



Environmental monitoring

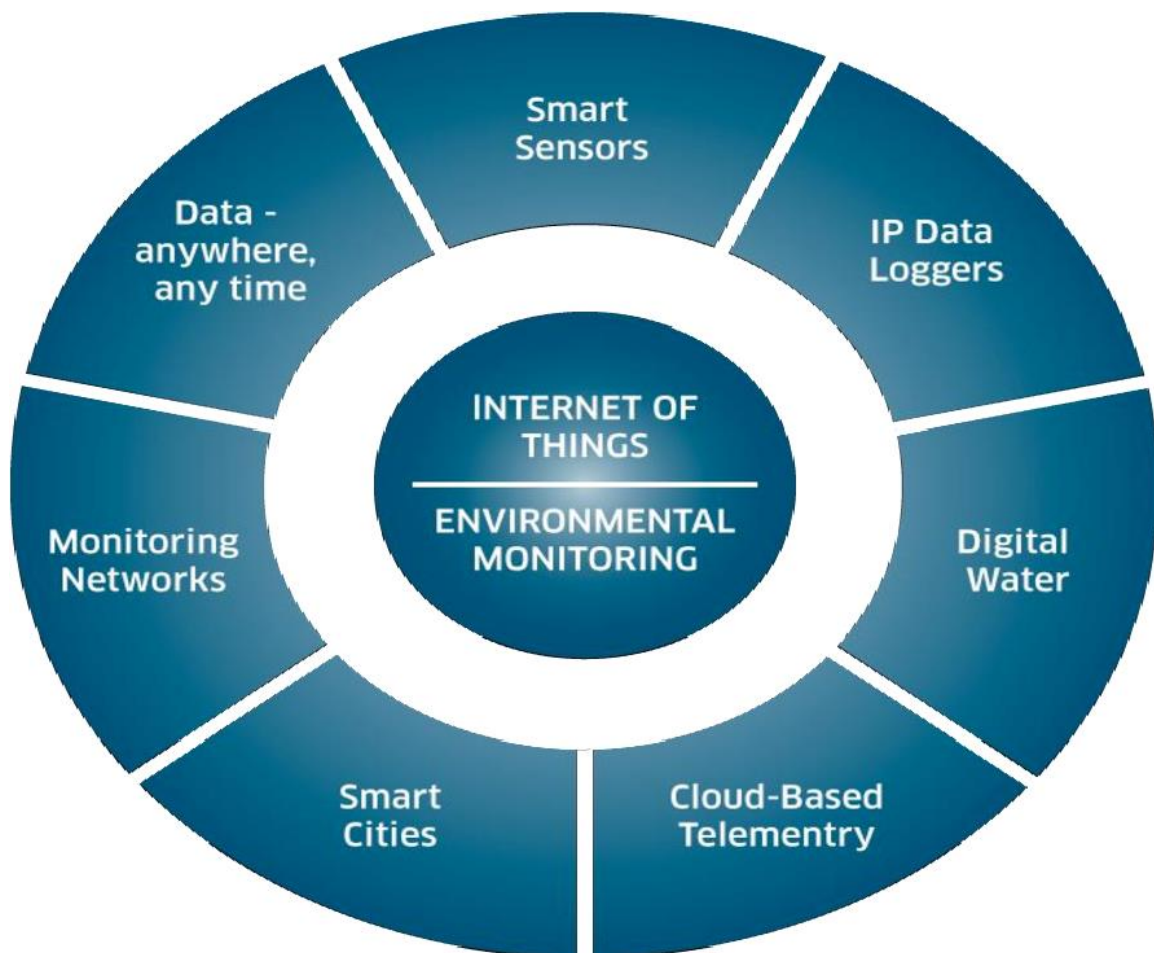
IOT PROJECT PHASE-3

IoT environmental monitoring:

- ❖ process that uses Internet of Things (IoT) technology to collect data about the environment, such as air quality, temperature, and humidity levels.

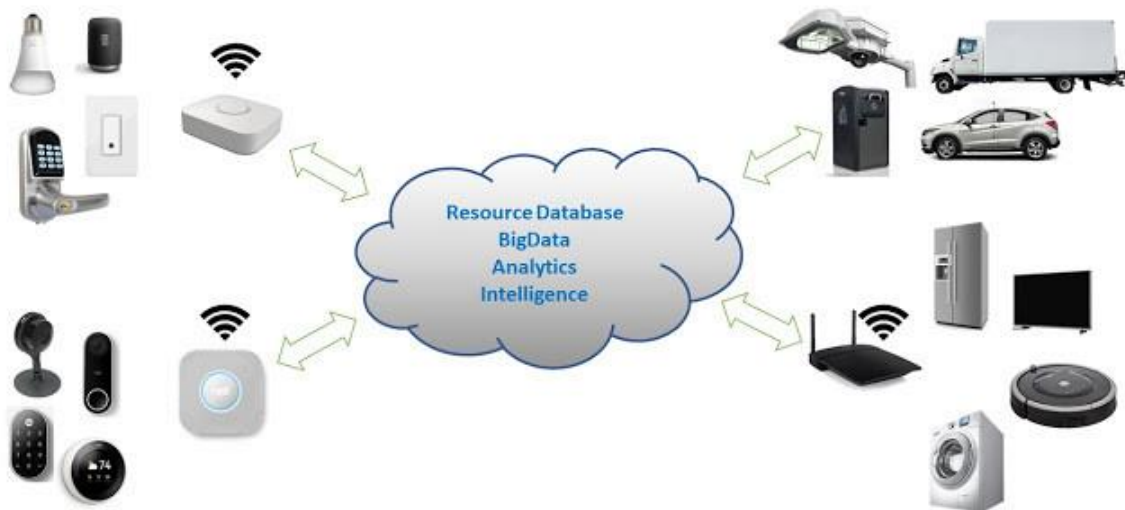
IOT DEVICE USED IN ENVIRONMENTAL MONITORING:

- ❖ IoT devices, such as sensors and cameras, gather data on the environment, including air quality, water quality and soil moisture. This data is then transmitted wirelessly to a central database or cloud platform for analysis.



❖ IOT DEVICE DEPLOYING MODEL:

In this mode of deployment IoT devices (Things) in a WPAN (Wireless Personal Area Network) are connected to a gateway through short range connectivity protocols. And the gateway device is connected to cloud through internet or LPWAN.



IOT DEVICES USED TO APPLICATIONS IN IOT

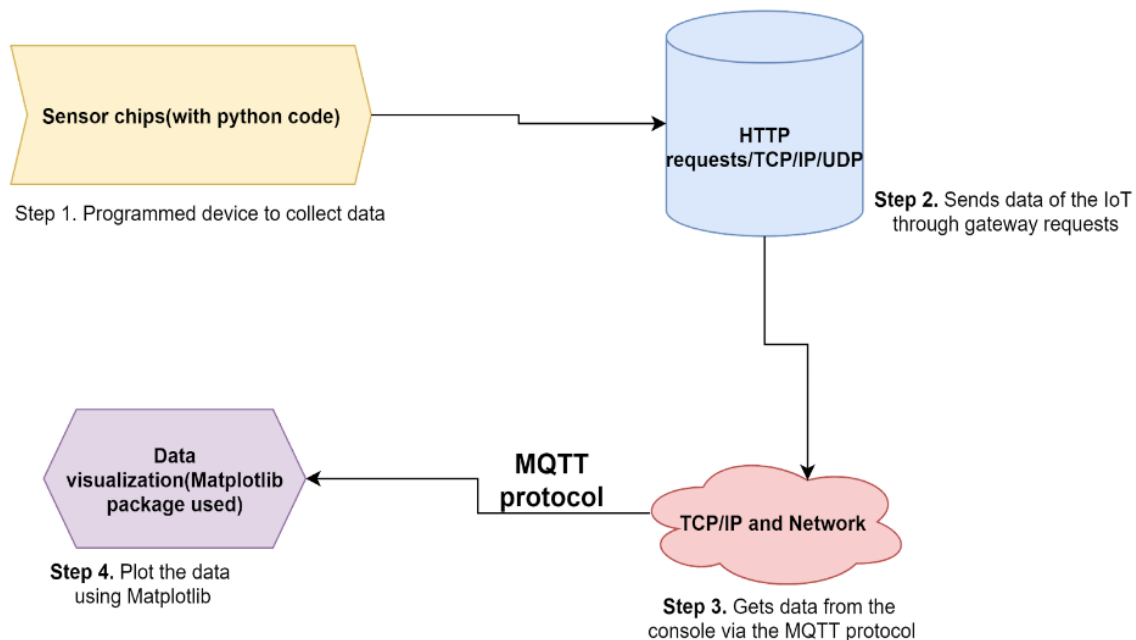
- ✓ Raspberry Pi Model 3
- ✓ Intel Edison
- ✓ Arduino

IOT SENSORS SIMULATORS USED IN PYTHON

PROGRAMMING INCLUDE:

MQ TELEMETRY TRANSPORT (MQTT) SENSOR SIMULATOR

- MQTT protocol for the IoT in Python enables high-speed data exchange with low payload communication between the devices. User-friendly requests of MQTT are made directly in Python. Data is collected in real-time and easily analyzed in mathematical computation libraries like matplotlib. The diagram below shows the steps used for the data flow:



Data logging using MQTT (install using **pip install paho-mqtt**) Python is displayed below:

```

import paho. mqtt.client as mqtt
#Callback
for received data from server
def on_connect(data_iot, user, events):
    print("connected with code" + str(events))
data = mqtt.Client()
Data.on_connect = on connect
Data.on_message = on_message
data. loop forever ()

```

COUNTLY IOT RASPBERRY PI SDK

- ❖ Sending data and visualizing data on a dashboard is simplified by involving the Countly IoT Pi SDK, which relies on internet connectivity for efficient and effective data insights from the device.

The code below is used to start the process of collecting data using Countly IoT Pi SDK in Python. Install by running **pip install Raspberry_SDK**:

```

from Raspberry_SDK.Countly
import Countly
#intiate the SDK
Countly = Countly("SERVER_URL", "APP_KEY", 0)
#Send an event
countly.event("NAME", VALUE)

```