Tishk International University
Science Faculty
IT Department

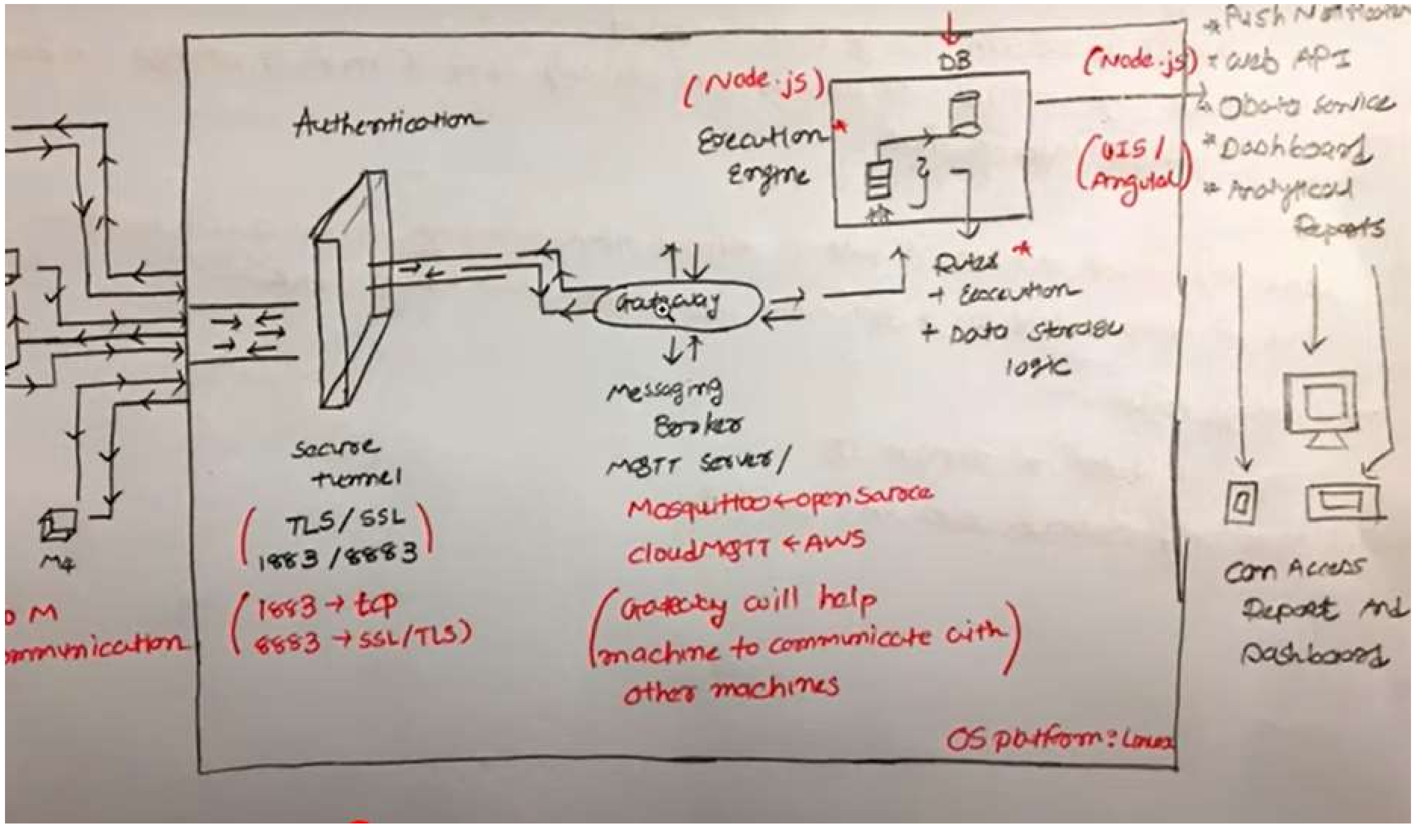
Introduction to IoT

Lecture 06: IoT Design Methodology

4th Grade - Spring Semester

Instructor: Alaa Ghazi

Lecture 6
IoT Design Methodology



Lecture Topics

❑ Introduction to IoT Design Methodology

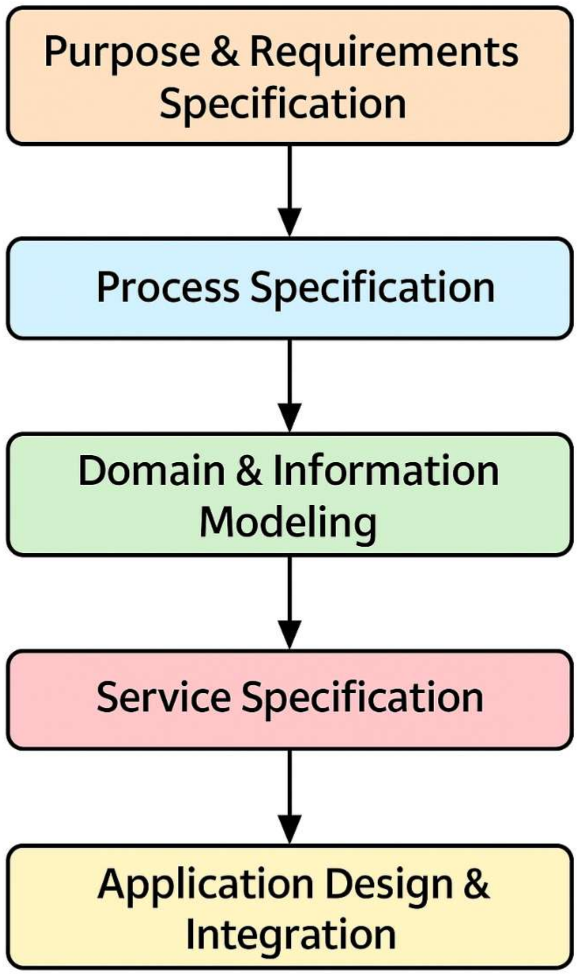
1. Purpose & Requirements Specification,
2. Process Specification,
3. Domain & Information Modeling,
4. Service Specification, and
5. Application Design & Integration.

Introduction to IoT Design Methodology

- **IoT design methodology** is a structured approach for developing Internet of Things systems. It provides a step-by-step guide for translating user needs and technical requirements into an operational IoT system.
- In this lecture, we simplify the methodology into 5 main structured stages:

1. Purpose & Requirements Specification,
2. Process Specification,
3. Domain & Information Modeling,
4. Service Specification, and
5. Application Design & Integration.

Structured IoT Design Methodology


Purpose & Requirements Specification
Process Specification
Domain & Information Modeling
Service Specification
Application Design & Integration

Benefits of Structured IoT Design

- Easier planning and requirement alignment
- Better system modularity and scalability
- Helps document logic for testing and deployment
- Encourages reuse of components and services

1. Purpose & Requirements Specification

- This phase defines why the IoT system is needed and what it is expected to do. It includes:
  - Purpose and behavior of the system
  - Data collection needs
  - System management and User Interface needs
  - Data privacy and security requirements
- A clear requirement specification ensures that users agree on what the system is meant to achieve.

Example – Purpose & Requirements (Smart Light)

- Purpose: A smart light system that automatically switches lights on or off based on environmental lighting and user commands.
- Requirements:
  - Auto Mode: Light turns on when ambient light is below a threshold.
  - Manual Mode: Users can control the light remotely.
  - Security: User authentication required.
  - Data: Sensor readings stored for audit or analytics.

2. Process Specification

- This step describes how the system will work operationally. It includes detailed use cases and the sequence of operations derived from the requirements.
- For each use case, identify:
  - Inputs (e.g., sensor data)
  - Outputs (e.g., light state)
  - Triggers (e.g., time of day, user action)
