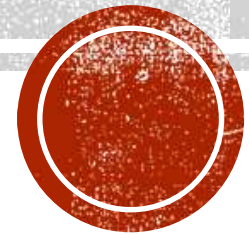


IOT PLATFORMS DESIGN METHODOLOGY

UNIT - II



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 makes updating the system design to add new features or replacing a particular product/service choice for a component very complex, and in many cases may require complete re design of the system.
- 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 PLATFORMS DESIGN METHODOLOGY

It includes:

- 1) Purpose & Requirements Specification
- 2) Process Specification
- 3) Domain Model Specification Information
- 4) Model Specification Service Specification
- 5) IoT Level Specifications
- 6) Functional view Specification
- 7) Operational View Specification
- 8) Device & Component Integration
- 9) Application Development





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 are captured.

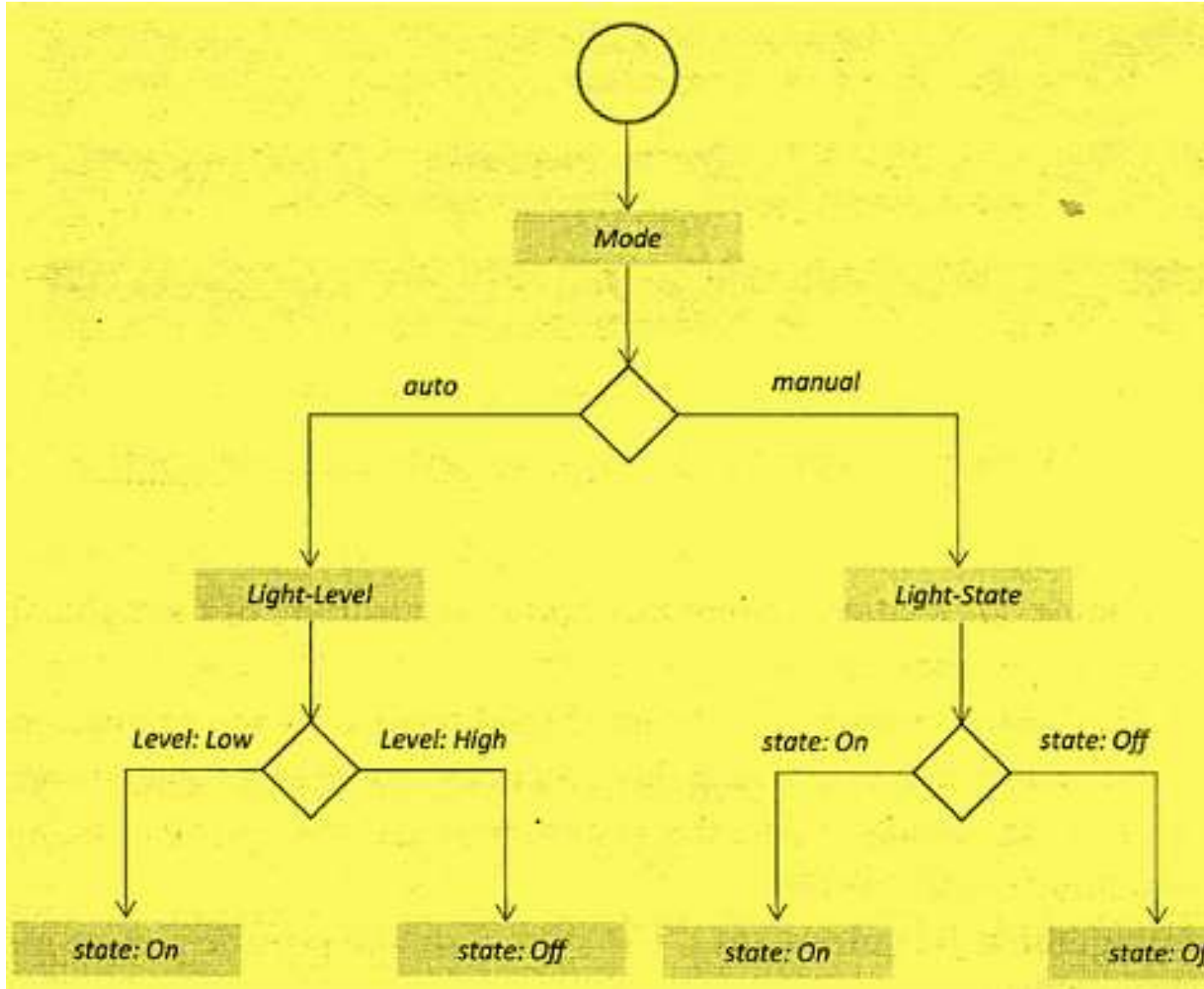
- **Purpose** : A home automation system that allows controlling of the lights in a home remotely using a web application.
- **Behavior** : The home automation system should have auto and manual modes. In auto mode, the system measures the light level in the room and switches on the light when it gets dark. In manual mode, the system provides the option of manually and remotely switching on/off the light.



