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-5 (October 2024)
https://doi.org/10.5281/zenodo.13888533

# MQTT Protocol for Efficient AI Communication

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

[*]**Corresponding Author:** Suraj Khot

Received: 27-08-2024 Revised: 14-09-2024 Accepted: 01-10-2024


## ABSTRACT

**In recent years, advancements in Internet of Things (IoT) and artificial intelligence have revolutionized various fields, including biometric authentication systems. This project focuses on integrating IoT with face recognition technology to create a robust and efficient authentication system. The proposed system leverages MQTT (Message Queuing Telemetry Transport), a lightweight messaging protocol ideal for IoT environments, to facilitate real-time communication between face recognition devices and centralized software.**

**The core of the system involves deploying face recognition devices equipped with cameras and processing units capable of capturing and analyzing facial features. These devices are subscribed to an MQTT broker, such as Mosquitto, enabling them to publish real-time data regarding recognized faces and authentication status. Simultaneously, a centralized software service, also subscribed to the MQTT broker, receives this data and provides a user interface for administrators to monitor and manage access control.**

**Key functionalities include face detection, feature extraction, and matching against a database of enrolled faces. MQTT ensures low latency and efficient data transmission, crucial for real-time applications. The software component integrates MQTT client libraries to seamlessly interface with the MQTT broker, facilitating bi-directional communication between devices and the central server.**

**The project aims to address security, scalability, and real-time performance challenges inherent in face recognition systems by harnessing the power of IoT and MQTT. By implementing this system, organizations can enhance security measures while simplifying access control processes through automated facial recognition technology.**


***Keywords—*** MQTT, Protocol, AI

## I. INTRODUCTION

In the realm of biometric security systems, face recognition technology stands out as a reliable and efficient method for identity verification. With the proliferation of Internet of Things (IoT) devices and advancements in communication protocols like MQTT (Message Queuing Telemetry Transport), integrating face recognition into IoT frameworks presents a compelling solution for real-time access control and monitoring applications.

This project focuses on developing a comprehensive system that combines IoT-enabled face recognition devices with MQTT-based communication to facilitate seamless data exchange between distributed endpoints and a centralized management interface. The integration of MQTT ensures efficient, low-latency communication, crucial for maintaining real-time updates and responsiveness in security-sensitive environments.

The project encompasses the development of both hardware and software components. On the hardware side, IoT devices equipped with cameras and processing units will capture and analyze facial data locally. These devices will subscribe to an MQTT broker to publish real-time facial recognition results and authentication statuses. Concurrently, a centralized software service, also leveraging MQTT, will subscribe to the broker to receive and process these updates. This software component will provide administrators with a web-based interface for monitoring and managing access control.

## II. IMPLEMENTATION

### *Software Development*

The software development involves creating two main components:

1. **Desktop Service**: This component runs on the IoT face recognition devices. It handles tasks such as capturing facial images, processing them using face recognition algorithms, and publishing the results to the MQTT broker. Technologies such as Python and MQTT client libraries will likely be used for this purpose.
2. **Web Application**: This serves as the central interface for administrators. It allows them to view real-time updates from the face recognition

Peer Reviewed & Refereed Journal
Volume-14, Issue-5 (October 2024)
https://doi.org/10.5281/zenodo.13888533

devices, manage user access permissions, and configure system settings. The web application will be developed using web technologies such as HTML5, CSS3, JavaScript (or frameworks like React, Angular, etc.), and server-side scripting languages (e.g., Node.js, Python Flask/Django).

***Third-Party Device Acquisition***

Acquiring IoT devices suitable for face recognition involves selecting cameras and processing units capable of handling real-time image capture and analysis. The devices should support integration with MQTT for seamless communication with the central server.

***Installation of MQTT***

Setting up MQTT involves:

- **Choosing an MQTT Broker**: Installing and configuring a broker like Mosquitto on a server or cloud instance to facilitate message exchange between devices and the software service.
- **MQTT Client Integration**: Implementing MQTT client libraries in both the IoT devices (for publishing data) and the software service (for subscribing to data).

