# Unit 43- Internet of Things

**Week1_Lecture**

**Presented by – Daw Thi Thi Thandar Saw Htay**

**thithithandar@gusto-education.com**

# Learning Objectives

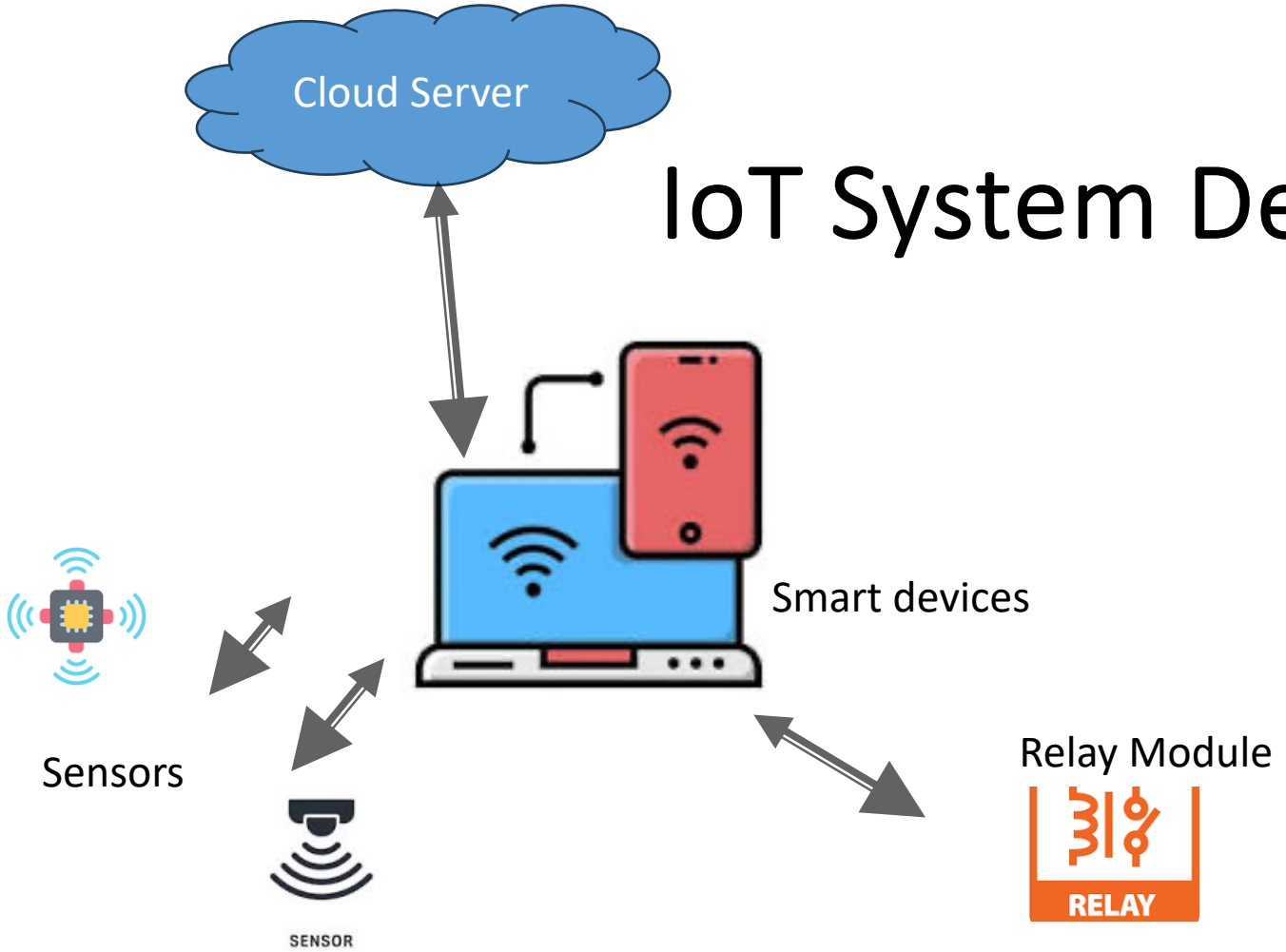
- Purpose of Internet of Things
- Characteristics of IoT
- IoT Architecture and Frameworks
- IoT interoperability and its design considerations
- Application of Internet of Things

