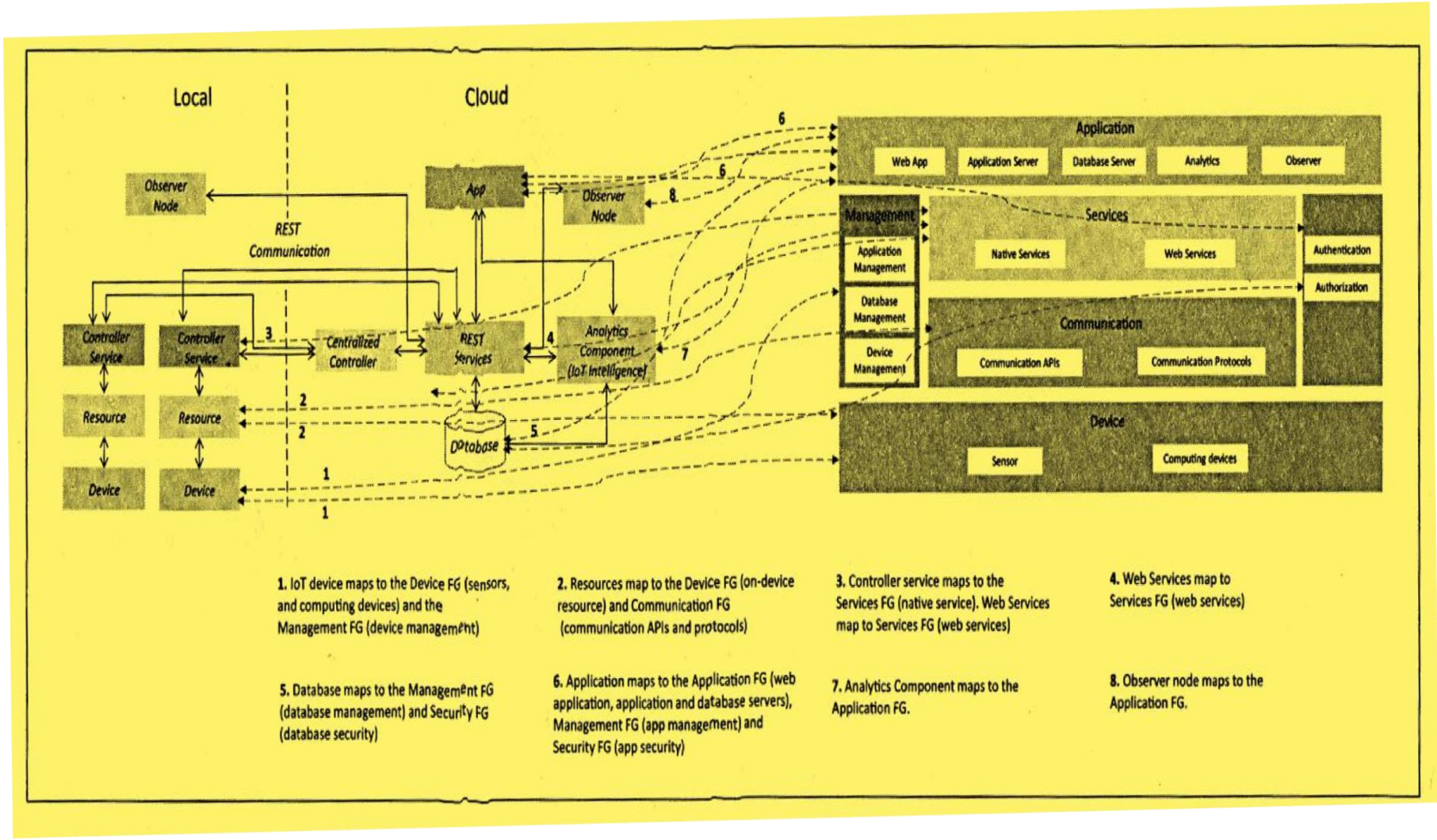


Case Study Weather Monitoring System
(deployment design for the system)

- The system consists of multiple nodes placed in different locations for monitoring temperature, humidity and pressure in an area.
- The end nodes are equipped with various sensors .
- The end nodes send the data to the cloud and the data is stored in a cloud database.
- The analysis of data is done in the cloud to aggregate the data and make predictions

