

MODULE-3

IoT Platforms Design Methodology: Introduction, IoT Design Methodology, Case Study on IoT System for Weather Monitoring,

IoT Systems - Logical Design using Python: Introduction, Installing Python, Python Data Types and Data structures, Control flow, Functions, Modules, Packages, File Handling, Operations, Classes, Python Packages of Interest for IoT.

Text Book : Ch.5.1-5.3,6.2-6.11

3.1 Introduction:

- Designing IoT systems can be a complex and challenging task.
- These systems involve interactions between various components such as IoT devices and network resources, web services, analytics components, application and database servers.
- Due to a wide range of choices available for each of these components, IoT system designers may find it difficult to evaluate the available alternatives.
- IoT system designers often tend to design IoT systems keeping specific products/services in mind.
- For such systems, 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.
- A generic design methodology for IoT system design which is independent of specific product, service or programming language as been proposed.

3.2 IoT Design Methodology:

- Figure shows the Steps involved in IoT system design methodology. To explain these steps, we use the example of a smart IoT-based home automation system.



Figure 3.1: Steps involved in IoT system design methodology

Step 1: Purpose & Requirements Specification:

- 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.
- 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.

Step 2: 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 Process specification for home automation IoT system.

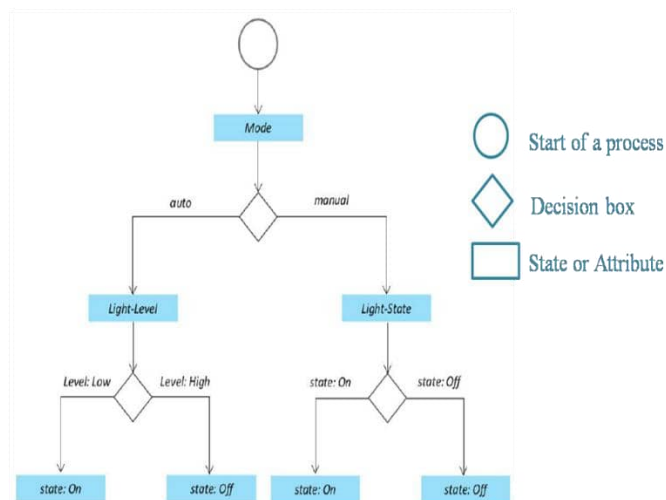


Figure 3.2: Process specification for home automation IoT system

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 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.

Figure 3.3 shows the domain model for the home automation system example. 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.). In the home automation example, there are two Physical Entities involved - one is the room in the home (of which the lighting conditions are to be monitored) and the other is the light appliance to be controlled.
Virtual Entity	Virtual Entity is a representation of the Physical Entity in the digital world. For each Physical Entity, there is a Virtual Entity in the domain model.
Device	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, perform actuation upon Physical Entities or used to identify Physical Entities
Resource	Resources are software components which can be either "on-device" or "network-resources". On-device resources are hosted on the device .Network resources include the software components that are available in network (such 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.

In the home automation example, there are three services:

- (1) a service that sets mode to auto or manual, or retrieves the current mode;

- (2) a service that sets the light appliance state to on/off, or retrieves the current light state; and
- (3) a controller service that runs as a native service on the device.

When in auto mode, the controller service monitors the light level and switches the light on/off and updates the status in the status database.

When in manual mode, the controller service retrieves the current state from the database and switches the light on/off. The process of deriving the services from the process specification and information model is described in the later section

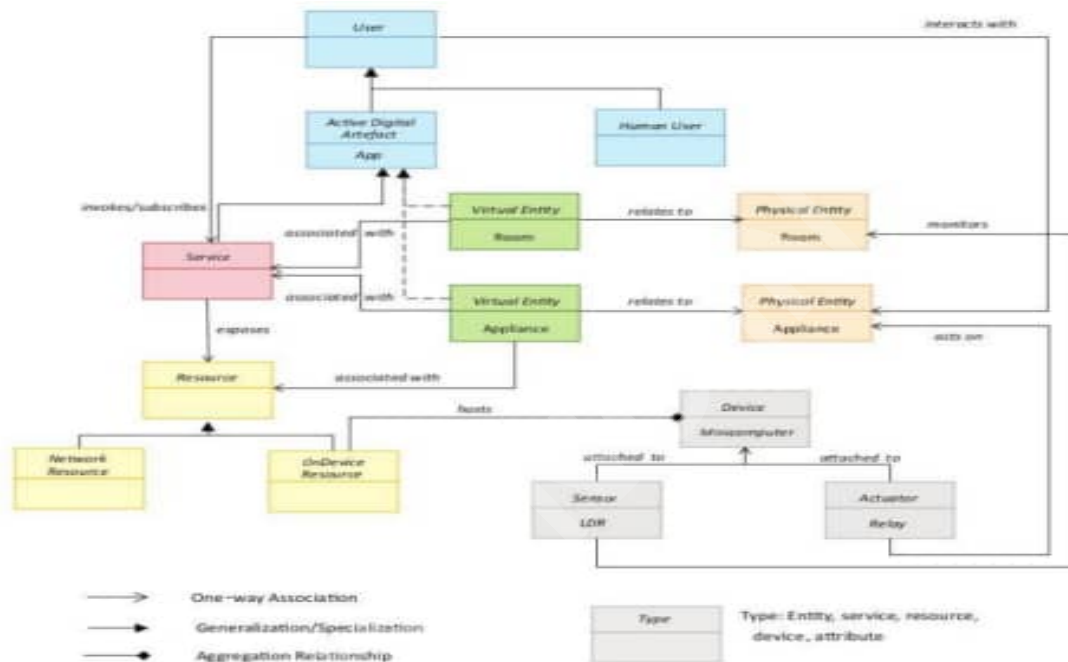


Figure 3.3: Domain model of the home automation IoT system

Step 4: Information Model Specification

- 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.

