International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-14, Issue-3 (June 2024)
https://doi.org/10.5281/zenodo.11467295

Implementation of MQTT Protocol for Artificial Intelligence

Suraj Khot[1] and Abhijeet Mali[2]
[1]Student, Department of Electronics & Telecommunication Engineering, Tatyasaheb Kore Institute of Engineering and Technology (An Autonomous Institute), Warananagar, Kolhapur, Maharashtra, INDIA
[2]Professor, Department of Electronics & Telecommunication Engineering, Tatyasaheb Kore Institute of Engineering and Technology (An Autonomous Institute), Warananagar, Kolhapur, Maharashtra, INDIA

[1]Corresponding Author: sdkhot.etc@gmail.com

Received: 02-05-2024 Revised: 19-05-2024 Accepted: 02-06-2024

ABSTRACT

The IoT-Based Attendance Management System is a sophisticated solution leveraging Azure services and MQTT protocol for efficient attendance tracking. The system comprises a Face Device equipped with facial recognition capabilities, which captures attendance data and communicates with Azure IoT Hub. The Azure IoT Hub, acting as the central hub, receives and processes attendance data from the face device, utilizing MQTT for real-time communication. A dedicated MQTT service, facilitated by the MOSQUITTO BROKER, subscribes to the Azure IoT Hub, ensuring seamless data flow between the face device and the broader system. The web application, hosted on Azure App Service and integrated with SQL Server, serves as the user interface. It not only interacts with the face device through Azure IoT Hub but also processes attendance data and generates comprehensive reports, enhancing the overall efficiency of attendance management through the power of IoT.

This architecture offers a scalable, secure, and real-time attendance management solution, bridging the physical and digital realms through facial recognition technology and Azure IoT services. The synergy of components, from the MQTT-enabled face device to the Azure-based web application, establishes a robust ecosystem for capturing, processing, and presenting attendance data with seamless integration and reliability.

***Keywords*--** MQTT, IoT, Mosquitto Broker, SQL, RFID, Azure IoT, AI

I. INTRODUCTION

Message Queue Telemetry Transport (MQTT) is a messaging protocol that is lightweight enough to be supported by the smallest devices, yet robust enough to ensure that important messages get to their destinations every time. With MQTT devices such as smart energy meters, cars, trains, satellite receivers, and personal healthcare devices can communicate with each other and with other systems or applications.

There is a revolution happening right now, the whole client-server concept and the way we experience internet is rapidly changing. “Internet of things” is here, “things” as diverse as smart phones, cars and household appliances to industrial strength

Sensors are being linked to each other and the internet. And while we currently only have some simple intercommunication and autonomous machine-to-machine (M2M) data transfer, the potential benefits to lifestyles and businesses are huge.

Telemetry technology allows us to measure and monitor things from a distance. By improving technology we made it possible to interconnect devices at remote places and reduce cost of building and maintaining these systems Information is more important today that any gold, black or regular, and challenges lie in getting the information from the devices to the people and applications they are using in timely manner. Information is only as good as the time received allows it to be and ability to respond accordingly only increases its value. If the devices are widely distributed geographically, or if they have limited storage or computational abilities, the challenges increase considerably, as do costs. Fortunately, these challenges are being overcome through the use of improved telemetry technologies and communication protocols that are making it possible to send and receive this information reliably over the Internet, even if the network is unsteady or the monitoring device has little processing power. MQTT provides telemetry technology to meet the information challenges of today’s internet users.

II. LITERATURE SURVEY

With respect to said work an extensive literature survey is conducted accordingly which is presented as below, Qaiser et al. (2006) proposed time attendance system using RFID technology and window services to make some facilities taking parents in account, sending of SMS, email, reports with concern. This system discussed various issues like proxy, weightage, probation analysis, submission warning and teaching time loss. [1]

T. S. Lim et al. (2009) focuses on the issue of student irregular attendance. This system used RFID technology with microcontroller and tends to implemented in school, colleges and universities. It is an automated scanning system that is represented by parallel line of different width which stores information regarding an object. The proposed system uses real time clock function to gain accurate results, and viewed using software called HyperTerminal, which is connected to computer using RS232 (USB) to store the data into the database. The system tends to be vital with physical advantages as low cost, light weight, user friendly, compact design and portable. [2]