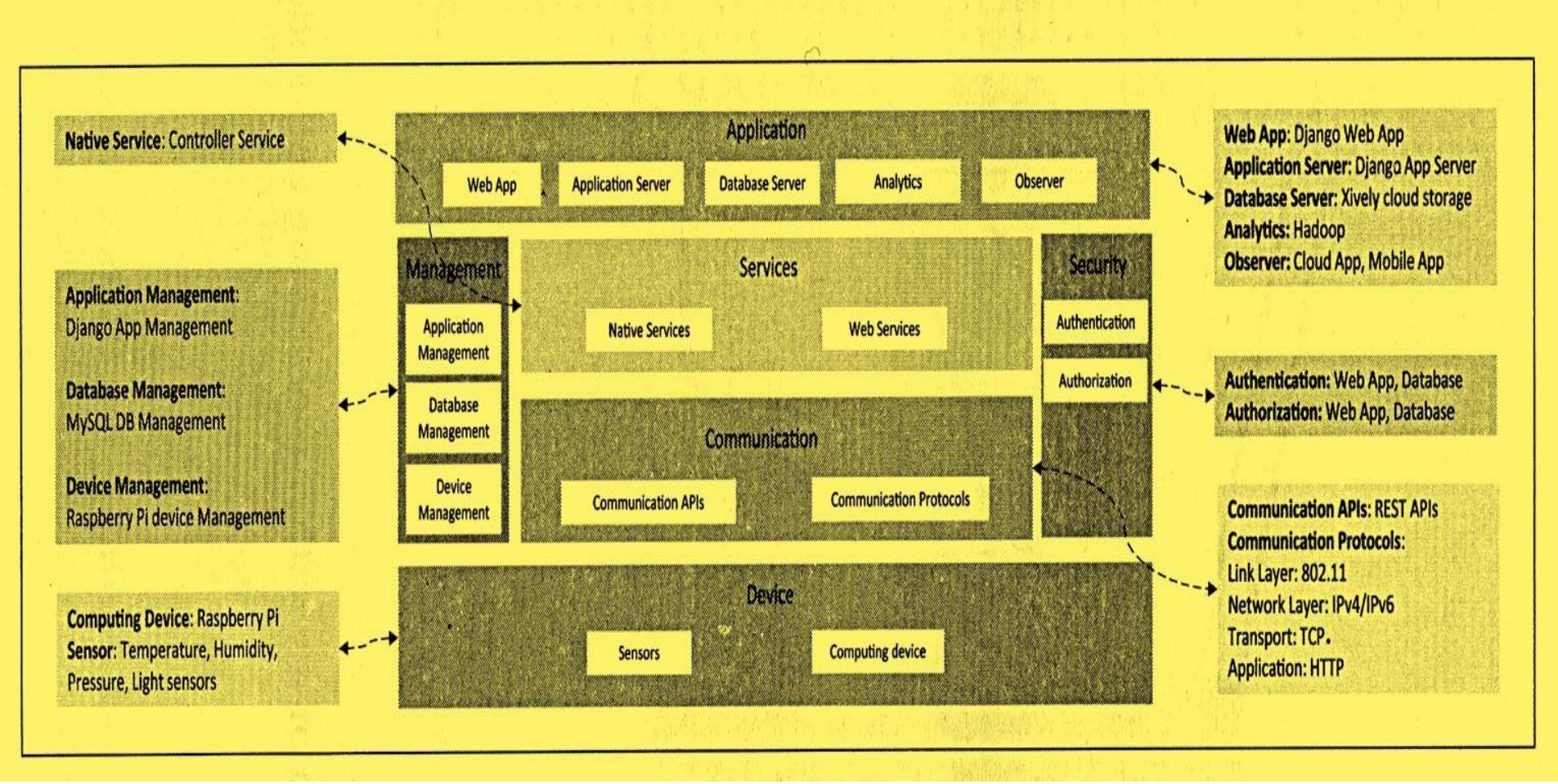


Case Study Weather Monitoring System



Mapping deployment level to functional groups for the weather monitoring system.

Case Study Weather Monitoring System



Mapping functional Groups to operational view specifications for the weather monitoring system.

Case Study Weather Monitoring System (controller service)

- The schematic diagram of the weather monitoring system.
- The devices and components used in this example are Raspberry Pi mini computer, temperature sensor, humidity sensor, pressure sensor and LDR sensor.