IIHT

IoT Use Case

Contents

1. Top 10 Most adopted IoT Use Cases 2

2. Background 2

3. Problem Statement 2

4. Proposed Solution 2

5. Schema of Air Quality Monitoring System (AQMS) 3

6. Proposed App Architecture 4

7. Application Usage Workflows / Bare-Bones Architecture 5

8. Suggested Implementation of Bare-bones Architecture 6

9. Rubrics/Expected Deliverables / Evaluation Basis 7

Use Rest Framework As Necessary 7

Database 8

Analytics & Reporting 8

Java / C# / ASP .NET, Angular / React: 8

Log/ Monitoring: 8

Debugging & Troubleshooting 8

10. Reference Architectures – Reference Only 9

# Top 10 Most adopted IoT Use Cases

Remote Asset Monitoring(Read-Only)

IoT-based process automation

Remote asset monitoring and control (Read / Write)

Vehicle Fleet Management

Location Tracking

IoT for Asset/Plant Performance Optimization

IoT-based Quality Control and Management

IoT-based Good Conditions Monitoring in Transit

Predictive Maintenance

On-site Track and Trace

# Background

A Software IT Company is planning to restart operations from it Corporate Office, asking the employees to return to work at the office.

# Problem Statement

From the recent learning of the Pandemic, detecting levels of pollutants is crucial to make Work-From-Office safe, secure and possible.

It is decided to implement a continuous air quality monitoring system to make Work-From-Office safe and secure for all the employees.

Additionally, it is decided to implement an alternate day Work-From-Office Schedule for all the employees, and the continuous air quality monitoring system would provision air quality index to each individual’s desktop and to the Smartphone present within the Office premises.

The continuous air quality monitoring sensor systems are to be installed for each floor; the company is spread across three floors. The building is 3 floors. The air quality data from the sensors is transmitted to a Cloud based web-server.

The streaming sensor data is stored on Cloud based web-server. Air Quality data streamed from the sensors are continuously monitored and when the air quality level exceeds threshold values, the alerts and preventive actions are to be communicated to the individual’s in the office premises to their desktop and Smartphone

In the first phase of implementation, the alerts and preventive actions are only needed for the Company Employees; Guests are not permitted to the Work Place and need not receive alerts or preventive actions.

In the second phase of implementation, Guests are installing the Mobile App and update if necessary when they sign-in at the front-desk.

# Proposed Solution

1. Design an air quality monitoring system that can cover each floor of the office building
2. Monitor air quality index through sensors installed in each floor
3. Provide alerts and preventive actions to employees smartphones when the air quality level exceeds threshold values
4. Provision to be made where alerts to be sent when pollutant level is out of range and users can access the application to view the status anytime.
   1. to **ensure full coverage of each floor span,**
   2. **Air Quality Sensors to be installed adjacent to every existing node of overhead Smoke Detecting Sensors**
   3. **Additionally Air Quality Sensors to be installed in Smoking Zones on each floor**
5. **The sensors shall be connected to a microcontroller to control the sensors network**
6. **Data collected by microcontroller is transmitted to the cloud based web-server for storage**
7. **Application on cloud based web-server Analysis data**
8. **The whole system is to be fully integrated and automated with no need for manual intervention and independent of manual monitoring**

# Schema of Air Quality Monitoring System (AQMS)

**“AQMS - Air Quality Monitoring System”**

* **The Software Product would be called, Air Quality Monitoring System, AQMS**
* **AQMS to Wireless sensors placed at strategic locations, sense the level of dust particles, carbon dioxide, carbon monoxide, nitrogen dioxide and sculpture dioxide in the air**
* **Sensor data is continuously ‘streamed’ to a cloud database**
* **In the cloud, the data is analyzed to provide information on air quality**
* **Analyzed Air Quality Data is stored to a backend server for posterity. Data to be backed-up and be readily accessible for period of 3 years in on-premise data server.**
  + **After period of three years, data to be retained at central repository off-premises**
  + **This allows relevant authority to take remedial action and community to take precautionary measures and improve existing AQMS**
* **Alerts, Preventive actions based on Analysis of the air quality, is shared through a Android-based Smartphone app**
* **Major Components of the AQMS**
  + **Sensor Device which ‘streams’ air quality data to a Cloud**
  + **Software program on Cloud server to access the sensor data and analyze Air Quality**
  + **Threshold Parameters of air quality level are periodic monitored**
* **Other Components of AQMS (NOT IN SCOPE OF THE DEVELOPMENT PROJECTS)**
  + **These following components, are support mechanism to ensure AQMS is successful in achieving the intent of Safe Guarding the People within the Office Premises**
    - **All employees to own a Android Smartphone**
    - **All stair ways must be clear of any hindrance**
    - **Red Beacon Alarm system to be installed on all floors and are adequate to maintain line-of-sight from any point on the floor**