A. Kassem et al. (2010) focuses on the factors such as reliability, time saving and easy control. This system defined prototype using RFID system. This implementation only applicable for small scale setup. The further attempt will be made to modify the RFID for completing application on a large scale. The application has three modules student faculty and administrator. [3]

M. Kassim et al. (2012) described web-based attendance for academic performance and progress uses MySQL database, focuses on time consuming and inefficient method of conventional manual attendance. It consists of three module RFID reader, data reporter and web server module and works on TCP/IP protocol. [4]

M. B. Srinidhi et al. (2015) developed system safe and secure four-tier architecture using biometric, GSM and RFID technology, records attendance of students and staff members and tracks their location in campus. The SMS and email facilities were proposed for lagging attendances for both students and parents. It has been developed with desktop and android version for remote access. System gives student's consistency graph throughout semester.[5]

Sri Madhu B. M. et al. (2017) developed attendance system to reduce time consumption by traditional attendance. It's implementation of IoT through raspberry Pi 3 and RFID technology to make automated one and developed android app for students can check their attendance anywhere.[6]

H. K. Nguyen et al. (2017) developed using RFID based on the mobile communications and IT technologies, to put into use at any gatherings such as conferences, exhibitions, training courses, etc. and scaled from small to large venues depending on the need. The system is designed to generate real time consolidated reports on attendance, which collects records and processes data on participants of any conference or event. This paper suggests to use ESP8266 for low cost WIFI chip purpose as MRF24WB0MA module were used for development, which is three times costlier. [7]

H. U. Zaman et al. (2017) presents RFID reader as established approach for recording attendance system, used SD card and the Thing speak API cloud server, as it provides real time data collection with ESP8266 WIFI module. [8]

The proposal of merging the IoT and RFID technology to establish attendance system is discussed in number of researches. There are many papers presented for making advance effective, efficient and reliable system.[9]