Figure 3.4 shows the Information Model for the home automation system example.

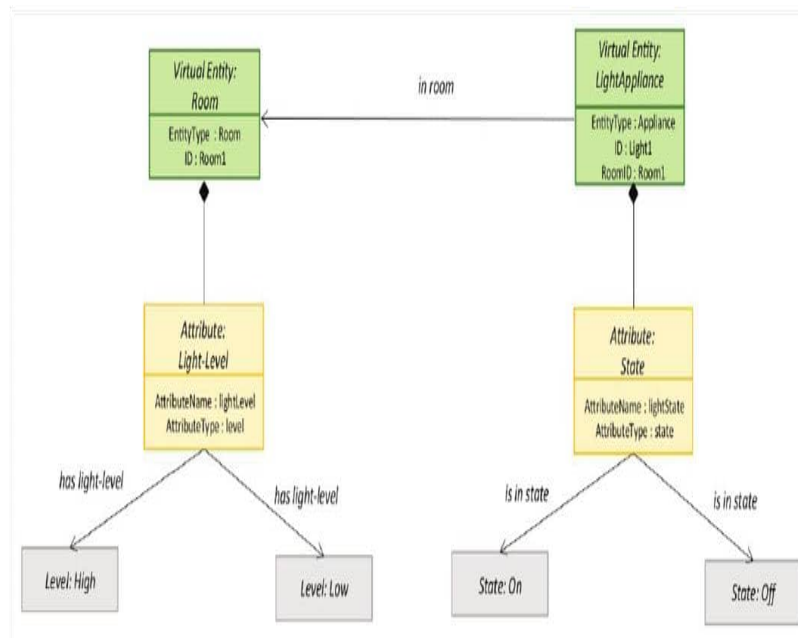


Figure 3.4: Information model of the home automation IoT system

Step 5: 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.
- 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.
- For example, the Mode service sets mode to auto or manual or retrieves the current mode. The State service sets the light appliance state to on/off or retrieves the current light state.
- 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.

Figure 5.5 shows an example of deriving the services from the process specification and information model for the home automation IoT system.

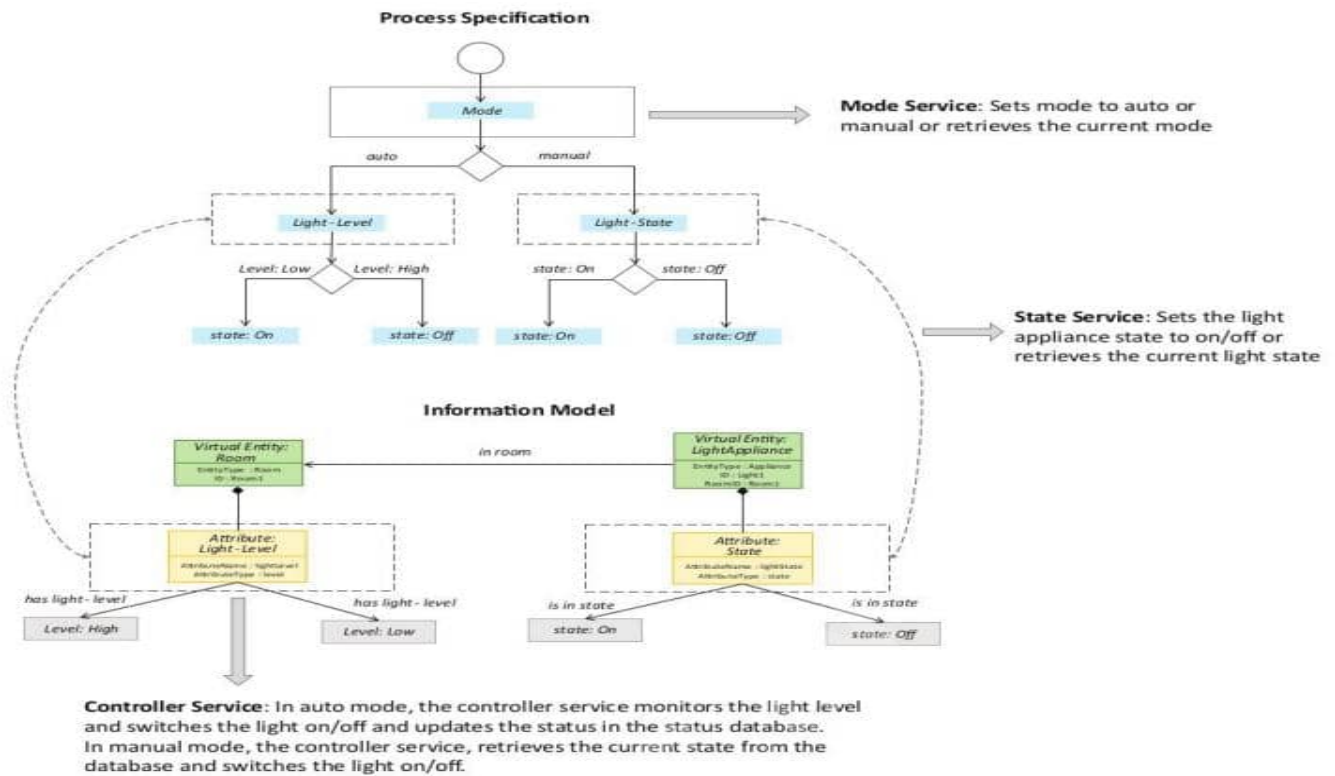


Figure 3.5: Deriving services from process specification and information model for home automation IoT system

Controller service:

Figures 3.6 show specifications of the controller service of the home automation system. The Controller service runs as a native service on the device. When in auto mode, the controller service monitors the light level and switches the light on/off and updates the status in the status database. When in manual mode, the controller service, retrieves the current state from the database and switches the light on/off.