***Learning MQTT***

Learning MQTT entails understanding:

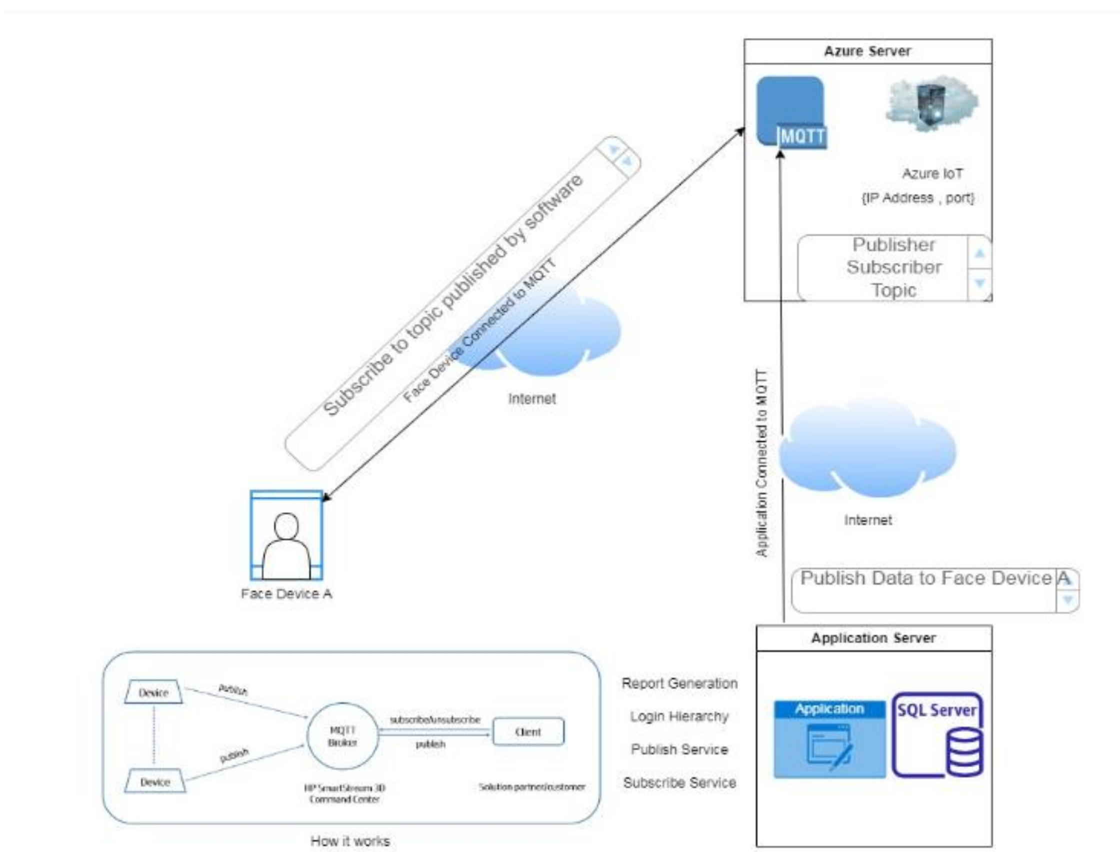
- **Publish/Subscribe Model**: How devices publish data (e.g., facial recognition results) to topics on the broker.
- **Quality of Service (QoS)**: Ensuring message delivery reliability based on QoS levels.
- **Security Considerations**: Implementing authentication and encryption mechanisms to secure MQTT communications.

***Connections and Integration***

Ensuring seamless integration involves:

- **Device-to-Broker**: Configuring devices to connect to the MQTT broker securely.
- **Broker-to-Software**: Implementing MQTT client subscriptions in the software service to receive and process incoming data.
- **Web Application Integration**: Developing APIs or using MQTT directly in the web application to display real-time updates and allow for administrative actions.