# Proposed App Architecture

Microcontroller - Device to detect Air Pollutants like CO2, CO, and other pollutants

Sensor Nodes - Linked to IoT gateway (example, using ZigBee protocol)

IoT Gateway - Streams sensor data to a Cloud Database

IoT gateway is a centralized hub to connect IoT devices and Sensors to cloud-based computing and data processing. IoT gateways allow bidirectional data flow between the cloud and IoT devices

**Gateway (Simulator)** should send data to Azure IoT Hub/ AWS IoT Core

One must register the IoT Gateway in either Azure IoT Hob or in AWS IoT Core.

**Azure IoT Hub / AWS IoT Core**

**Azure IoT Hub** provides cloud-hosted solution back end to connect any device. Microsoft Azure enables native support for SQL Server database files stored as blobs with SQL Server Data Files; allows creating a database in SQL Server running in on-premises or in virtual machine in Microsoft Azure

**AWS IoT Core** is a managed cloud service to let connected devices interact with cloud applications and other devices, and can process and route messages to AWS endpoints and to other devices. Amazon RDS supports native backup and restore for SQL Server databases by using differential and full backup files

**IoT Gateway - Sensor mapper**

One should prefer Gateway - Streams sensor data to a Cloud Database as a Best

Practice

AWS Dynamo DB is recommended and Azure Cosmos DB is recommended to

Store Gateway sensor mapping details.

AQMS -

* Air Quality Monitoring App on Cloud Server constantly accesses Cloud Database
* AQMS analyses data
* AQMS transmit Alert and Preventive Action to Every Desktop and Mobile Device

For first phase of the Project, the following Integration Architecture is proposed

“Sensor Nodes” linked to “IoT gateway” using “ZigBee” protocol

“IoT Gateway” forwards “sensor data” to “Cloud based Database Server”

“Air Quality Monitoring System” will access Cloud Database

“Air Quality Monitoring System” analyses “sensor data”

When “Threshold Parameters” is crossed, “AQMS” transmits to “Desktop”

When “Threshold Parameters” is crossed, “AQMS” transmits to “Mobile App”

# Application Usage Workflows / Bare-Bones Architecture

For first phase of Project, following Development Design is proposed in Case 1 and Case 2

**Case 1**

Sensing Parameters => Microcontrollers => Gateway Simulator =>Azure IoT / AWS IoT (for Cloud, Database, Sensor Mapping to floor) => AQMS Analyzer => Dashboard ( Web Application

Dashboard: Desktop + Android App - Displays Alerts and Preventive Action

Working:

* Sensing parameters to be defined
* Microcontrollers are programmed to take sensors as input and transmit data to Cloud
* An application is developed to access the data and analyse the data in real-time
* The application sends alerts and preventive actions to each desktop in office premise and to individual smartphones

The suggested tool chains include,

1. Microcontrollers to detect pollutants
   1. ***Implement or Create Mocks or Use Any source to simulate streaming of Sensed Data***
2. IoT Gateway to connect IoT devices and sensors to cloud-based computing and data processing
   1. ***Incorporate Azure IoT Hub / AWS IoT Core to connect to devices (example, Sensor, Mobile, Desktop, Red Alert Beacon) “of-all-three-floor”***
3. Cloud Database to store data - MySQL,SQLite, [Dynamo DB or Cosmos DB]
   1. ***Implement Database of Cloud provider or Create SQL Database to store Data received from Data Simulator (Point 1.1 above)***
   2. ***Implement Data Persistence***
4. GUI - Angular or Dot Net C#, ASP.Net, Java, Python
   1. ***Implement Dashboard***

# Suggested Implementation of Bare-bones Architecture

You are free to design the implementation; below is merely a guided suggestion for implementation and visualization and RAD / Prototype