III. PROPOSED SYSTEM

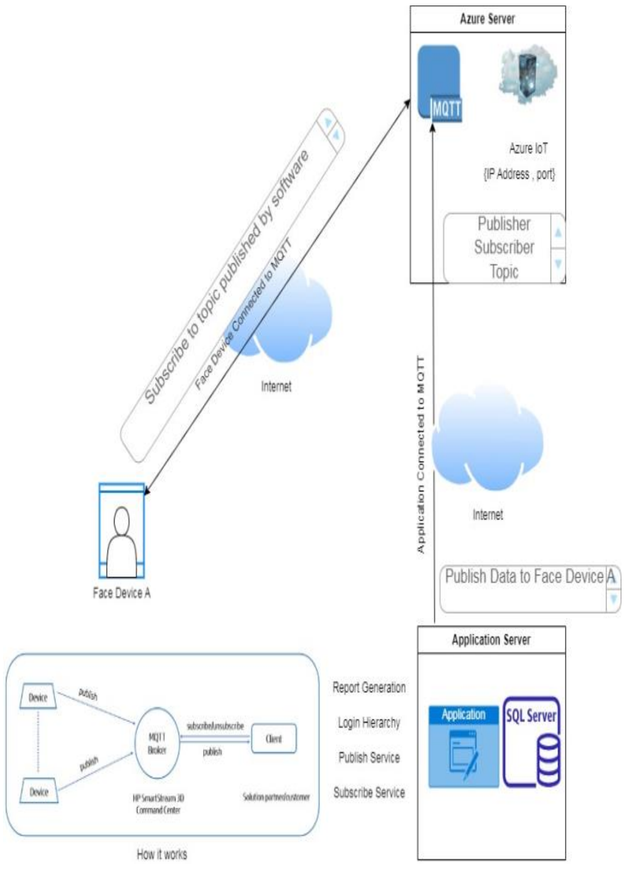

Figure 1: Block Diagram

In this system architecture, Face Data and Face Service collaborate by utilizing the MQTT protocol to publish and subscribe data. The Face Service, responsible for processing facial data, connects to an Azure IoT or Windows MQTT server, serving as a central communication hub. This server facilitates interactions not only between Face Data and Face Service but also with other potential components of the system. Simultaneously, the Face Service establishes a connection with an Application Portal, a web application developed using the Serenity platform. This portal acts as the user interface, allowing users to input data and control the Face Service. All these components are interconnected over the internet, emphasizing the importance of secure communication channels. Implementing this architecture involves configuring the Face Service for MQTT communication, developing the Application Portal, and setting up the chosen MQTT server or Azure IoT Hub. Careful consideration of security measures is crucial, particularly when handling sensitive facial recognition data, ensuring compliance with best practices for IoT and web application development.

3.1 Objectives

Automated Attendance Capture: Develop a facial recognition-enabled device capable of accurately capturing attendance data in real-time, eliminating the need for manual attendance marking.

Azure IoT Integration: Implement seamless integration with Azure IoT Hub and leverage MQTT for efficient communication between the face device and the IoT Hub, ensuring reliable data transfer.

Web Application Development: Create a user-friendly web application hosted on Azure App Service, providing a platform for users to interact with the face device, manage employee data, and access attendance reports.

Role-Based Access Control: Implement a role and scope management system to control user access and privileges within the web application, ensuring data security and privacy.

SQL Database Integration: Connect the web application to a SQL Server database for efficient data storage, retrieval, and management of employee information and attendance records.

Real-time Reporting: Enable the web application to process attendance data in real-time, generate comprehensive reports, and allow users to customize and export reports for further analysis.

Scalability and Reliability: Design the system with scalability in mind to accommodate a growing number of devices and users. Ensure high reliability and availability of the system components.

User Training and Support: Provide training materials and support resources to ensure users can effectively utilize the system, fostering user adoption and satisfaction.

3.1.2 Hardware Requirement

Face Recognition Terminal

Face Recognition Terminal is a biometric machine that uses facial recognition to identify individuals. It is perfect for use in airports, banks, and other high security areas.

12 Volt Power Supply

The voltage required is 12 Volts.

Cat 45 Lan Cable

The communication is done using TCP/IP at 10/100 Mbps.

3.1.3 Software Requirements

Face Recognition Software

Facial recognition algorithms integrated into the face device.

Compatibility with the selected 1 and operating system.

Azure IoT SDK

SDKs for the chosen programming language (e.g., C#, Python) to facilitate communication between the face device and Azure IoT Hub.

MQTT Implementation

Mosquitto broker software or another MQTT broker compatible with the chosen architecture.

Configuration for secure communication (TLS/SSL).

Web Application

Development framework (e.g., ASP.NET, Node.js) for web application development.

Compatibility with major web browsers (Chrome, Firefox, Edge, etc.).

Secure coding practices to prevent vulnerabilities.

Database Management System

SQL Server database management system.

Database connectivity libraries compatible with the web application framework.

SQL scripts for database schema and initial data setup.

Security and Authentication

Implementation of secure authentication mechanisms (e.g., OAuth, JWT) for user login.

Encryption protocols (TLS/SSL) for secure data transmission.

Access controls and permissions configured within the system.

Reporting Module

Reporting tools or libraries integrated into the web application.

Compatibility with the chosen web development framework.

Documentation and User Training

Comprehensive documentation for system administrators, developers, and end-users.

Training materials for end-users to understand system functionalities.

3.1.4 Software Implementation

Implementing the described system architecture involves several steps. Below is a general guideline for the software implementation:

Configure MQTT Communication for Face Service

Integrate an MQTT client library into the Face Service code to enable it to publish and subscribe to MQTT topics.

Implement logic for handling face data and any other relevant information within the Face Service.

Develop Application Portal Using Serenity

Utilize the Serenity platform to create the web application for the Application Portal.

Design and implement user interfaces for inputting data and controlling the Face Service.

Integrate the necessary components for secure communication with the Face Service.

Set Up Azure IoT Hub or Windows MQTT Server

Create an Azure IoT Hub account if using Azure, or set up a Windows MQTT server if using a local server.