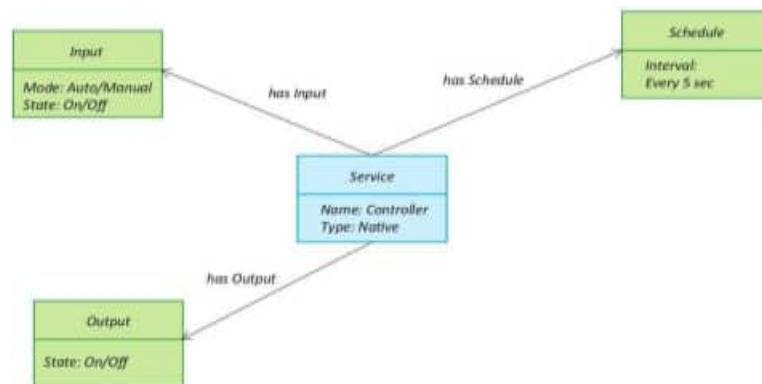


Figure 3.6: Controller service of the home automation IoT system

Mode service:

The Mode service is a RESTful web service that sets mode to auto or manual (PUT request), or retrieves the current mode (GET request). Figures 3.7 show specifications of the Mode service of the home automation system.

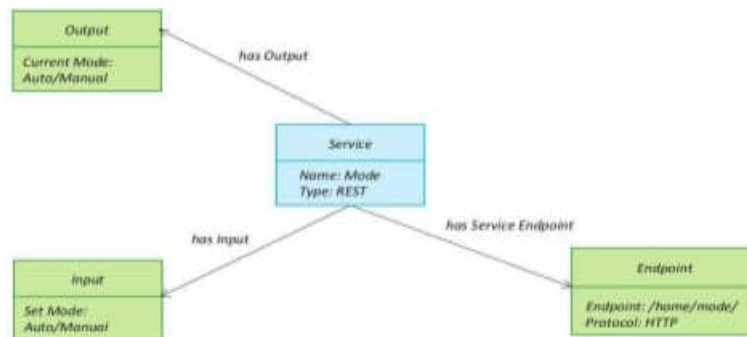


Figure 3.7: Service specification for home automation IoT system - mode service

State service:

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. Figures 3.8 show specifications of the State service of the home automation system.

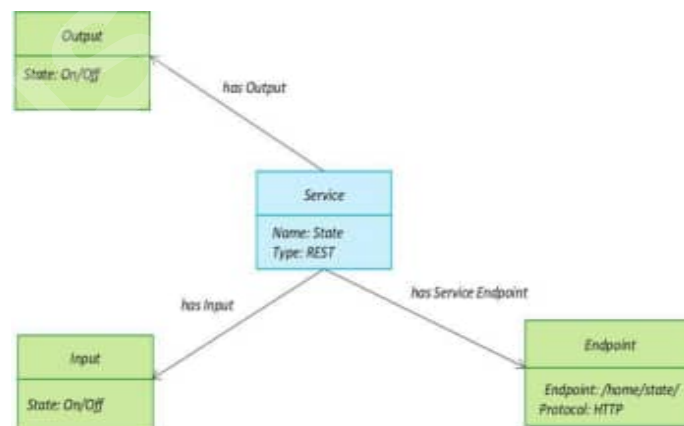


Figure 3.8: Service specification for home automation IoT system - state service

Step 6: IoT Level Specification

- The sixth step in the IoT design methodology is to define the IoT level for the system.
- Figure 3.9 shows the deployment level of the home automation IoT system, which is level-1.

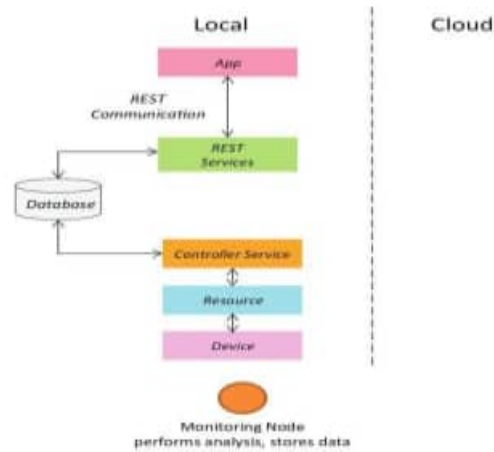


Figure 5.9: Deployment design of the home automation IoT system

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.
- 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
Communication	The communication FG handles the communication for the IoT system. It includes communication protocols that form the backbone of IoT systems and enable network connectivity. It also includes the communication APIs (such as REST and WebSocket) 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. In the home automation example, there are two REST services (mode and state service) and one native service (controller service).
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.

Figure 5.10 shows an example of mapping deployment level to functional groups for home automation IoT system.

IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management). Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols). Controller service maps to the Services FG (native service). Web Services map to Services FG. Database maps to the Management FG (database management) and Security FG (database security). Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security).