# System Design

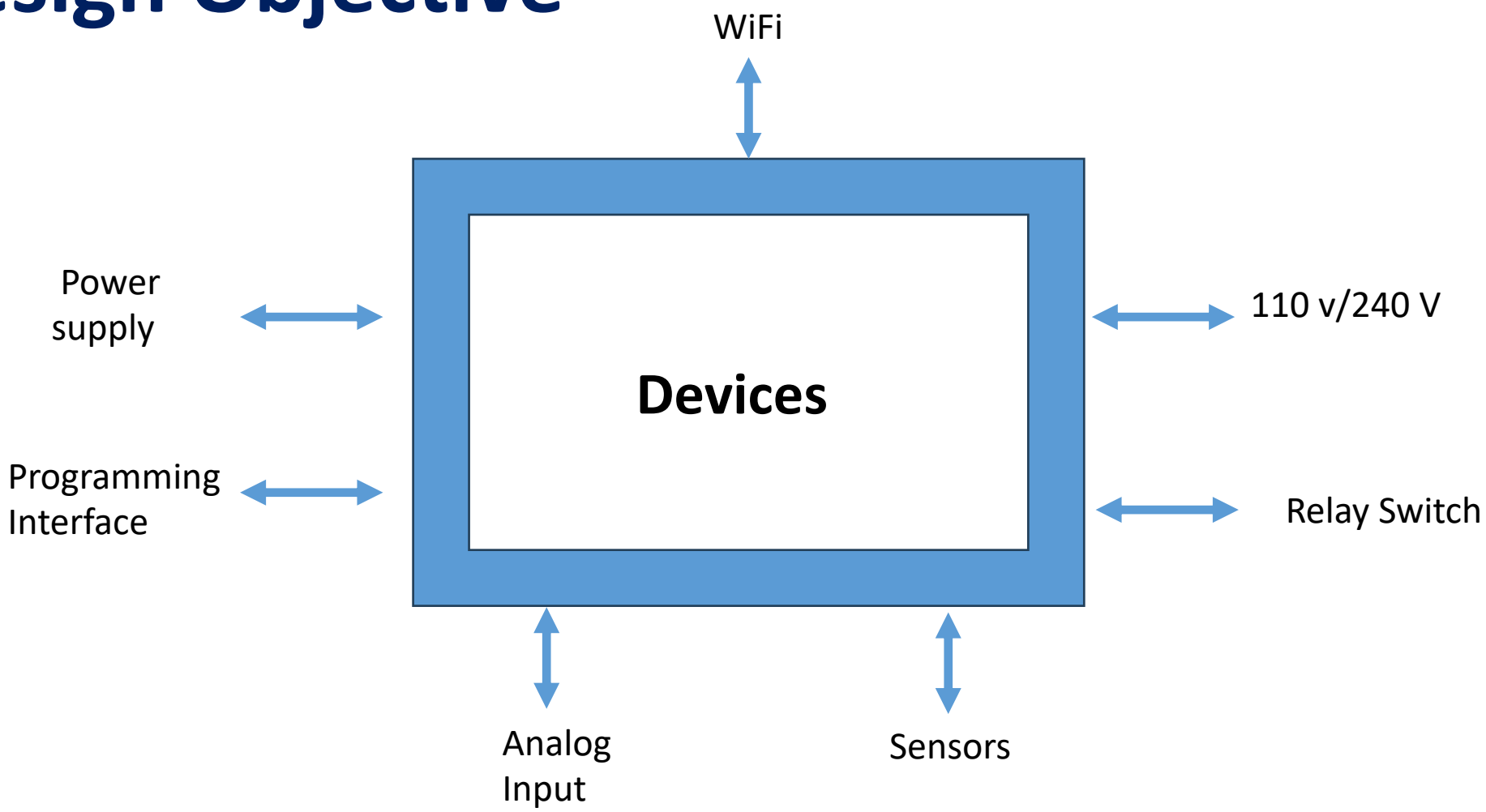
# Definition: System Design

- **System design** is the process of defining the architecture, modules, interfaces, and data for a **system** to satisfy specified **requirements**.
- **System design** could be seen as the **application** of **system's theory** to **product development**.