- **System Management Requirement** : 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



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



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.



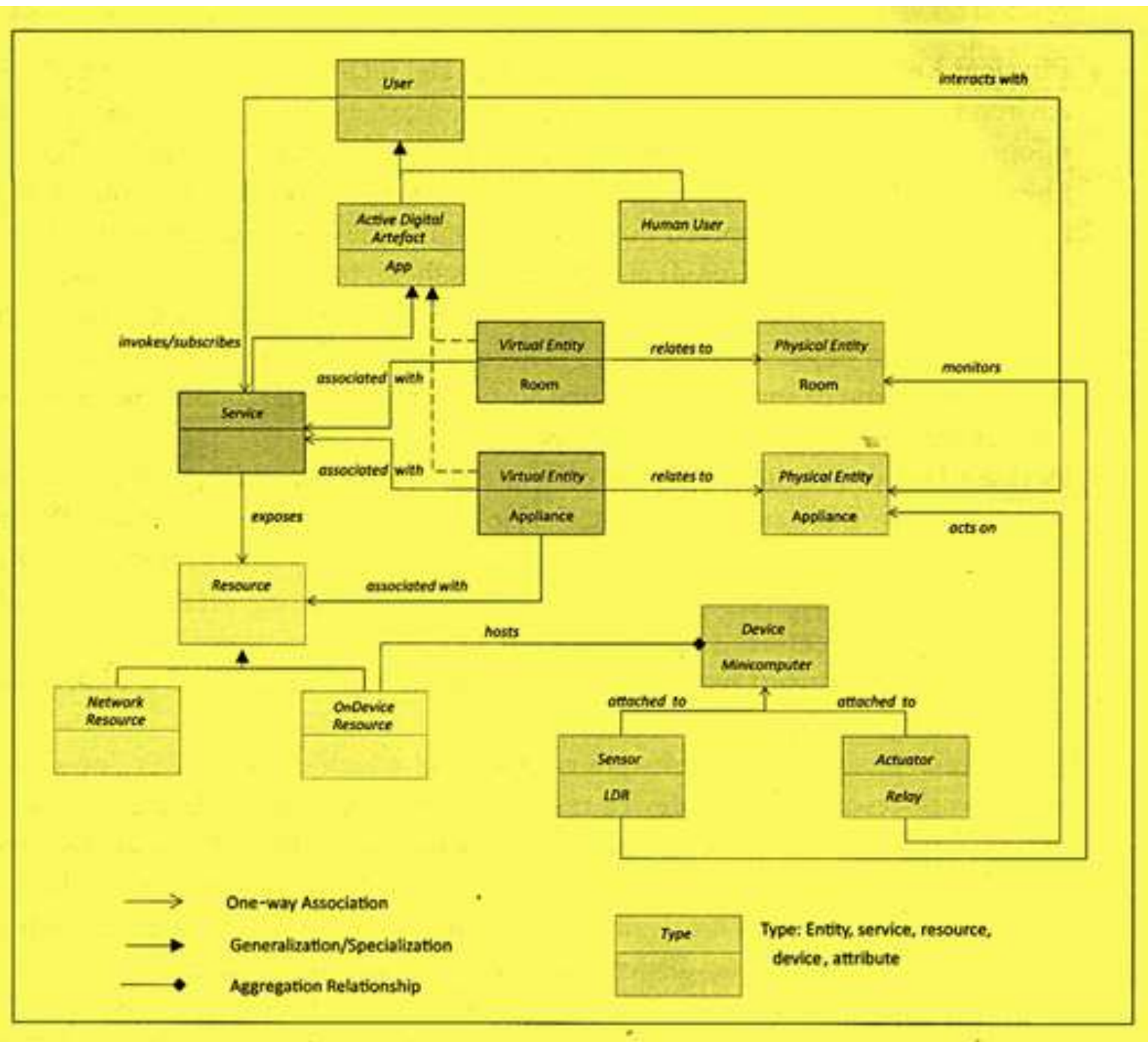
The entities, objects and concepts defined in the domain model include