## III. BLOCK DIAGRAM



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-5 (October 2024)
https://doi.org/10.5281/zenodo.13888533

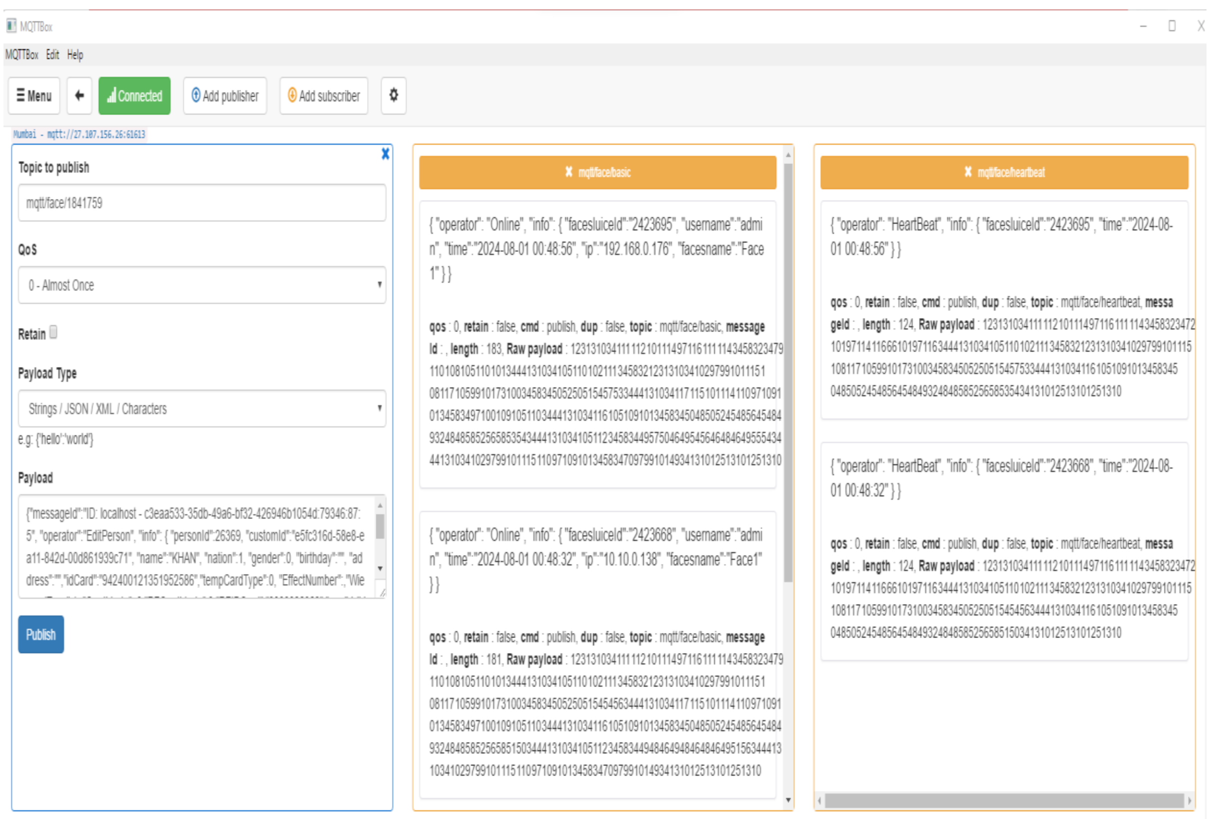
In this system architecture, Face Data and Face Service collaborate by utilizing the MQTT protocol topublish and subscribe data. The Face Service, responsible for processing facial data, connects to an Azure IoT or Windows MQTT server, serving as a central communication hub. This serverfacilitates interactions not only between Face Data and Face Service but also with otherpotential components of the system. Simultaneously, the Face Service establishes a connectionwith an Application Portal, a web application developed using the Serenity platform. This portalacts 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 securecommunication channels. Implementing this architecture involves configuring the Face Servicefor 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 whenhandling sensitive facial recognition data, ensuring compliance with best practices for IoT and web application development.