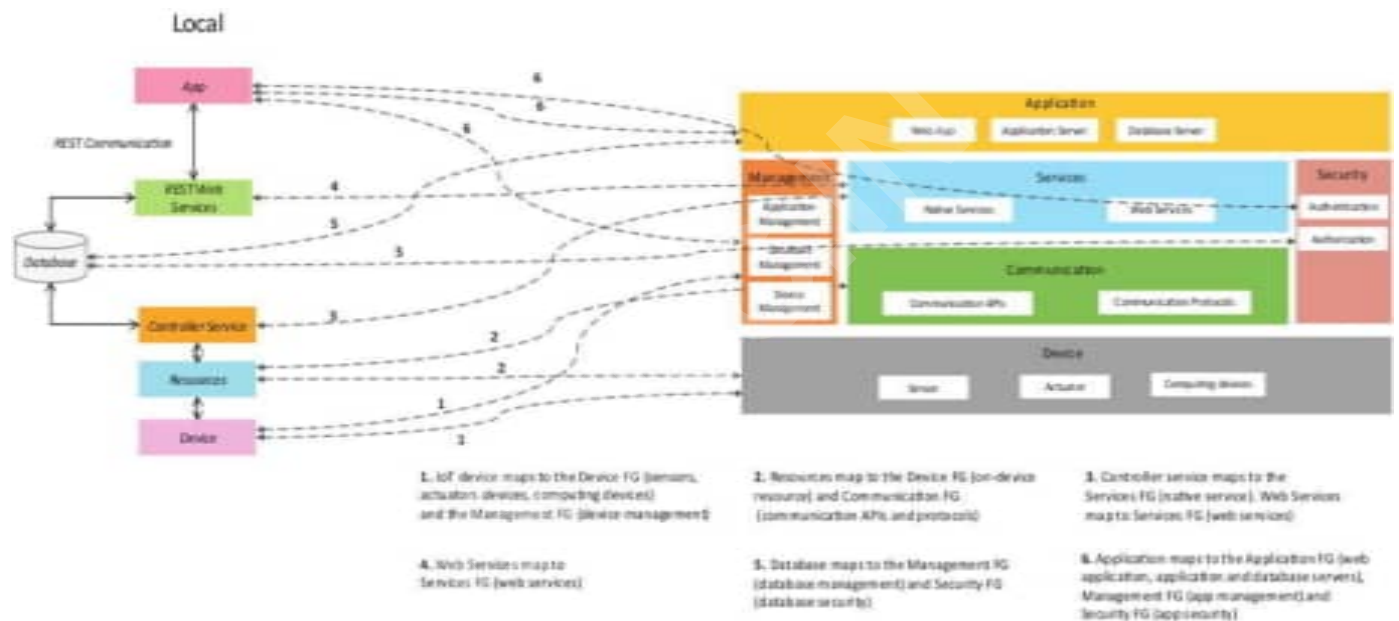


Figure 3.10: Mapping deployment level to functional groups for home automation IoT system.

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

Figure 3.11 shows an example of mapping functional groups to operational view specifications for home automation IoT system.

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.

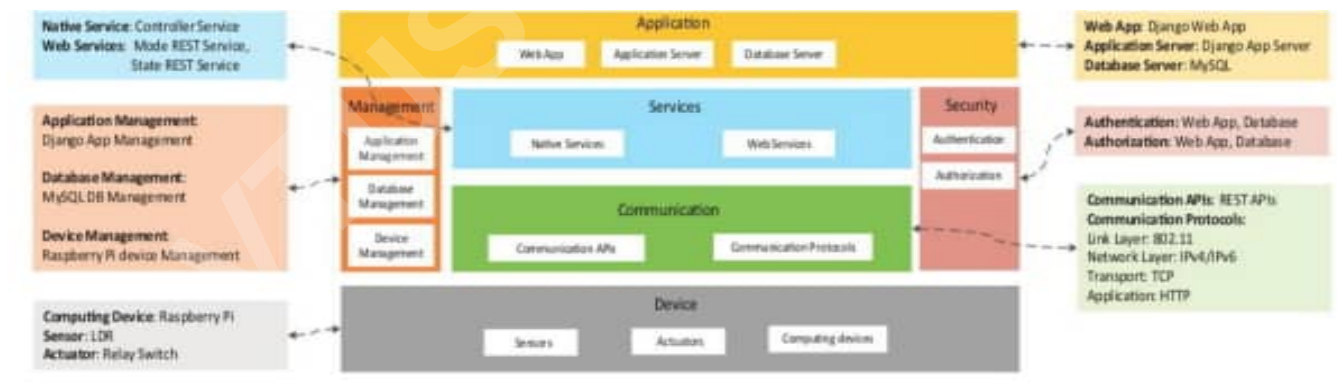


Figure 5.11: Mapping functional groups to operational view for home automation IoT system.

Step 9: Device & Component Integration

- The ninth step in the IoT design methodology is the integration of the devices components.
- Figure shows a schematic diagram of the home automation IoT system.
- The devices and components used in this example are Raspberry Pi mini computer, LDR sensor and relay, switch actuator.