- **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.



- **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.

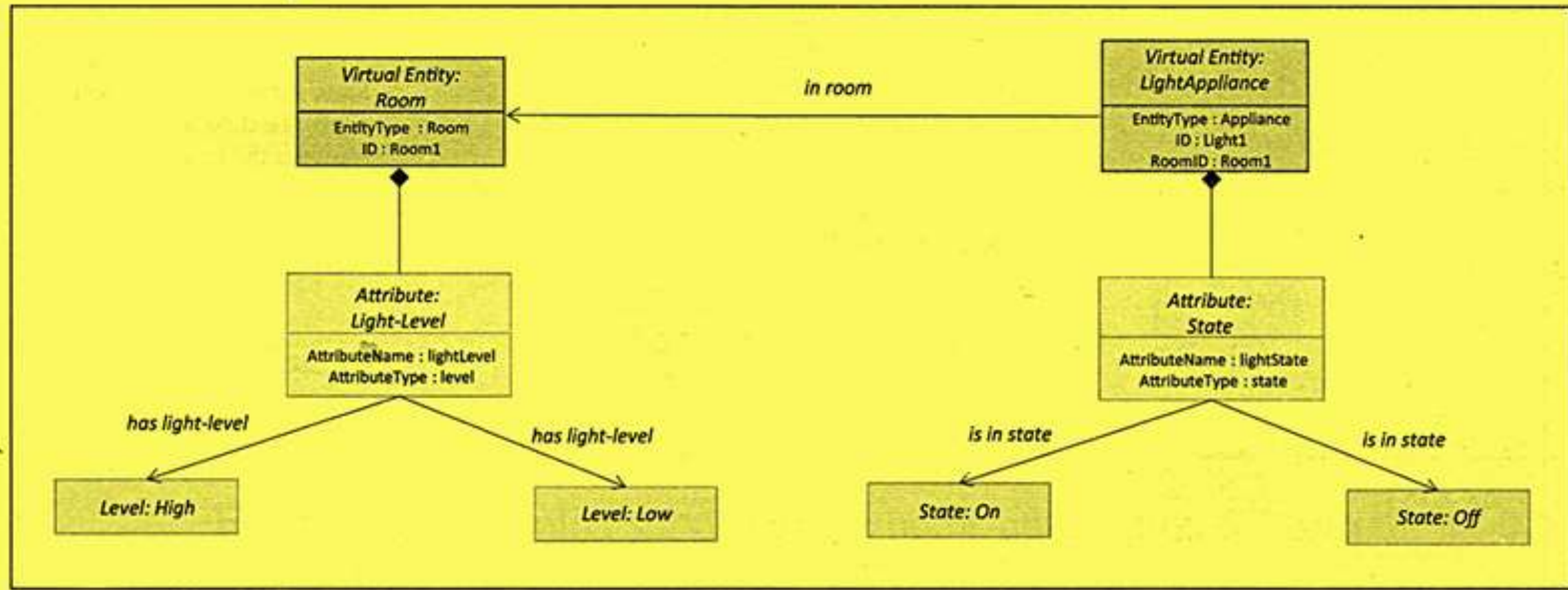




INFORMATION MODEL SPECIFICATION

- 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



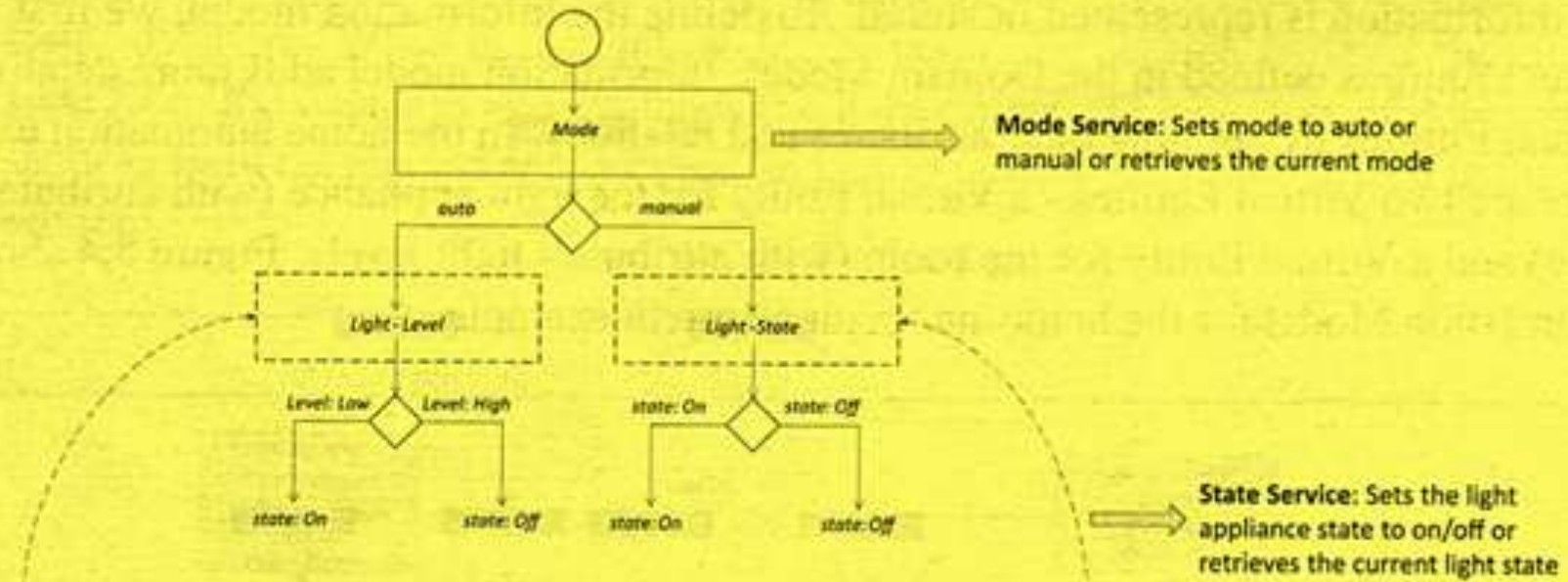


SERVICE SPECIFICATION

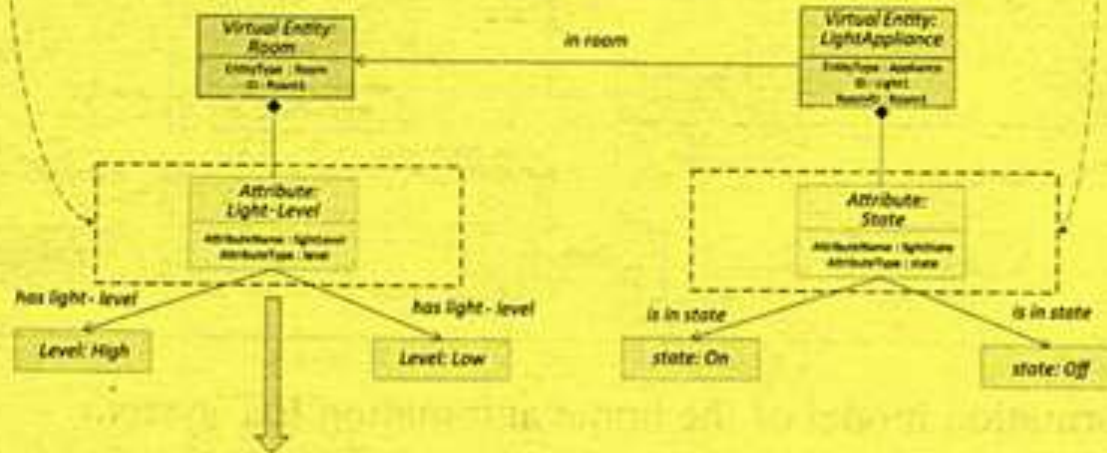
- The fifth step in the IoT design methodology is to define the service specifications.
- Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects .
- From the process specification and information model, we identify the states and attributes .
- For each state and attribute define a service . we These services either change the state or attribute values or retrieve the current values .



