Project Design Phase-II

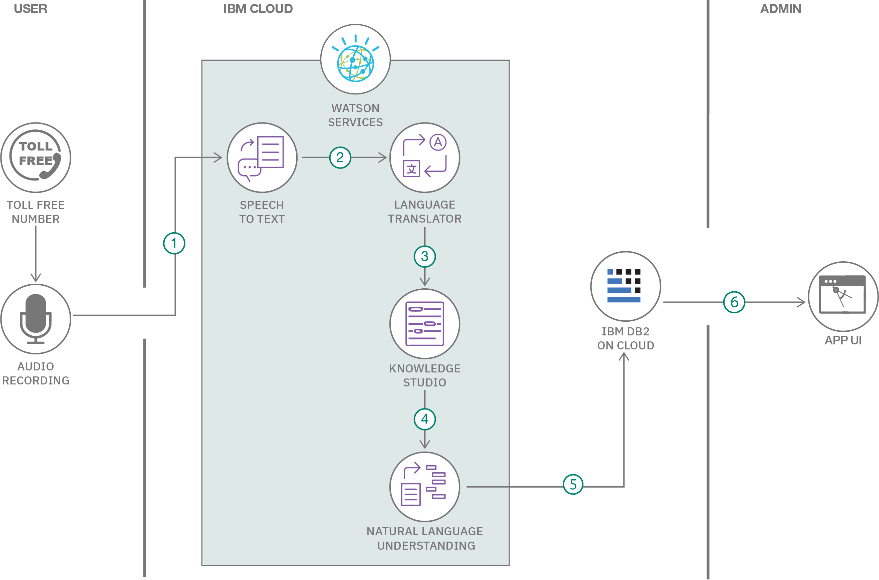
Technology Architecture

|  |  |
| --- | --- |
| Date | 03 October 2022 |
| Team ID | PNT2022TMID08726 |
| Project Name | Smart Farmer - IoT Enabled Smart Farming Application |
| Maximum Marks | 4 Marks |

# Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

# Example: Order processing during pandemics for offline mode



Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API’s etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Characteristics** | **Description** | **Technology** |
| 1. | USER INTERFACE/USER DISPLAY | Hardware Output display to the user by means of Web UI and LCD Display | Embedded C++ |
| 2. | (Application logic-1) Connection of Hardware between ESP8266 with required sensor. | Integrating the sensors along with the ESP8266 and Node Red. | Arduino IDE, Cloud |
| 3. | (Application Logic-2) | Connecting Hardware Applications with Internet of Things through IBM cloud | Node Red |
| 4. | Server side Logic mechanism | Integrating with the Webhooks. (e.g.) Select if the alert to be sent  which condition exist or does not exist in the case | IBM DB2,IBM Watson STT  service |
| 5. | Integrating with the IBM cloud Monitoring | Configuring monitoring instance detail. Specifying the API Key with the function call. | CRUD operation, JSON file format , API function call |
| 6. | Notification Sending application | Communication through cloud notifications. | IBM Cloudant DB, Node RED service |
| 7. | Buzzer indicating and LCD display | Integrating the ESP8266 with BUZZER with specified Delay mode and LCD display. | Embedded |

# Table-2: Application Characteristics:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | OPEN SOURCE FRAMEWORKS | Django, which is an open source framework under python, has been used. | Technology used is python |
| 2. | SECURITY IMPLEMENTATION | As a cloud-hosted service the IBM Watson IoT Platform service embeds security as an important aspect of its architecture | IBM Watson |
| 3. | SCALABLE ARCHITECTURE | The browser-based GUI and REST APIs are fronted by HTTPS, so it can trust that they are connecting to the genuine Platform Service. Access to the web-based GUI is authenticated by your IBMid. Using the REST API requires an API key, generated through the GUI, can use this to make  authenticated REST API calls against the organization. |  |
| 4. | AVAILABILITY | Using ESP8266 Wi-Fi module the datas are pushed to the cloud. With the help of IoT applications user can monitor continuously, if there is any critical condition user can intimate immediately. | Wi-Fi module |
| 5. | PERFORMANCE | Design consideration for the performance of the application (number of requests per sec, use of Technology used Cache, use of CDN’s) etc. |  |