Smart Water Management

IOT Sensors:

Sure, here's a simplified explanation:

1. **Choose the Right Sensors**:

   - Pick sensors (like special water meters) that can tell you how much water is being used.

2. **Put Sensors in Place**:

   - Install these sensors in public areas like bathrooms, parks, or water pipes.

3. **Connect to the Internet**:

   - Connect the sensors to the internet, so they can send data.

4. **Collect and Send Data**:

   - Program the sensors to send information about water use to a computer on the internet. They do this regularly, like sending messages.

5. **Store and Analyze Data**:

- The computer collects and stores the data. It also looks at this data to understand how much water people are using.

6. **See the Information**:

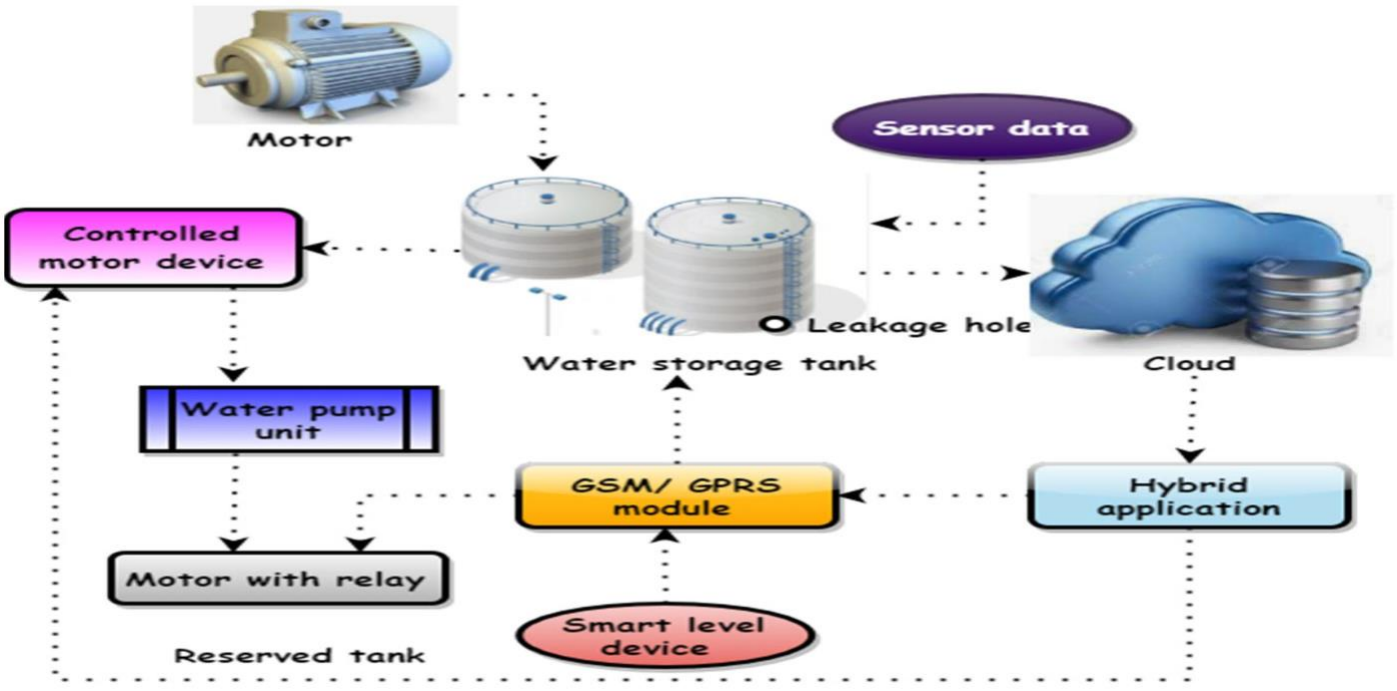
- You can see this data on a computer screen. It tells you how much water is used in public places, like how much water people are using in a park.

7. **Use the Data**:

- You can use this information to find out if too much water is being used or if there are any problems. It helps save water and money.

8. **Keep Everything Running**:

Make sure the sensors and the computer always work properly. Fix them if they have problems.This way, you can keep track of how much water is used in public places and make sure



- it's used wisely.

Python code:

Install required libraries:

Pip install paho-mqtt

Here's a basic Python script to send water consumption data to an MQTT broker, which is a common method for IoT data transfer:

```python
import paho.mqtt.client as mqtt
import json
import random
import time

# Define MQTT broker settings
broker_address = "mqtt.example.com"  # Replace with your MQTT broker address
port = 1883  # Default MQTT port
topic = "water_consumption_data"  # Topic to publish data

# Simulated sensor data
sensor_id = "sensor001"
location = "kitchen"
```

```
while True:
    # Simulate water consumption data (replace this with actual sensor data)
    water_consumption = random.randint(1, 10)

    # Create a JSON payload
    data = {
        "sensor_id": sensor_id,
        "location": location,
        "water_consumption": water_consumption
    }

    # Connect to the MQTT broker
    client = mqtt.Client()
    client.connect(broker_address, port)

    # Publish data to the topic
```

```
client.publish(topic, json.dumps(data))


# Disconnect from the broker

client.disconnect()


# Send data every 5 seconds (adjust as needed)

time.sleep(5)
```

In this script, the MQTT broker address, port, and topic should be replaced with the actual values for your environment.