Process Specification



Information Model

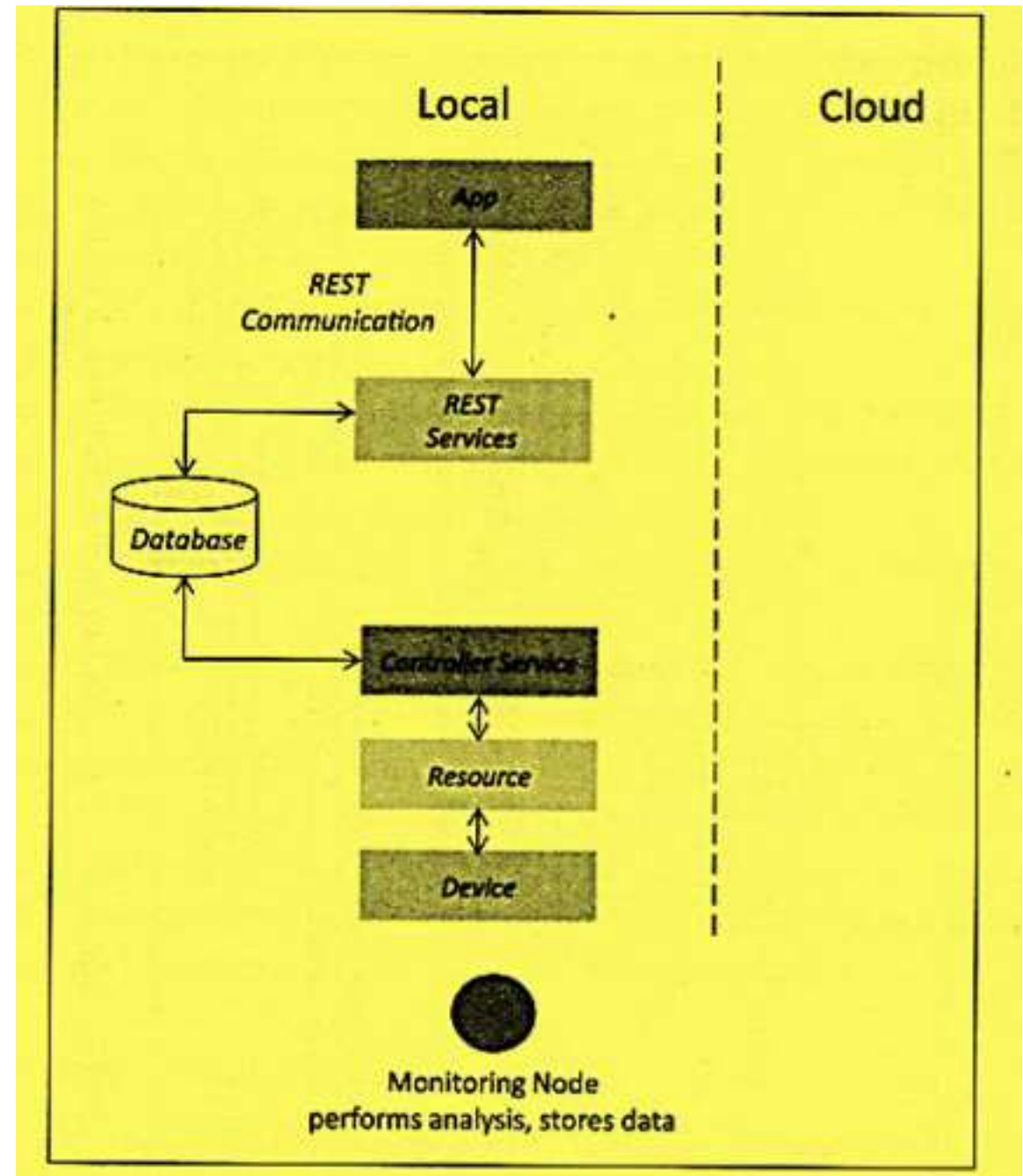


Controller Service: In auto mode, the controller service monitors the light level and switches the light on/off and updates the status in the status database. In manual mode, the controller service, retrieves the current state from the database and switches the light on/off.