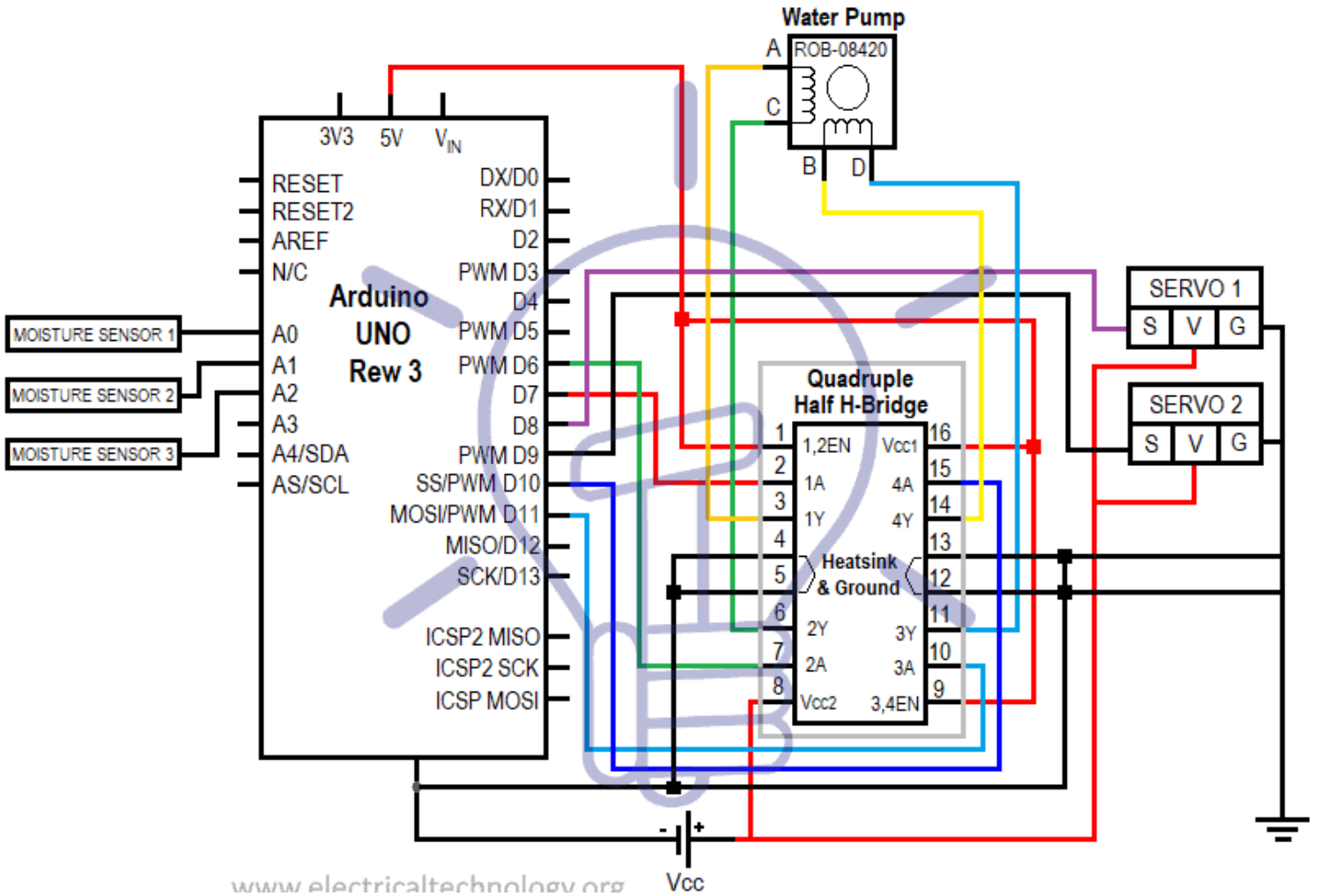
# IoT System Design



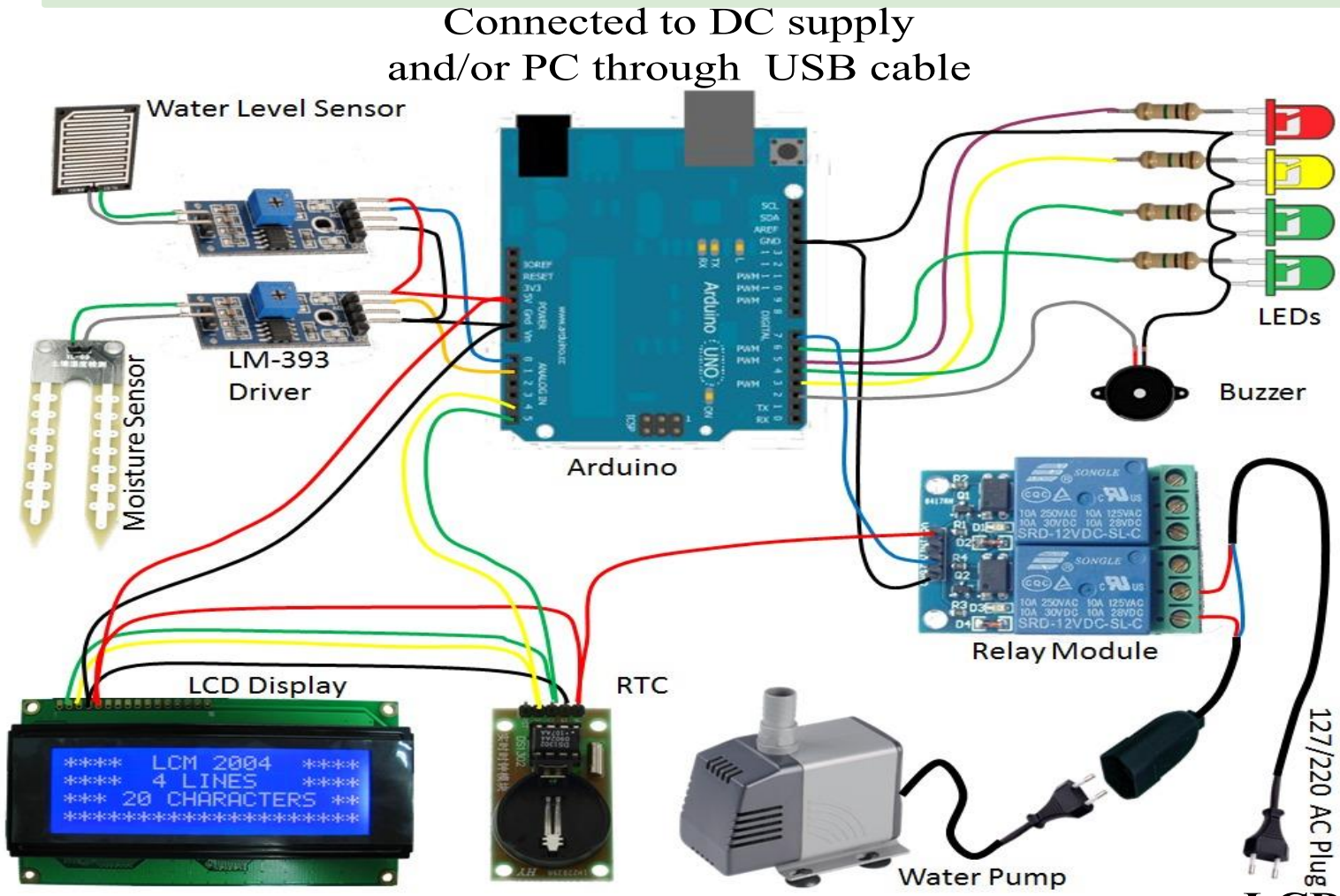
# Design Objective



# Hardware design for Plant watering system



# Example: Arduino based System

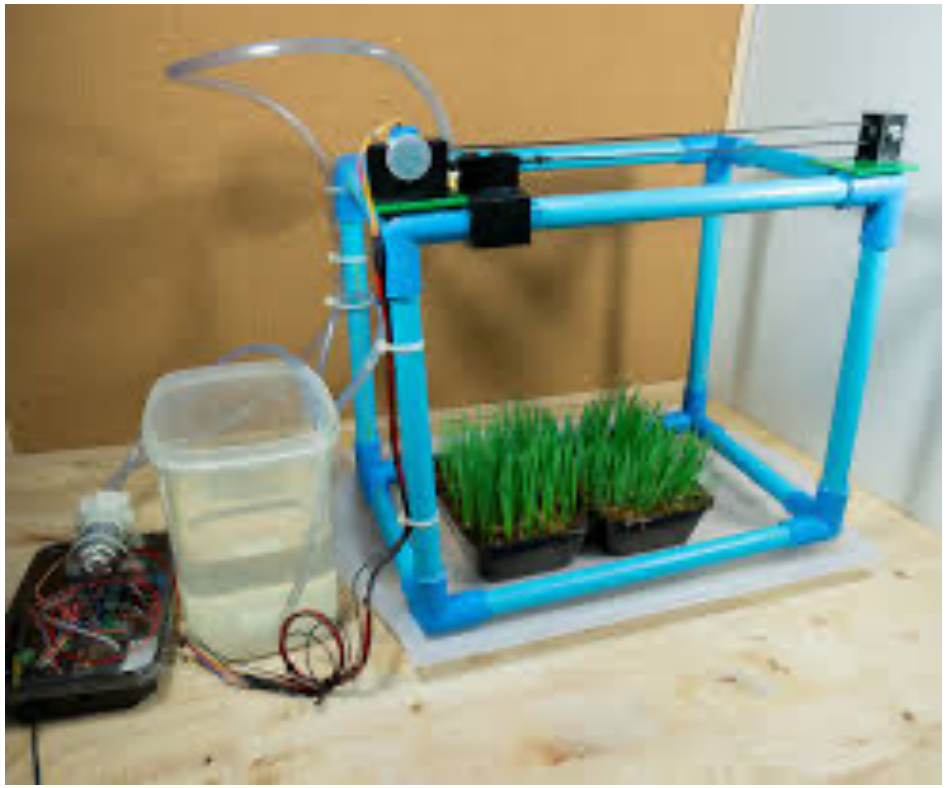


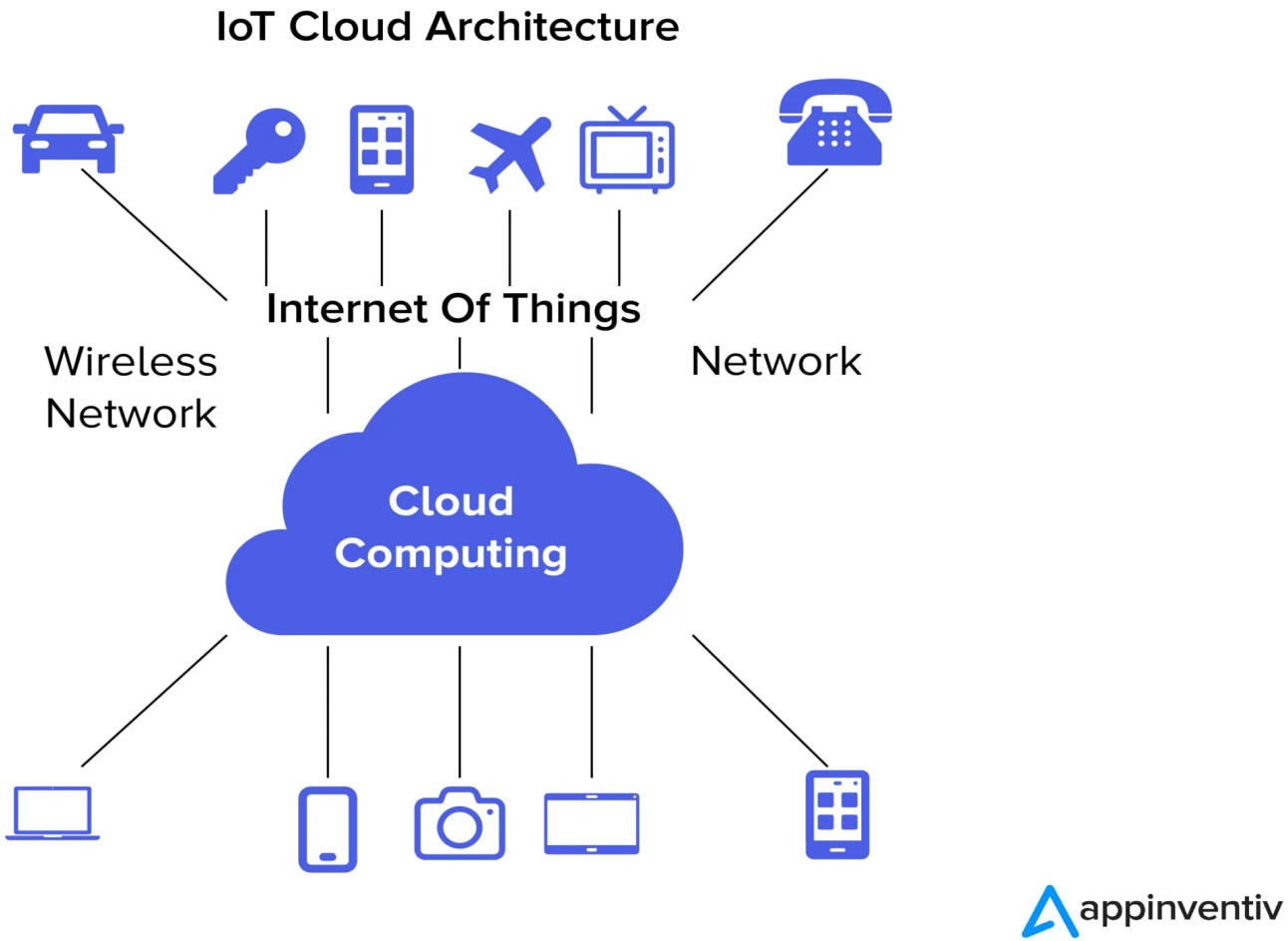

**Plant Watering System**

- It senses the moisture level When it is below a threshold