## IV. RESULT AND DISCUSSION

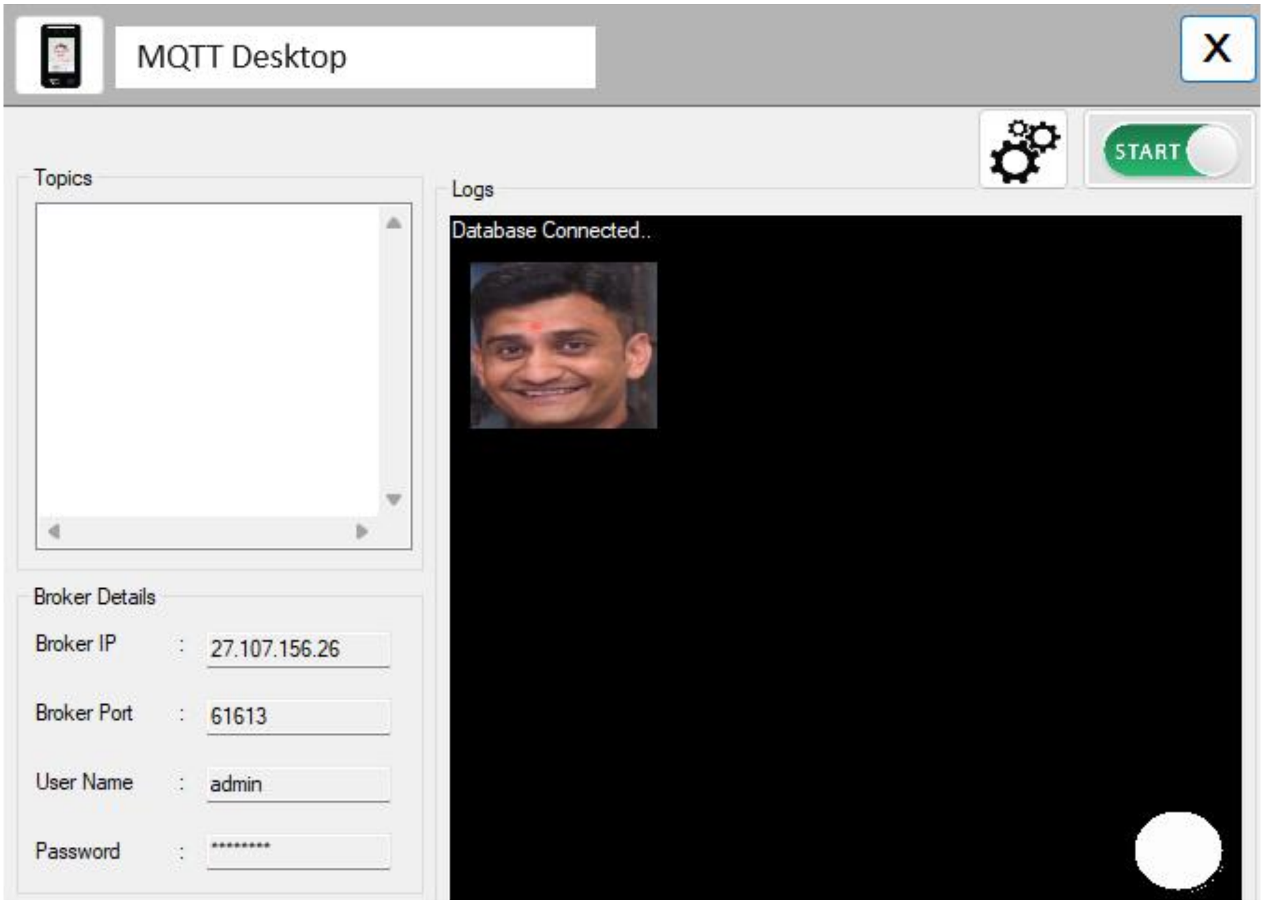
***Results***

The implementation of the face recognition system integrated with IoT and MQTT resulted in several key outcomes:

1. **Real-time Face Capture and Recognition**:
   - The IoT devices successfully captured real-time device status.
   - Results of facial recognition, including identified individuals and authentication statuses, were transmitted in real-time via MQTT. In below image Third party ( HQVT ) pushed data in MQTT BOX
     Pic: Getting real time heartbeat/basic data. ( MQTT BOX )




2. ***Desktop Display of Real-time Data***
   - The desktop service displayed captured facial images and recognition results almost instantaneously.
   - Administrators could view and monitor these updates directly on the desktop interface.