Given the workflows, suggested design of AQMS Project Version 0.1

* 1. Thresholds for each Sensing Parameters to be defined as “Protected Values”
  2. Worksheet - Excel or CSV file to provide Sensing data
     + Each column has specific Sensing Parameters: 02, CO2, SO2, CO, C
     + Random function to generate continuous Sensing data
  3. Data Persistence - Persist the incoming data in database against each floor based on sensor to floor mapping
  4. .Data Analytics and Alerts: Generate alerts based on data present in database and stored it in database. Create APIs to retrieve the alerts and show it on web dashboards.Microcontrollers - Implement or Create Mocks or Use to Any source to simulate streaming of Sensed Data
  5. IoT Gateway - Incorporate Azure IoT Hub / AWS IoT Core to connect to devices (example, Sensor, Mobile, Desktop, Red Alert Beacon) “of-all-three-floor”
  6. Cloud Database - MySQL,SQLite, Implement Database of Cloud provider or Create SQL Database to store Data received from Data Simulator
  7. Data Persistence -
  8. GUI - Angular or Dot Net C#, ASP.Net, Java, Python -
  + Build Desktop Application - Features to build in Desktop Application
    - Build Angular / ReactJS Dashboard to show
      * Device Status (is\_Transmitting = True / False, per device id, per floor),
      * trends of alerts, alerts per floor, and, cumulative alert of three floors
      * visual interface to access available device logs of IoT Gateway, Microcontrollers
      * Alerts and Preventive Action
  + Build Analyzer to continuously monitor sensor data if it crosses threshold parameters
  + [Optional]Provide data of user profiles of active and connected desktops [and mobiles (mobile number)] by each floor

# Rubrics/Expected Deliverables / Evaluation Basis

## Use Rest Framework As Necessary

1. Implement HTTP methods like GET, POST, PUT, DELETE, PATCH to implement RESTful resources:
2. Use constructor-based dependency injection in few classes and setter-based dependency injection in few classes.
3. Follow Proper naming Conventions

## Database

1. Build Views to display on Dashboard

## Analytics & Reporting

One can leverage Azure Power BI tool for analytics & reporting or AWS QuickSight

## Java / C# / ASP .NET, Angular / React:

1. Use MVC development style to implement the Code
2. Generate project documentation, and share it as a part of deliverables
3. Implement using proper SOLID design principles
4. Use angular directives in the UI layer where necessary
5. Build Tests using Nunit / Junit / TestNG
6. Build Mock(s) / Stub(s) as place holders for components not yet designed, and replace them as you progress in End-to-End implementation, example for Android OS based Mobile App.
   1. NOTE - Time Permitting
      1. Build a mini Android App to receive a simple text message from AQMS.
         1. Use [Messages from Pc To Android Device](https://messages.google.com), or
         2. Use [Microsoft Phone Link as the PC app experience, and Link to Windows as our mobile app for Android devices](https://support.microsoft.com/en-us/topic/introducing-microsoft-phone-link-and-link-to-windows-2e4bb4c0-f99a-4464-92a8-5264c7c39734)

## Log/ Monitoring:

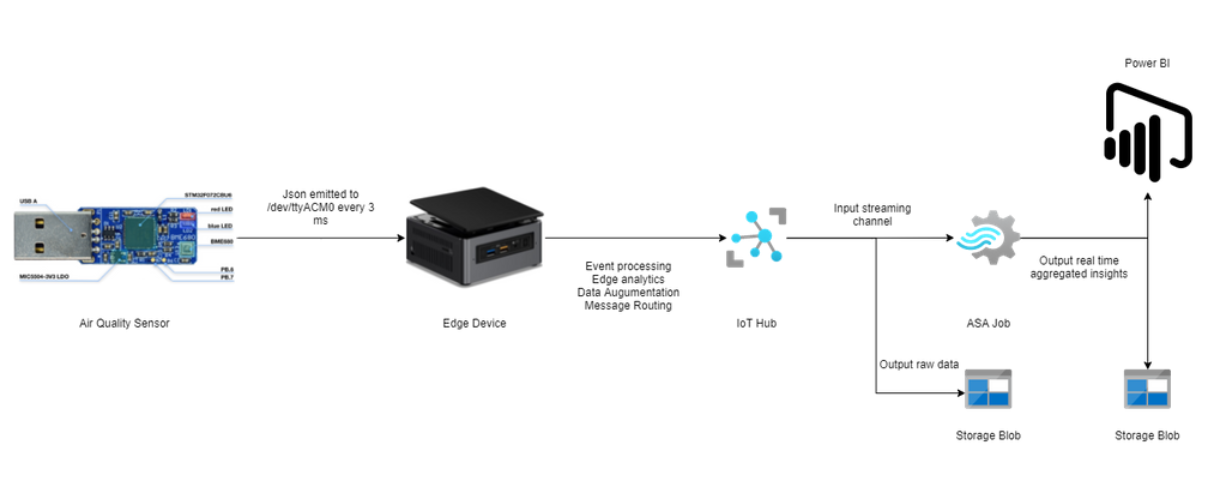
1. Integrate CI CD Pipeline to your application using
   1. Azure DevOps or
   2. AWS CodePipeline
2. Create or document scripts for auto execution of tests on a significant modification
3. Integrate AWS or Azure IoT APIs with Angular or React