  - Logical steps the system takes to handle those triggers.

Example – Process Flow (Smart Light)

1. The system starts and sets its mode (auto/manual).
2. In auto mode, the light sensor reads brightness every 5 seconds.
3. If brightness < threshold, turn light ON; else, turn OFF.
4. In manual mode, the system waits for user input via web/mobile app.
5. Based on the command, the light is switched ON or OFF.
6. Status is stored in a local database.

3. Domain & Information Model

- The Domain Model defines key entities (objects) and their relationships. The Information Model builds upon this by defining attributes and data structure of each entity.
- For example:
  - Domain: Light, Sensor, Controller
  - Information: Light.state, Sensor.lux, Controller.mode
- This abstraction helps organize the system in terms of components and their roles.

Modeling in IoT Systems

- Virtual Entities:
  - Light: physical actuator, states = on/off
  - Sensor: detects light levels, outputs = lux
  - Mode: user selection (auto/manual)
- Relationships:
  - Sensor feeds data to Controller
  - Controller sends commands to Light
  - Mode affects Controller logic.

4. Service Specification

- Services define what the system offers and how it interacts with the environment and users.
- Each service should include:
  - Type of service (e.g., sensing, actuation, notification)
  - Inputs and outputs
  - Preconditions (e.g., active mode)
  - Effects (e.g., light turns on)
- Services should be modular and reusable.

Example – Services for Smart Light

1. Auto Light Service:
   - Input: light sensor value
   - Output: turn light on/off
   - Runs when mode = auto
2. Manual Control Service:
   - Input: user button press
   - Output: toggle light
   - Runs when mode = manual
3. Status Update Service:
   - Input: current light/mode
   - Output: send status to User Interface

5. Application Design & Integration

- This step focuses on the technical implementation and integration of all hardware and software components.
- It includes:
  - System architecture (logical/physical)
  - Device connectivity (sensors, actuators)
  - Communication protocols (e.g., HTTP, MQTT)
  - Backend services and user interface

Example – Full System Layout (Smart Light)

1. Raspberry Pi as the main controller
2. Light Sensor (LDR) connected via GPIO
3. Relay module to switch light on/off
4. Python script reads sensor and runs logic
5. ThingsBoard web interface widgets to store and visualize data

System responds to sensor data or user commands and acts accordingly.

END OF SEMESTER

Wish You Good Luck