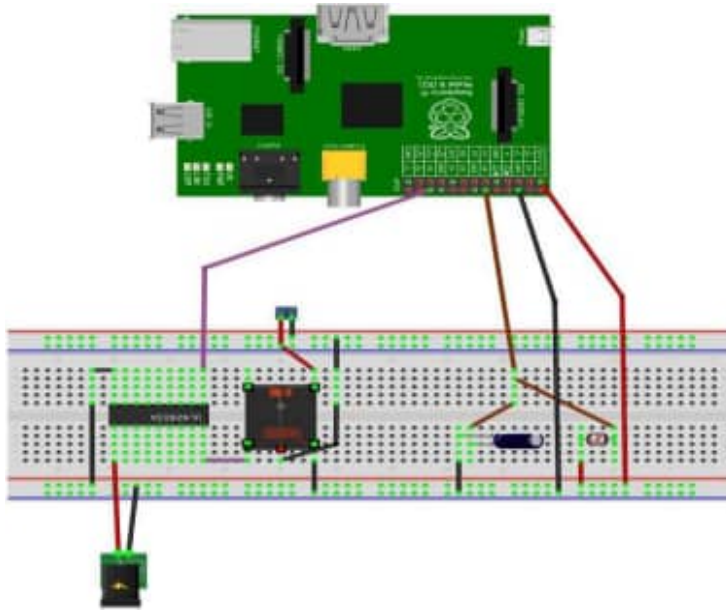


Figure 3.12: Schematic diagram of the home automation IoT system showing the device, sensor and actuator integrated

Step 10: Application Development

- The final step in the IoT design methodology is to develop the IoT application.
- Figure 3.13 shows a screenshot of the home automation web application.
- 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.

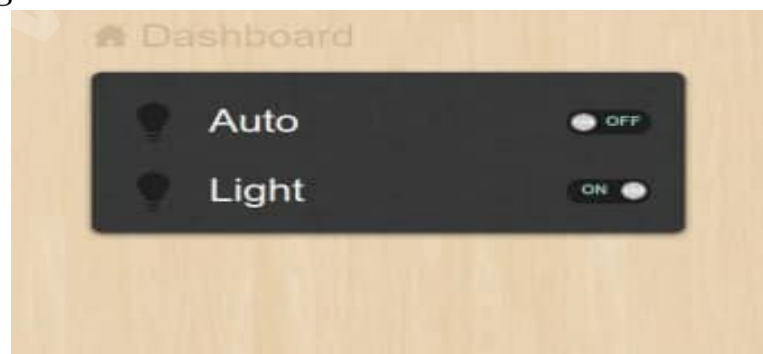


Figure 3.13: Home automation web application screenshot

Case Study on IoT System for Weather Monitoring

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. Figure 3.14 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.

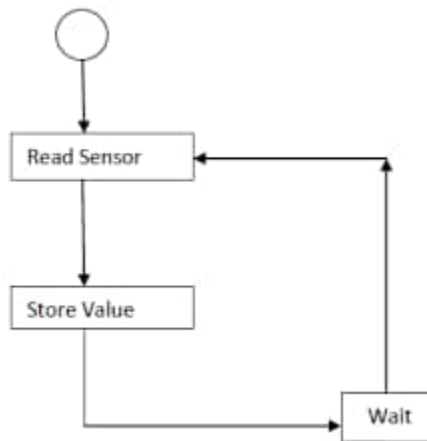


Figure 3.14: Process specification for weather monitoring IoT system

Figure 3.15 shows the domain model for the weather monitoring system. 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 minicomputer. 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 the readings to the cloud.

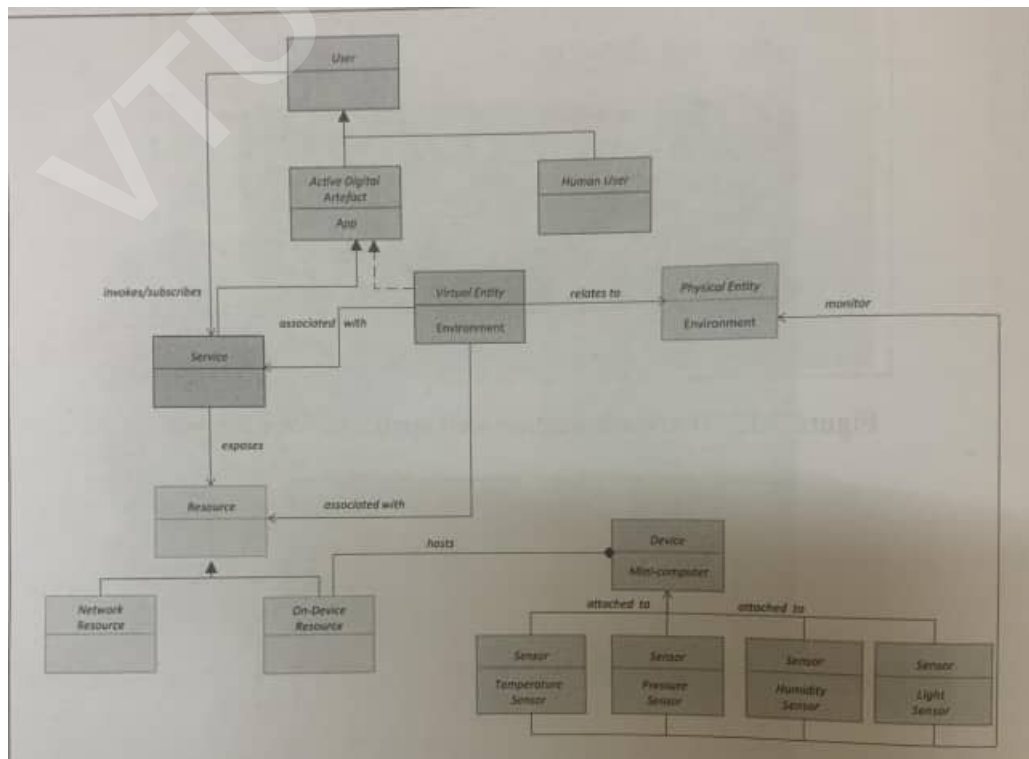


Figure 3.15: Domain model for weather monitoring IoT system

Figure 3.16 shows the information model for the weather monitoring system. In this example, there is one virtual entity for the environment being sensed. The virtual entity has attributes - temperature, pressure, humidity and light.