Configure the server to manage MQTT communication between the Face Service and the Application Portal.

Ensure proper security configurations, such as using secure protocols (e.g., TLS/SSL) and managing access control.

Implement Security Measures

Apply encryption for data in transit and at rest, especially when dealing with sensitive face data.

Implement secure authentication and authorization mechanisms for both the MQTT server and the Application Portal to prevent unauthorized access.

Testing and Debugging

Thoroughly test the communication between the Face Service, MQTT server, and Application Portal to ensure seamless integration.

Address any issues that arise during testing and debugging.

Documentation and Monitoring

Document the implemented architecture, including configurations and security measures.

Set up monitoring tools to keep track of system performance, security events, and potential issues.

Compliance and Privacy

Ensure compliance with relevant data protection and privacy regulations, especially when dealing with facial recognition data.

Implement privacy measures such as data anonymization and consent mechanisms.

Scaling and Optimization

Consider scalability requirements and optimize the system for efficient resource utilization.

Implement measures to handle increased loads and potential future expansions.

Continuous Improvement

Establish a process for continuous improvement, incorporating feedback from users and monitoring system performance over time.

Stay updated with security patches and updates for the implemented software components.

3.1.5 Methodology

Requirements Analysis

Clearly define the requirements for the system. Understand the functionalities that the Face Service, Application Portal, and MQTT server need to perform.

Identify security and privacy requirements, especially when dealing with facial recognition data.

Technology Selection

Choose the appropriate technologies for each component based on the requirements. Select a suitable programming language, frameworks, and libraries for the Face Service, Serenity platform for the Application Portal, and the MQTT server technology.

System Architecture Design

Design the overall system architecture, including the interactions between components. Clearly define the roles of the Face Service, Application Portal, and MQTT server.

Consider scalability, security, and privacy in the architectural design.

Detailed Design

Break down the system into smaller components and design the detailed functionalities of each component.

Specify the data formats, communication protocols, and interfaces between components.

Development

Implement the Face Service, Application Portal, and MQTT server components according to the design specifications.

Follow coding standards and best practices for each chosen technology.

Integration Testing

Perform integration testing to ensure that the components communicate correctly and that the system as a whole functions as expected.

Address any issues identified during testing.

Security Implementation

Implement security measures such as encryption, authentication, and authorization in both the Face Service and the Application Portal.

Configure the MQTT server with secure communication protocols.

Privacy Measures

Implement privacy measures in accordance with regulations. This may include anonymization of data, user consent mechanisms, and access controls.

Documentation

Document the system architecture, design decisions, and configurations.

Create user guides and developer documentation for future reference.

Deployment

Deploy the Face Service, Application Portal, and MQTT server to their respective environments.

Monitor the deployment process and address any issues that may arise.

Monitoring and Optimization

Set up monitoring tools to track system performance, security events, and user interactions.

Optimize the system for efficiency and scalability based on monitoring data.

Compliance Check

Conduct a compliance check to ensure that the system adheres to relevant regulations and standards, especially concerning data privacy and security.

User Training

Provide training for users who will interact with the Application Portal or other components of the system.

Educate users on security practices and privacy considerations.

Maintenance and Continuous Improvement:

Establish a maintenance plan for ongoing support and updates.

Continuously improve the system based on user feedback, emerging technologies, and changing requirements.

Algorithm

Face Detection Algorithm

Input:

Image frames captured by a camera connected to the Face Service.

Algorithm Steps

Use a pre-trained deep learning model (e.g., a Convolutional Neural Network) for face detection.

For each frame, apply the face detection model to identify faces and their bounding boxes in the image.

Extract the facial regions defined by the bounding boxes for further processing.

Output

Bounding boxes around detected faces and facial regions.

Facial Recognition Algorithm: (Predefined)

Input

Facial regions extracted from the detected faces.

Algorithm Steps

Use a pre-trained facial recognition model to extract features from facial regions.

Compare the extracted features with a database of known faces to identify individuals.

Assign unique identifiers to recognized faces for tracking purposes.

Output

Identification results for each recognized face.

MQTT Communication Algorithm

Input

Identified faces, recognition results, and other relevant data from the Face Service.