IOT LEVEL SPECIFICATIONS

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



FUNCTIONAL VIEW SPECIFICATION

- 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 .



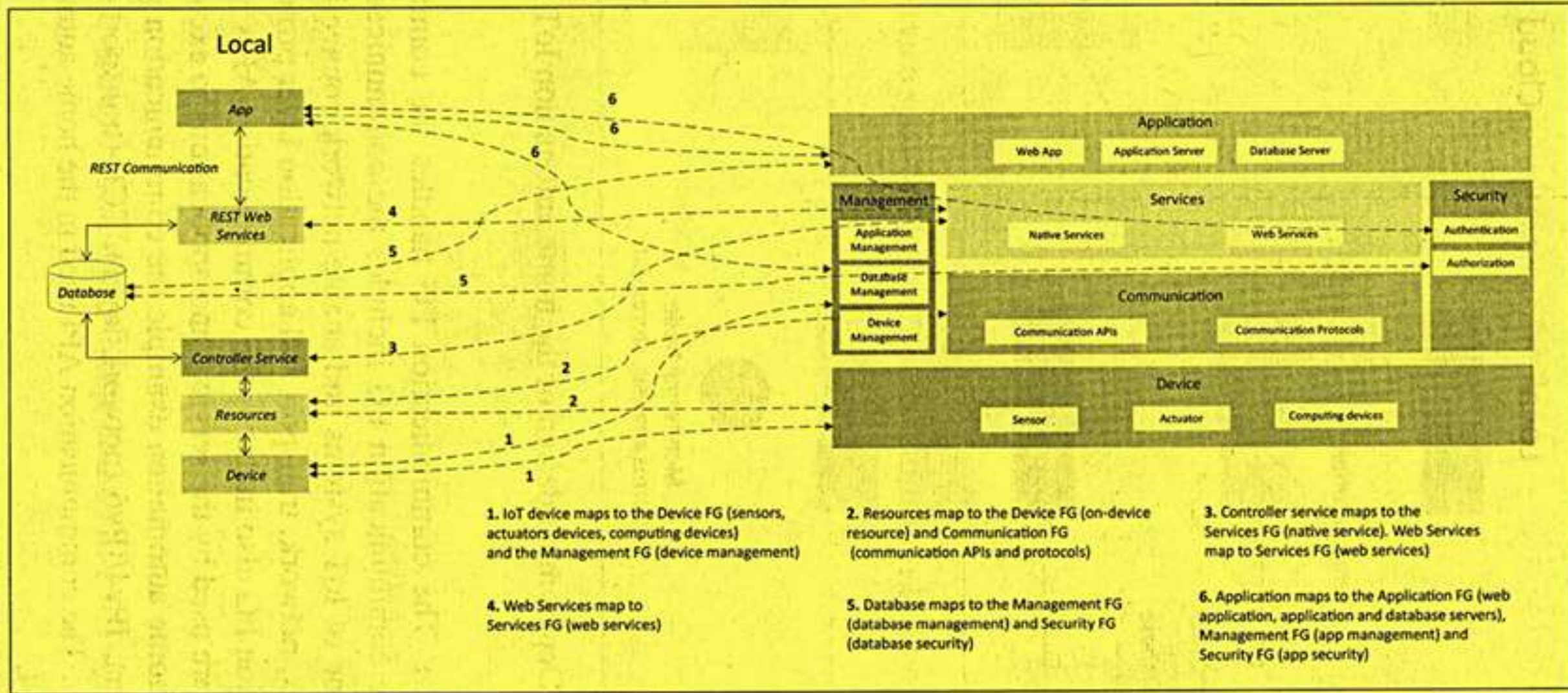
The Functional Groups (FG) included in a Functional View include:

- **Device** : The device FG contains devices for monitoring and control. In the home automation example, the device FG includes a single board minicomputer, 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 (such as REST and WebSocket APIs) 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 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.



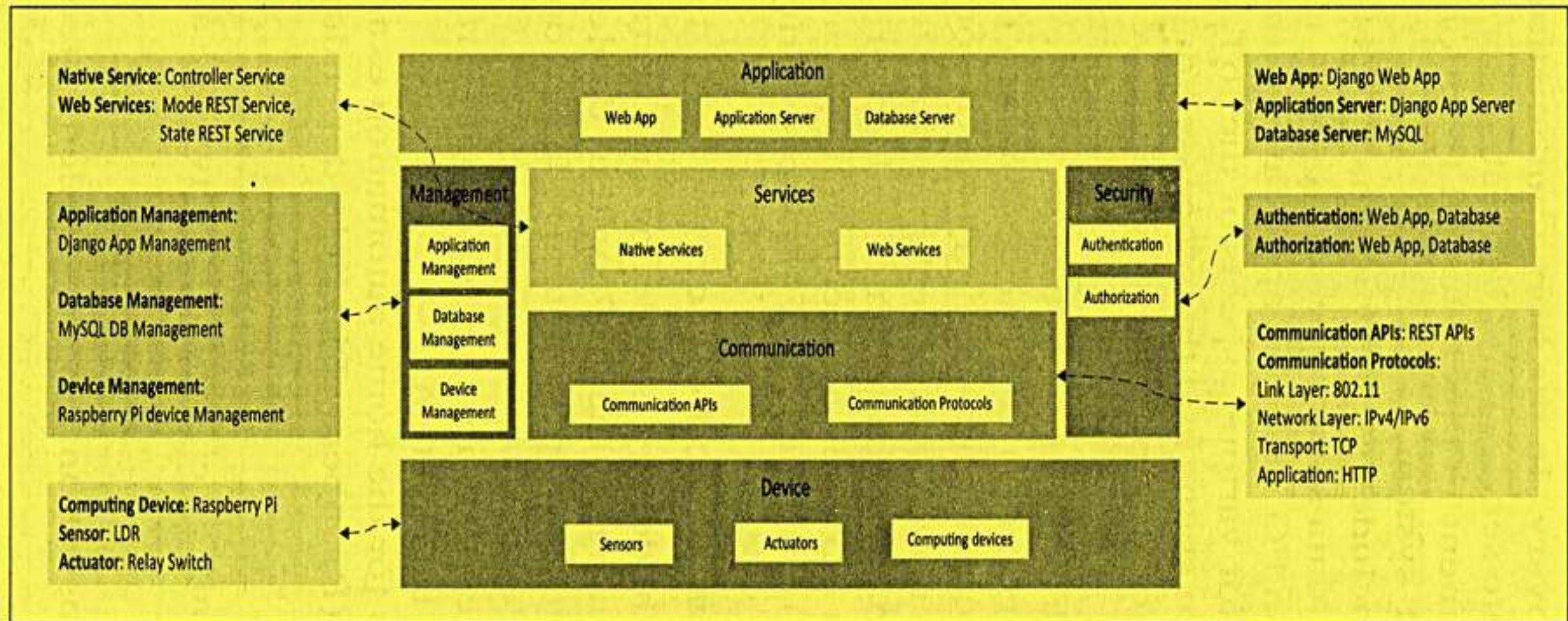


OPERATIONAL VIEW SPECIFICATION

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.