it starts the water pump to water the plant through sprinkler
- When the water level reaches a maximum value
- It stops watering by switching off the pump

**RTC**: Real Time Clock
**LCD**: Liquid Crystal Display

# Real world: Plant watering system






Each devices must be able to communication through cloud and ideally would like that communication


# MQTT

## MESSAGE QUEUE TELEMETRY TRANSPORT

- Invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom Control Systems).
- They needed a reliable protocol for minimal battery loss and minimal bandwidth to connect with oil pipelines via satellite.
- The protocol evolved to be used today for Internet of Things (IoT)

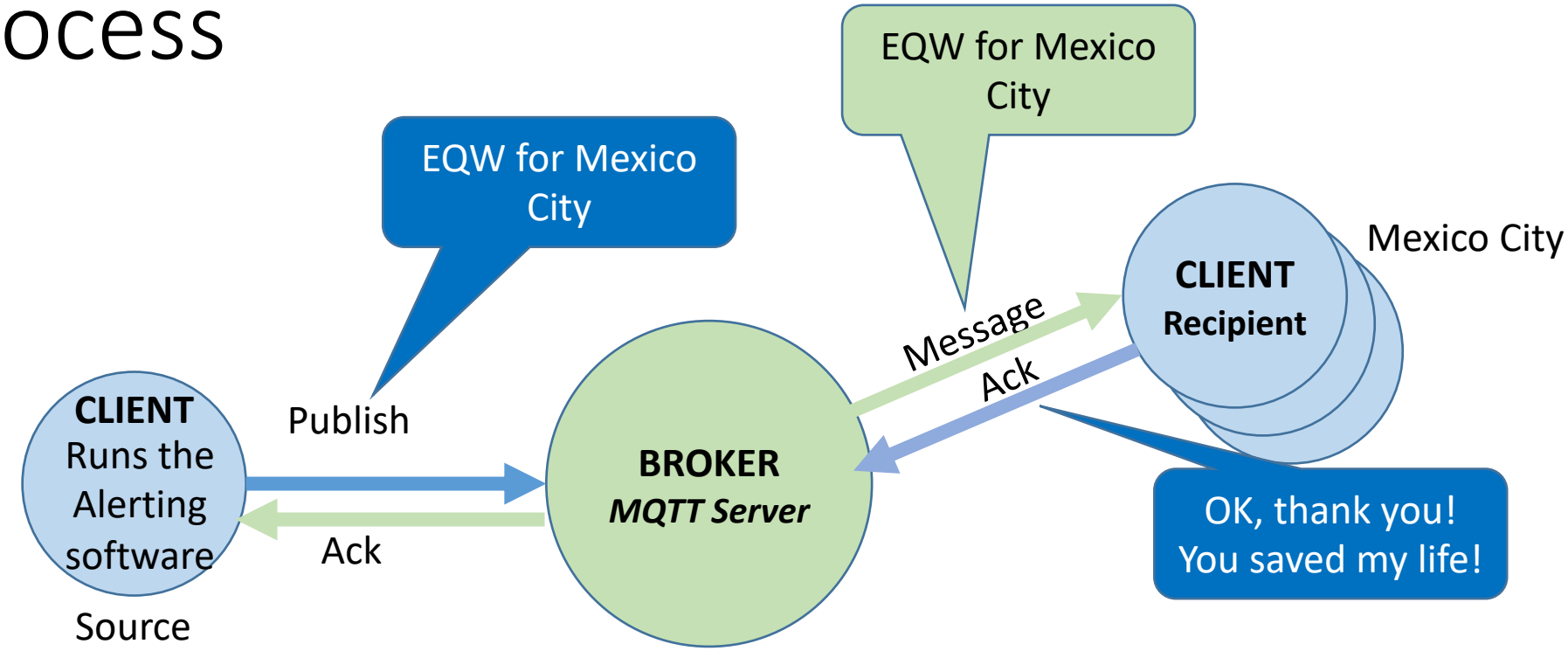
# MQTT is a messaging transport protocol