Algorithm Steps

Establish an MQTT connection to the designated MQTT server (Azure IoT Hub or Windows MQTT server).
Define MQTT topics for publishing and subscribing to different types of data, such as "face data,""recognition results," etc.

Publish relevant data (e.g., detected faces, recognition results) to the appropriate MQTT topics.

Subscribe to MQTT topics for receiving commands or updates from other components, such as the Application Portal.

Output

Bi-directional communication through MQTT for sharing face-related data between the Face Service, Application Portal, and potentially other components.

Serenity Web Application Algorithm:

Input

User inputs and Commands from the Web Interface

Algorithm Steps

Develop web pages using the Serenity platform to create an intuitive and user-friendly interface.

Implement functionality for users to input commands, such as requesting face data or controlling the Face Service.

Establish communication with the Face Service through API calls or other mechanisms.

Display real-time updates and results from the Face Service on the web interface.

Output

A web application providing a user interface for interacting with the Face Service and accessing face-related functionalities.

Security and Privacy Algorithm

Input

Security and privacy requirements for handling facial recognition data.

Algorithm Steps

Implement encryption for data in transit using secure communication protocols (e.g., TLS/SSL) for MQTT and API communication.

Integrate secure authentication mechanisms for accessing the Application Portal and the Face Service.

Apply access controls to restrict unauthorized access to sensitive data.

Implement privacy measures such as data anonymization and user consent mechanisms to comply with regulations.

Output

A secure and privacy-aware system for handling facial recognition data.

3.1.6 Advantages

Real-Time Facial Recognition

The system enables real-time facial recognition, allowing for quick and accurate identification of individuals based on facial features.

Distributed Architecture

Using MQTT for communication allows for a distributed and scalable architecture. Components like the Face Service, MQTT server, and Application Portal can operate independently and communicate seamlessly over the internet.

Secure Communication

The use of MQTT and the emphasis on security measures in the project, such as encryption and authentication, ensure secure communication between components. This is crucial when dealing with sensitive facial recognition data.

Web-Based User Interface

The Serenity-based web application provides a user-friendly interface accessible through web browsers. This facilitates easy interaction and control of the facial recognition system.

Remote Accessibility

As the components are connected over the internet, the system can be accessed remotely. This is especially advantageous for monitoring and controlling the facial recognition system from different locations.

Scalability

The architecture supports scalability, allowing for the addition of more Face Services, Application Portals, or other components as the system's requirements grow.

MQTT Efficiency

MQTT is a lightweight protocol designed for efficient communication, making it suitable for IoT and real-time applications. It minimizes network bandwidth and is well-suited for resource-constrained environments.

Flexibility and Extensibility

The modular architecture allows for flexibility and extensibility. New features or improvements can be added to the Face Service, Application Portal, or other components without significant disruptions to the overall system.

Compliance with Privacy Regulations

The project includes privacy measures such as data anonymization and user consent mechanisms, which are essential for compliance with privacy regulations, especially when dealing with facial recognition data.

Continuous Improvement

The structured methodology and emphasis on continuous improvement allow the system to evolve over time. Regular updates and enhancements can be implemented based on user feedback, technological advancements, and changing requirements.

Integration with IoT Ecosystems

The use of MQTT and the potential integration with Azure IoT Hub open up possibilities for further integration with larger IoT ecosystems, enabling comprehensive solutions beyond facial recognition.

3.1.7 Application

Access Control Systems

During the Pandemic: Secure access to buildings, offices, or restricted areas has seen increased adoption of contactless face control. Unlike traditional fingerprint biometrics, which may pose a risk of surface transmission, facial recognition provides a non-touchable and more hygienic means of identification.

Airport Security and Boarding

During the Pandemic: Airports have prioritized contactless face control for passenger identification during security checks and boarding processes. This minimizes physical contact between travellers and airport staff, reducing the risk of virus transmission.

Public Transportation

During the Pandemic: Public transportation systems, such as buses and trains, leverage contactless face control for passenger access. This helps to maintain hygiene standards by eliminating the need for physical contact with ticketing or access control devices.