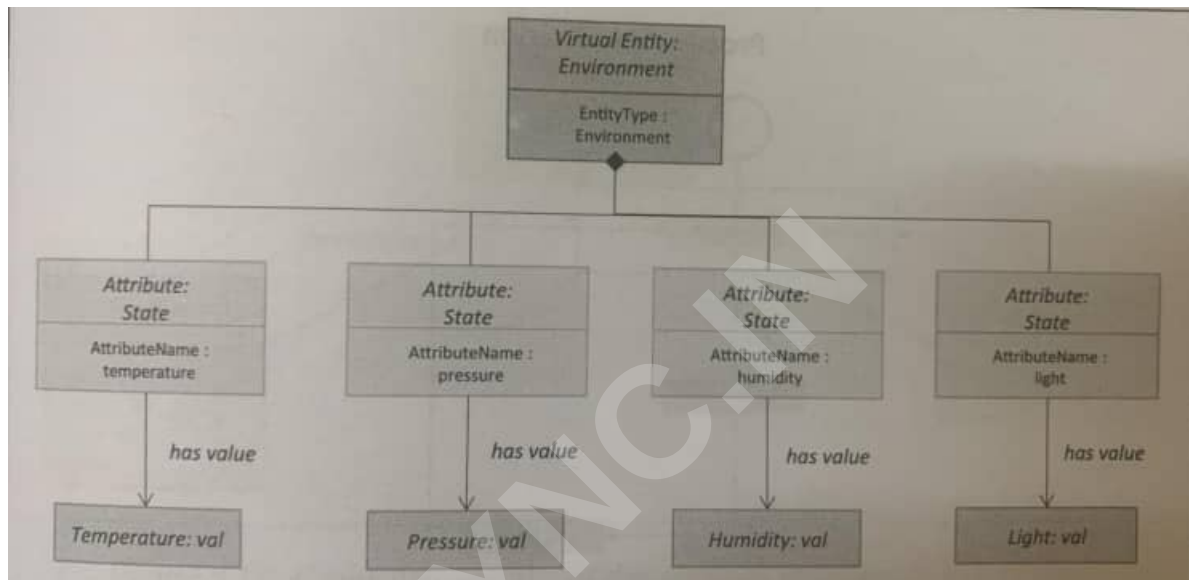


Figure 3.16: Information model for weather monitoring IoT system

Figure 3.17 shows an example of deriving the services from the process specification and information model for the weather monitoring system.

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.

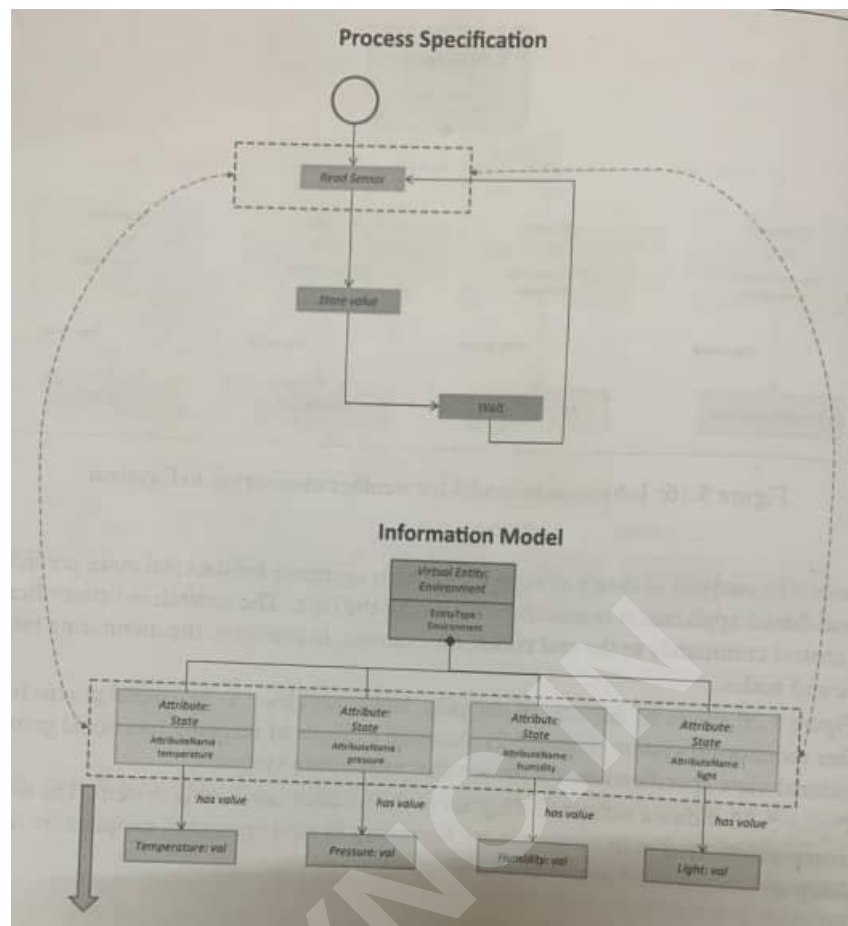


Figure 3.17: Deriving services from process specification and information model for weather monitoring IoT system.

Figure 3.18 shows the specification of the controller service for the weather monitoring system. 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.