Protocol for use in constrained environments that require:

- A small code footprint to save battery
- The network bandwidth is at a premium and with unpredictable quality
- Handling intermittent connectivity
- High quality of Service data delivery
- Continuous session awareness
- Data payload can be of any type

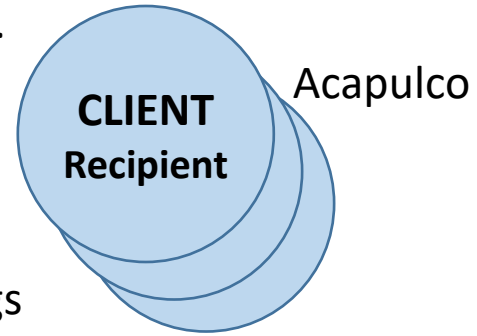
# MQTT Process

- **Broker (server)**
- **Client**
- **Topic**
- **Publish**

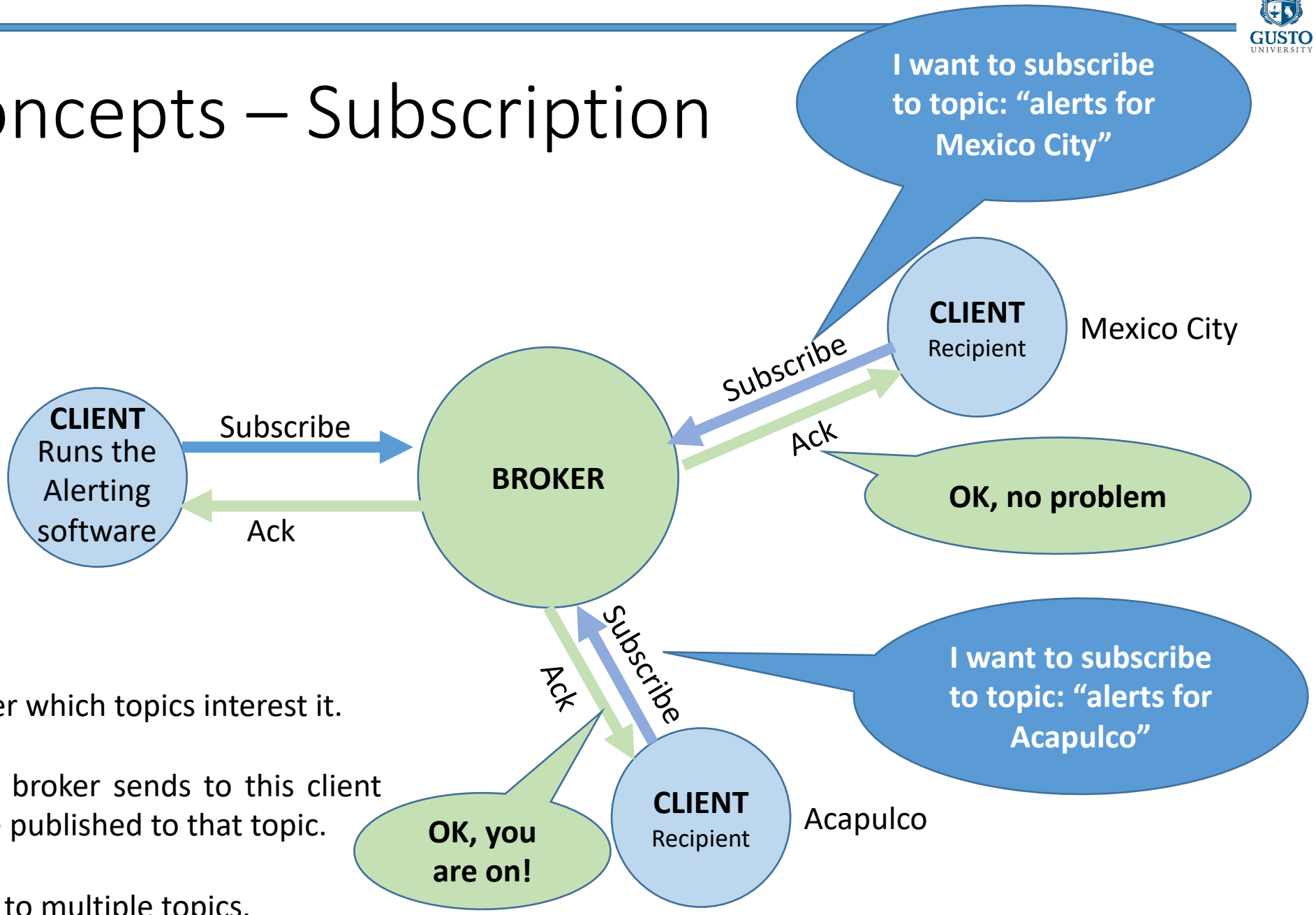



**Broker**: accepts messages from clients and then delivers them to any interested clients.
**Client**: A "device" that either publishes a message, subscribe, or both.
**Publish**: sending a message to the broker.