Retail and Payment Authentication

During the Pandemic: In retail environments, facial recognition is increasingly used for payment authentication and secure access to certain areas within stores. This not only enhances security but also provides a touch less shopping experience, mitigating concerns related to shared surfaces.

Healthcare Facilities

During the Pandemic: Hospitals and healthcare facilities employ contactless face control for secure access to critical areas and patient identification. The non-touchable nature of facial recognition aligns with heightened hygiene requirements in healthcare settings.

Education Institutions

During the Pandemic: Educational institutions rely on contactless face control for attendance tracking and secure access to classrooms or dormitories. This minimizes physical contact with identification cards and other devices, addressing concerns related to virus transmission.

Smart Homes and Residential Access

During the Pandemic: Home automation systems integrate contactless face control for secure access to smart homes. This is especially relevant in residential complexes, ensuring that residents can enter shared spaces without compromising hygiene.

Events and Conferences

During the Pandemic: Large events and conferences implement contactless face control for attendee identification and enhanced security. This touch less approach aligns with safety measures by reducing physical interactions during check-in processes.

Gyms and Fitness Centres

During the Pandemic: Fitness facilities adopt contactless face control for member access, allowing individuals to enter without touching access cards or fingerprint scanners. This contributes to maintaining a hygienic environment in shared spaces.

Government and Border Control

During the Pandemic: Government agencies and border control utilize contactless face control for secure identity verification at borders and checkpoints. This approach enhances security while addressing concerns about virus transmission through physical contact.

Hotels and Hospitality

During the Pandemic: Hotels prioritize contactless face control for guest check-ins, room access, and other services. This enhances the overall guest experience by providing a touch less solution at various touch points within the hotel.

IV. CONCLUSION

The implementation of the MQTT protocol in AI systems holds significant promise for advancing the state-of-the-art in intelligent computing. By embracing MQTT's lightweight, scalable, and flexible messaging capabilities, AI practitioners can design and deploy innovative solutions that harness the power of distributed intelligence to address real-world challenges across diverse domains. However, addressing security and optimization challenges will be crucial to realizing the full potential of MQTT-enabled AI systems.

REFERENCES

[1] Qaiser & S.A. Khan. (2006). Automation of time and attendance using RFID systems. *IEEE International Conference on Emerging Technologies*, pp. 60-63. DOI: 10.1109/ICET.335928.

[2] T. S. Lim, S. C. Sim & M. M. Mansor. (2009). RFID based attendance system. *IEEE Symposium on Industrial Electronics and Applications, 2*, pp. 778–782.

[3] Kassem, M. Hamad, Z. Chalhoub & S. El Dahdaah. (2010). An RFID attendance and monitoring system for university applications. ICECS, 851-854. DOI: 10.1109/ICECS.5724646.

[4] M. Kassim, H. Mazlan, N. Zaini & M. K. Salleh. (2012). Web- based student attendance system using RFID technology. In: *Proceedings - IEEE Control and System Graduate Research Colloquium*, pp. 213–218.

[5] M. B. Srinidhi & R. Roy. (2015). A web enabled secured system for attendance monitoring and real time location tracking using Biometric and Radio Frequency Identification (RFID) technology. *International Conference on Computer Communication and Informatics*, pp. 1-5.

[6] Sri Madhu B.M & Kavya Kanagotagi. (2017). IoT based automatic attendance management system. *International Conference on Current Trends in Computer, Electrical, Electronics and Communication*, pp. 83- 86.

[7] H. K. Nguyen & M. T. Chew. (2017). RFID-based attendance management system. *IEEE*. DOI: 10.1109/RTTR.7887874.

[8] H. Zaman, J. Hossain, T. Anika & D. Choudhury. (2017). RFID based attendance system. *8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. DOI: 101109/ICCCNT.8204180.

[9] Rinku Bhagat. (2020). A paho MQTT based IoT-RFID attendance system using nodemcufirmware. *International Research Journal of Engineering and Technology 07*(08).

[10] V. Lampkin, W. T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam & R. Xiang. (2012). Building smarter planet solutions with MQTT and IBM websphere MQ telemetry. *IBM Redbooks*.