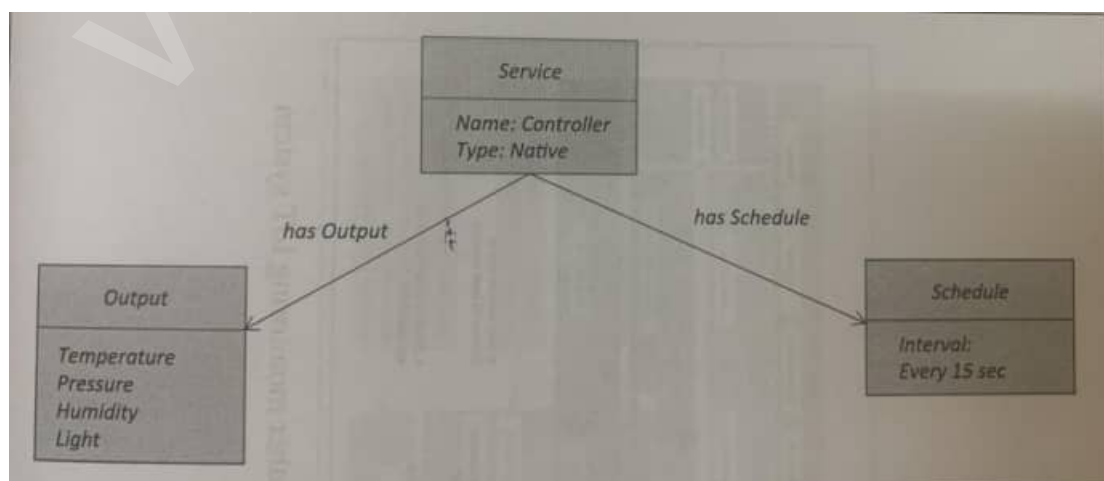


Figure 3.18: Controller service of the weather monitoring IoT system

Figure 3.19 shows the 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 (such as temperature, pressure, humidity and light). 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. A cloud-based application is used for visualizing the data. The centralized controller can send control commands to the end nodes.