3. ***Enhancements Implemented***
    - An acknowledgment section was integrated into the system.
    - This section provided feedback to users about the recognition process, enhancing user interaction and usability.
4. ***Device Compatibility and Integration***
    - The IoT devices acquired from Thread IoT demonstrated compatibility with the chosen face recognition algorithms and MQTT integration.
    - Smooth integration was achieved between the devices, MQTT broker, and software components.

***Discussion***

The results indicate successful implementation and functionality of the proposed system, validating its effectiveness in real-time face recognition and IoT integration. The following points are discussed:

- **Performance and Efficiency**: The use of MQTT facilitated efficient communication between IoT devices and the centralized software service. Real-time updates were achieved with minimal latency, meeting the project's requirement for responsive access control.
- **Scalability and Flexibility**: The modular architecture of MQTT and the chosen software frameworks (such as Python for device services and web technologies for the application) ensure scalability. The system can accommodate additional IoT devices and scale with increasing data volumes.
- **User Interface and Experience**: The inclusion of a desktop interface and web application provided intuitive control and monitoring capabilities for administrators. The real-time display of facial recognition results and acknowledgment section enhanced user experience and system usability.
- **Security Considerations**: MQTT's support for security mechanisms (e.g., TLS encryption, authentication) ensured secure data transmission between devices and the central server. This is critical for maintaining the integrity and confidentiality of sensitive biometric data.

**Future Enhancements**

While the current implementation meets initial project goals, several areas offer opportunities for future enhancement:

- **Machine Learning Integration**: Implementing advanced machine learning models for face recognition to improve accuracy and robustness.
- **Cloud Integration**: Moving parts of the system to cloud services for enhanced scalability and accessibility.
- **Additional Features**: Adding features such as facial emotion recognition, multi-factor authentication, or integration with other IoT sensors for enhanced security and user interaction.

## V. CONCLUSION

In conclusion, the integration of face recognition with IoT using MQTT has proven to be a viable solution for real-time access control and monitoring systems. This project successfully demonstrated the capability of IoT devices to capture and process facial images locally, leveraging MQTT for seamless communication with a centralized software service. The system achieved efficient real-time updates and enhanced user interaction through desktop and web-based interfaces. Key achievements include robust performance, scalability, and security, facilitated by MQTT's lightweight protocol and support for secure communication.

Looking forward, the project sets the stage for future advancements in biometric authentication systems. Potential enhancements include the incorporation of advanced machine learning algorithms, cloud integration for scalability, and the addition of features like emotion recognition and multi-factor authentication. These developments aim to further enhance system accuracy, reliability, and user experience across various applications in security and beyond.

## REFERENCES

[1] Li, S., et al. (2020). Real-time face recognition system based on MQTT protocol for IoT Devices. *IEEE Access*. DOI: 10.1109/ACCESS.2020.2993087.

[2] Zheng, Y., et al. (2019). Research on face recognition algorithm based on IoT. *Proceedings of the International Conference on Artificial Intelligence and Big Data (ICAIBD)*.

[3] *Mosquitto MQTT broker*. Retrieved from: https://mosquitto.org/.

[4] *Eclipse Paho MQTT client*. Retrieved from: https://www.eclipse.org/paho/.

[5] *OpenCV documentation*. Retrieved from: https://opencv.org/documentation/.

[6] Kim, J., et al. (2021). A secure face recognition system for IoT using MQTT protocol. *Sensors*, *21*(3), 1120. DOI: 10.3390/s21031120.

[7] Zhang, H., et al. (2018). Lightweight face recognition system based on MQTT communication for IoT Applications. *Sensors*, *18*(11), 3911. DOI: 10.3390/s18113911.

[8] Wang, Q., et al. (2017). IoT-based real-time face recognition system using MQTT protocol. In: *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI)*.

[9] Tan, L., et al. (2022). Edge computing-enabled face recognition system for IoT applications using MQTT protocol. *IEEE Internet of Things Journal*. DOI: 10.1109/JIOT.2022.3166527.

[10] Wang, H., et al. (2016). A scalable face recognition system for IoT devices based on MQTT protocol. In: *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*.