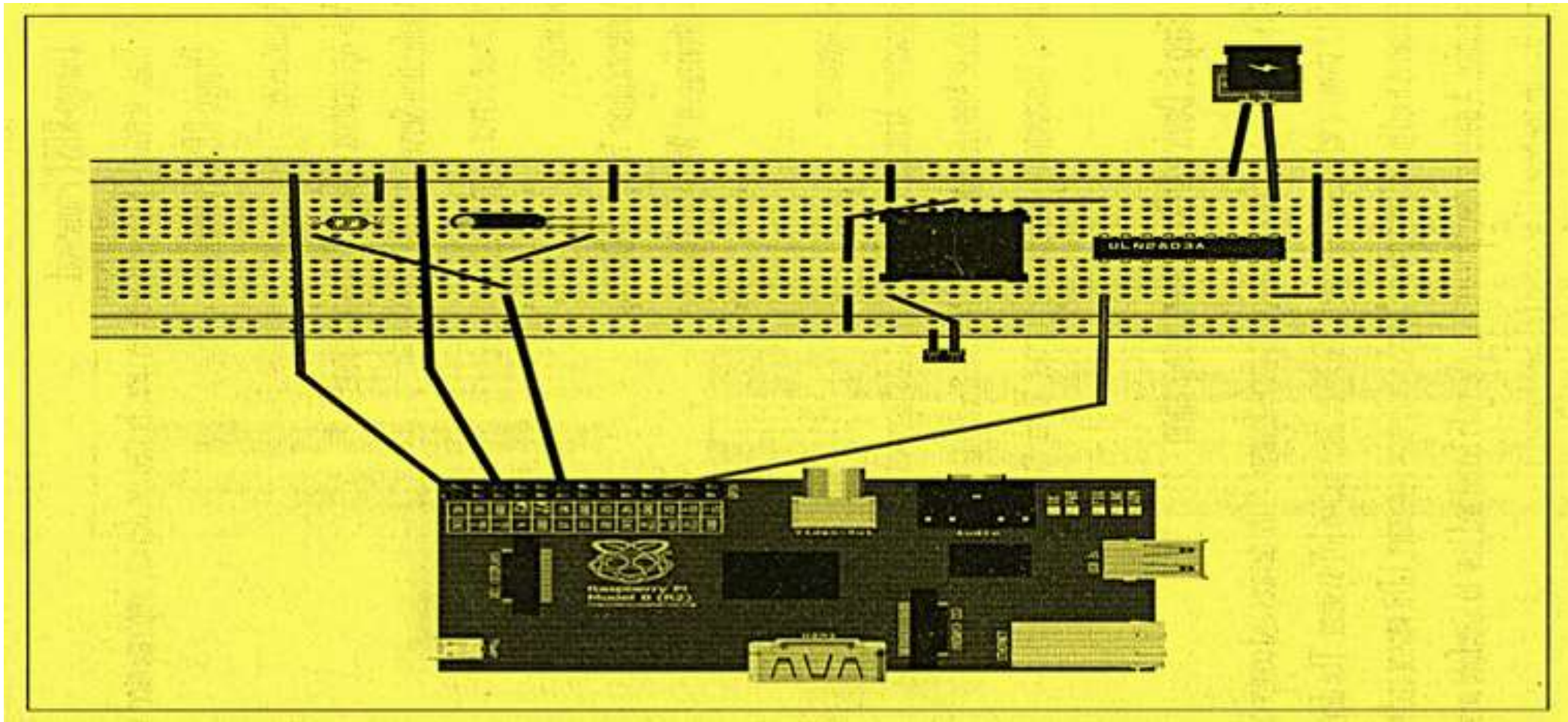
- Operational View specifications for the home automation example are as follows:
- **Devices** : Computing device (Raspberry Pi), light dependent resistor (sensor), relay switch (actuator).
- **Communication APIs** : REST APIs.
- **Communication Protocols**: Link Layer 802.11, Network Layer 1Pv4/1Pv6, Transport TCP, Application HTTP.





DEVICE & COMPONENT INTEGRATION

- The devices and components used in this example are Raspberry Pi, 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.