**Push based**: no need to continuously look for updates
**One-to-many** architecture



# MQTT Concepts – Subscription

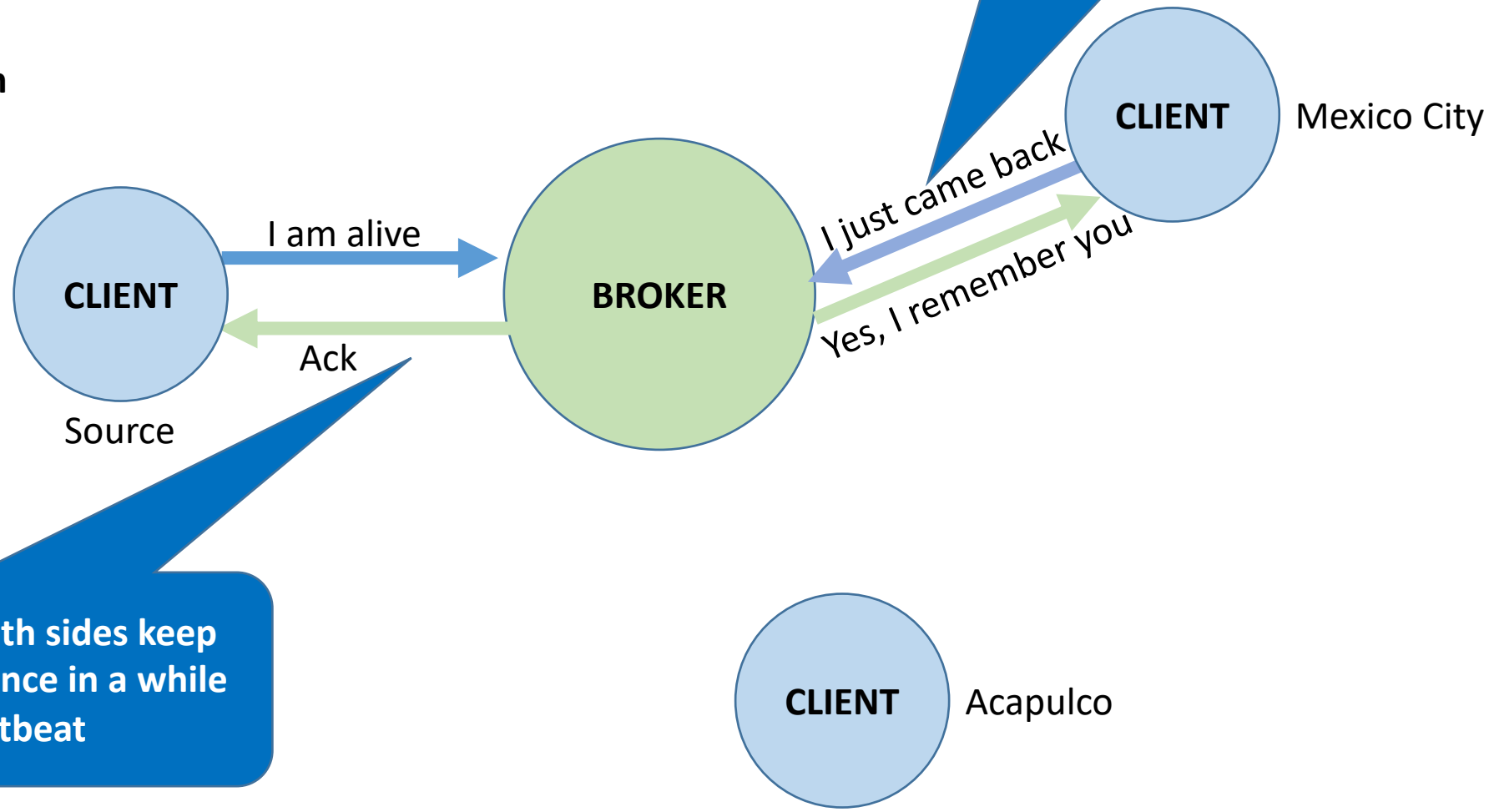



**Subscribe**:

- A client tells the broker which topics interest it.
- Once subscribed, the broker sends to this client the messages that are published to that topic.
- A client can subscribe to multiple topics.

# MQTT Concepts - Connection

- **MQTT Connection**
- **Persistent session**



# MQTT Concepts – Quality of Service

Quality of Services (QoS)

- **at most once (qos 0)**: best effort delivery (fire and forget).
- **at least once (qos 1)**: a message will be delivered at least once. But the message can also be delivered more than once (requires 2 messages).
- **exactly once (qos 2)**: guarantees that each message is received only once by the counterpart (requires 4 messages)