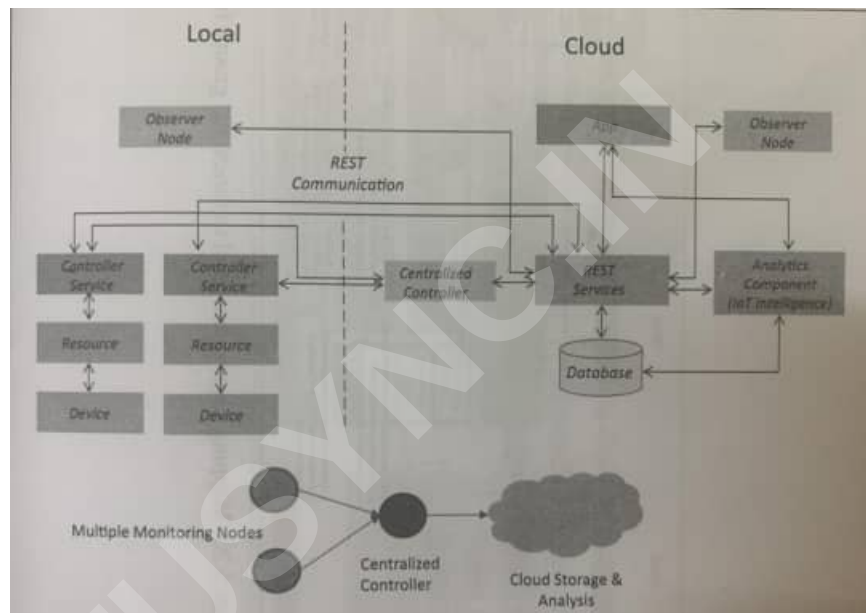


Figure 3.19: Deployment design of the weather monitoring IoT system

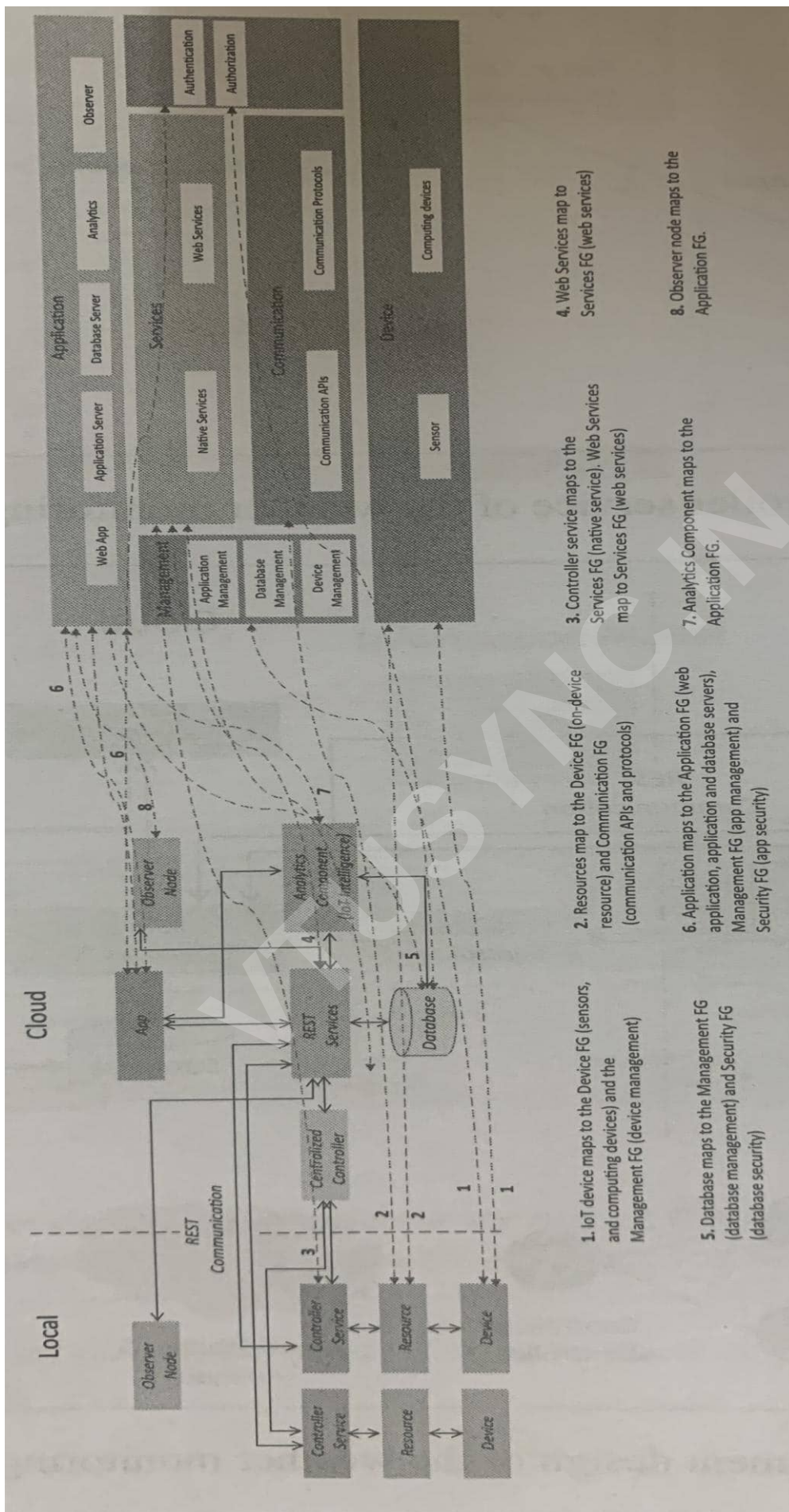


Figure 5.20 shows an example of mapping deployment level to functional groups for the weather monitoring system.

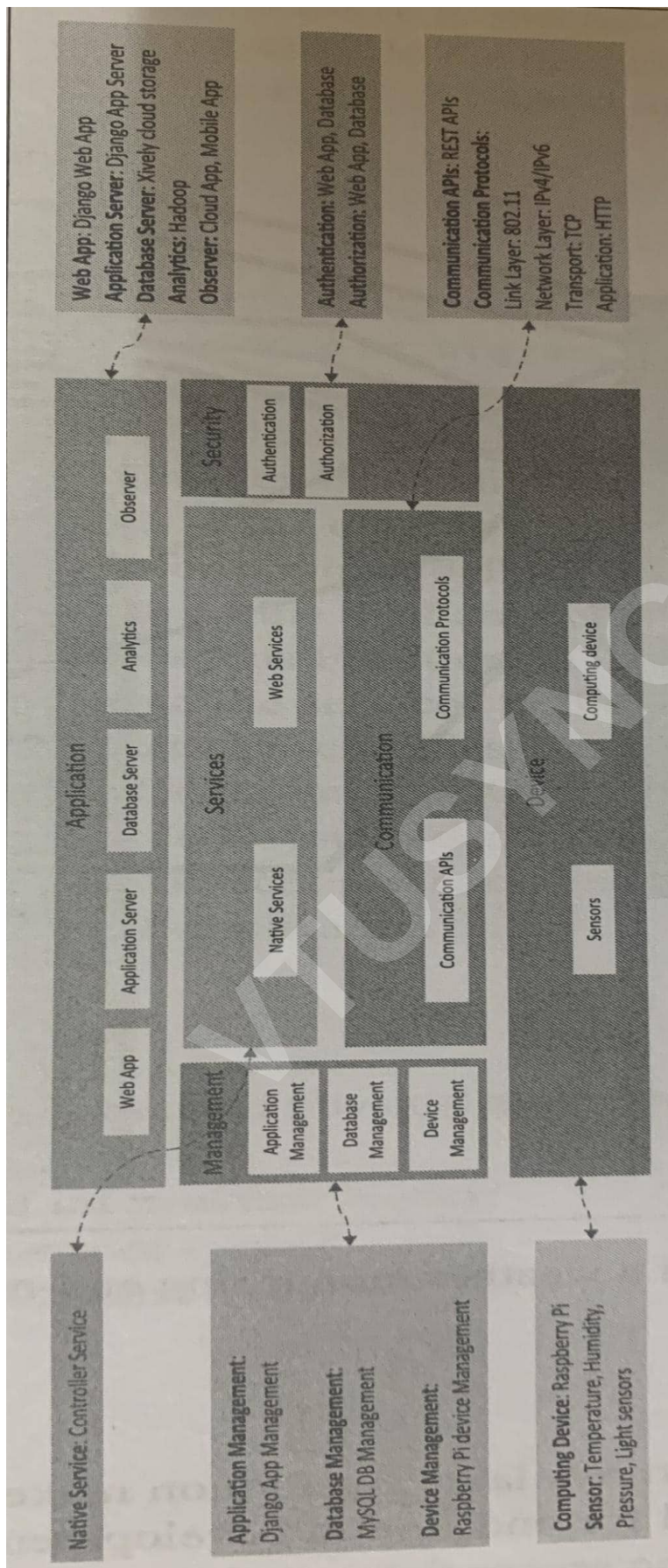


Figure 5.21: Mapping functional groups to operational view for the weather monitoring IoT system