## Debugging & Troubleshooting

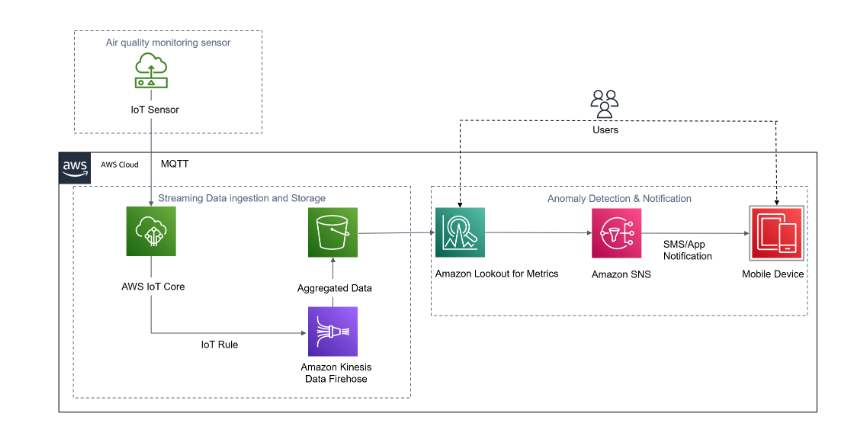
1. Generate bug report & error logs - Report must be linked with final deliverables which should also suggest the resolution or the encountered bugs and errors.
2. IoT based application, integration and debugging end points needs special attention and may be specialized IDE, Browser and Simulator APIs to capture errors and signals
3. Perform unit and integration testing for the front end application
4. Perform e2e testing of the UI enabled with IoT end points before Cloud Deployment
5. Ensure to enable required AWS or Azure to support Angular or React based frontend deployment with IoT
6. Ensure proper interfacing of Sensor Data to AQMS
7. Ensure proper interfacing of Desktop and Mobile with AQMS

# Reference Architectures – Reference Only

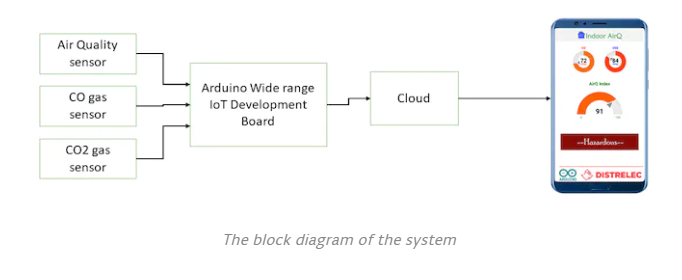
Using Azure IoT Hub - The following is schematic reference architecture for AQMS on Azure



Using Azure IoT Hub - The following is schematic reference architecture for AQMS on AWS



Using Arduino IoT Board - The following is schematic reference architecture for AQMS on prem and integration with either Cloud [AWS or Azure]



# Learning Flow – Component wise

**Component -3**

**Objective**

1. Integrate AWS or Azure IoT APIs with Angular or React
2. Perform all the integrations as defined in point [9 (13-2](#Point9)0)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COMPETENCY** | **SKILL** | **Track** | **Skill Details** | **Course Name** |
| **Component 3 - Azure** | | | | |
|  | IOT Services | Azure Dotnet | Azure Iot hub | Introduction to Microsoft Azure IoT |
|  | Azure Dotnet | Azure Stream Analytics | Azure MasterClass: Analyze Data With Azure Stream Analytics |
|  | Azure Dotnet | Azure Service Bus Queues | AZ-300 Azure Architecture Technologies Certification Exam |
|  | Azure Dotnet | Azure Service Bus Topics |
|  | Azure Dotnet | Azure Event Hub |
|  | Azure Dotnet | Azure SQL Database | Azure SQL Database - SD2 |
|  | Azure Dotnet | Azure Cosmos DB | Azure Cosmos DB - SD2 |
|  | Azure Dotnet | Azure RedisCache | Azure RedisCache - SD2 |
| Storage | Azure Dotnet | Azure Blob Storage | Azure Blob Storage - SD2 |
| Identity and Compliance | Azure Dotnet | AAD (app registration, SSO integration) | Azure Active Directory & SSO - Fundamentals |
| Content Delivery | Azure Dotnet | Azure API Gateway | Azure API Gateway - Fundamentals |
| Networking | Azure Dotnet | Azure Virtual Network | AZ-104: Microsoft Azure Administrator Full Course |
| Management and Governance | Azure Dotnet | Azure Application Insights | Azure Application Insights-SD1 |
|  | Azure Dotnet | Azure Monitor | AZ-104: Microsoft Azure Administrator Full Course |