Each subscriber could have a different level of QoS

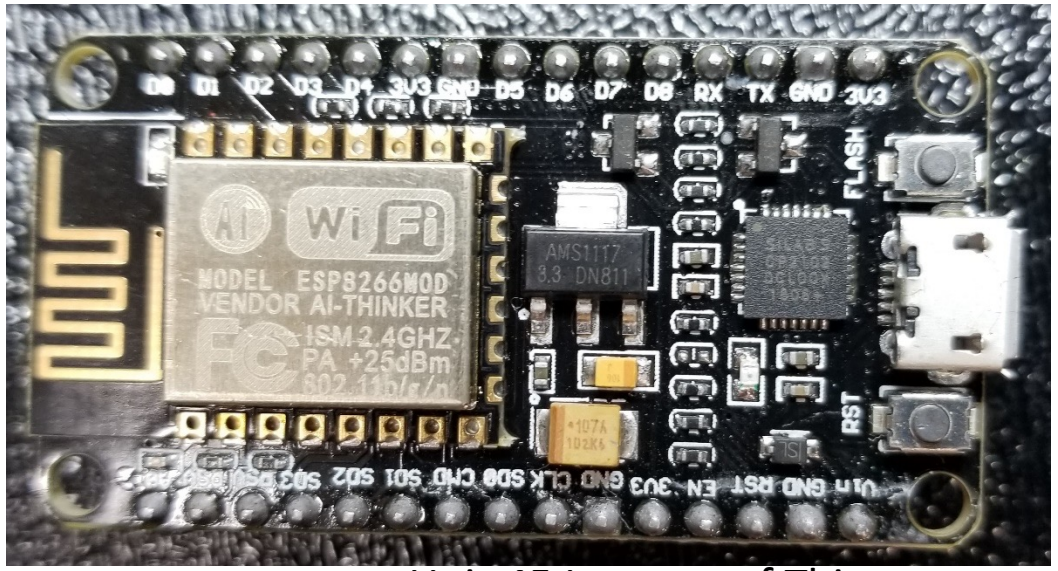
# MQTT vs HTTP comparison

| Features<br><br>Full Name | MQTT<br><br>Message Queue Telemetry Transport | HTTP<br><br>Hyper Text Transfer Protocol |
|---|---|---|
| Design Methodology | The protocol is **data** centric. | The protocol is **document** centric. |
| Architecture | It has **publish/subscribe** architecture. Here devices can publish any topics and can also subscribe for any topics for any updates. | It has **request/response** architecture. |
| Complexity | simple | more complex |
| Data security | **YES** (Payload Encrypted) | **NO**, hence HTTPS is used to provide data security. |
| message size | small, it is binary with 2Byte header. | Large, it is in ASCII format. |
| Service levels | 3 | 1 |
| Data distribution | 1 to many | one to one only |

# Use MQTT

- Include in your CAP design project MQTT:
  - Get one of the many free servers, or start by using Amazon free server
  - You can “play” with a simple end-user receiver using available hardware and simple programming:

    **ESP8266MOD – a Wi-Fi MQTT receiver on a chip**



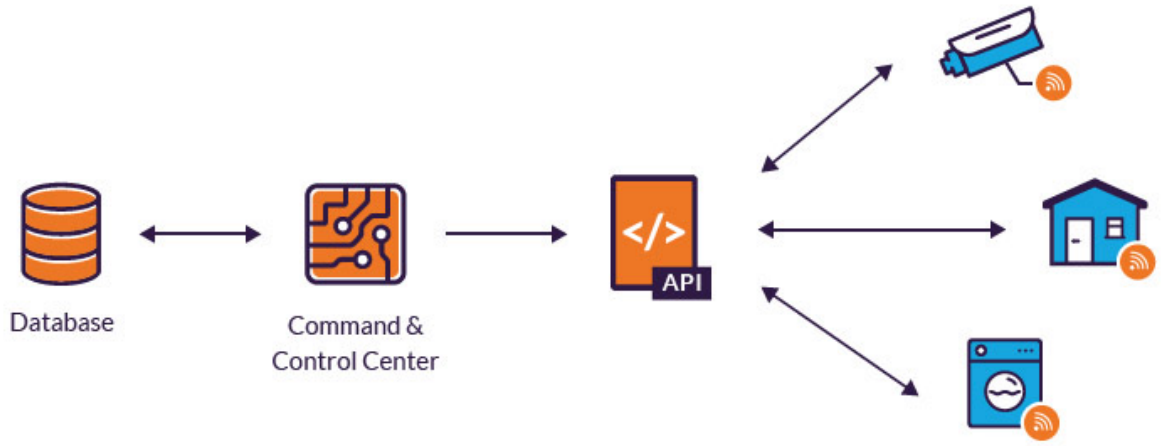
# IoT communication APIs:

- Mainly two types of communication APIs are used in IoT. Those are as follows
- **REST** based communication API
- **WebSocket** based communication API REST based communication API:

1. REST: Representational State Transfer.
2. It helps to design web services and web APIs that focus on a system's resource and how resource states are addressed and transferred.
3. It follows Request-response communication model and unidirectional communication for request. The clients send request to URIs using methods defined by the HTTP protocols (GET, PUT, POST, DELETE).
4. RESTful web service is a "web API" implemented using HTTP and REST principle. RESTful web service is a collection of resources which are represented by URIs.
5. RESTful Web services can support various internet media types (JSON, XML). JSON: Java script object notation (most popular web service). XML: Extensible mark up language .

# Management & Security

- Various management functions to govern the IoT system.
- It secures the IoT system by providing authentication, authorization, message and content integrity and data security.



- **End of Lecture**

# References:

Arshdeep, B. (2014) Internet of Things: A Hands on Approach. 1st Ed. VPT.
https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot
https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/
https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html?dtid=osscdc000283
https://www.powershow.com/view0/719aad-MmU3M/What_Is_Internet_of_Things_powerpoint_ppt_presentation
https://www.slideshare.net/jaswindersinghthind/a-basic-ppt-on-internet-of-thingsiot