Project Title

Design and Evaluation of a RESTful API Gateway for Secure Integration Between ESP32 IoT Sensors and Firebase Cloud Services

Author Name

University of Lincoln, School of Computer Science

author@students.lincoln.ac.uk

1. Introduction

This project sits within the field of Computer Science and more specifically, the subdomains of Internet of Things (IoT), cloud computing, and web application development. The project aligns with the aims of the MSc Computer Science programme by engaging with modern development frameworks, network security practices, and system integration strategies.

With the increasing proliferation of IoT devices in smart homes, agriculture, healthcare, and industry, there is a growing need to securely transmit data from edge devices to cloud services for storage and analysis (Chataut, Phoummalayvane, and Akl, 2023). However, many IoT devices have limited computational power and are vulnerable to attacks, making secure and scalable integration a challenge (Diab and Haile, 2024). Furthermore, developers often need to customize API infrastructure to meet specific data routing, transformation, and authentication requirements.

This project addresses these concerns by developing a secure RESTful API Gateway that bridges data communication between a microcontroller-based IoT device (ESP32) and a cloud database (Firebase Realtime Database). The goal is to provide an intermediary system that enables structured, authenticated, and efficient data transfer in a real-world use case: environmental monitoring.

1. Aims and Objectives

**Aim:**  
To design, implement, and evaluate a RESTful API Gateway that enables secure and scalable data integration between an ESP32-based environmental monitoring IoT device and the Firebase Realtime Database cloud platform.

**Objectives:**

1. To select and configure an ESP32 device with DHT11/DHT22 sensors to collect temperature and humidity data, and establish a Wi-Fi-enabled HTTP interface for outbound communication.
2. To design and develop a lightweight RESTful API Gateway using Node.js and Express, to receive sensor data from the ESP32 and forward it to Firebase Realtime Database in a structured and secure format.
3. To implement security features in the API Gateway, such as HTTPS communication, JWT-based authentication, and input validation to protect against unauthorized access and malformed requests.
4. To test the system's performance and scalability by simulating various device loads and measuring latency, throughput, and API response under different conditions using tools such as Postman and Apache JMeter.
5. To evaluate and document the gateway’s functionality, limitations, and performance, comparing it with direct IoT-to-cloud approaches, and provide recommendations for small-scale IoT deployments.
6. Literature Survey

Prior research has established the importance of API gateways in decoupling microservices and providing secure interfaces for data transmission. Katal et al. (2025) explains how microservice architectures benefit from API gateways for centralizing cross-cutting concerns like authentication, logging, and rate limiting.

Studies such as Pawar et al. (2024) show that traditional IoT-to-cloud communication using MQTT or HTTP can introduce latency and security vulnerabilities without intermediate services like gateways. Moreover, RESTful APIs remain one of the most widely adopted models for web-based communication (Sharma, 2021).

Firebase has been widely adopted for real-time data applications, but its direct integration with IoT devices often lacks access control features. Solutions like the one proposed by Patera et al. (2021) highlight the need for middleware layers to enforce access policies, validate inputs, and manage data formatting before submission to cloud endpoints.

This project builds on these foundations by applying best practices in RESTful gateway design to a focused hardware-software integration problem.

1. Research Methods

This project follows a quantitative and applied engineering approach. A prototype API Gateway will be developed using Node.js and Express, while an ESP32 microcontroller will be configured to send real-time sensor data (temperature and humidity) to the gateway.

Performance will be evaluated by measuring latency, throughput, and resource usage under different test conditions. Testing tools like Postman (for functional validation) and Apache JMeter (for stress/load testing) will be used to simulate real-world IoT traffic.

Security will be verified using manual penetration testing for API endpoints, and structured token validation via JWT to assess protection against unauthorized access.

All data collected will be original, generated via testing and instrumentation of the system.

1. Ethical Considerations

This project does not involve human participants or sensitive personal data. However, since it involves network communication and security implementation, ethical considerations include protecting simulated device identifiers and ensuring secure data handling during testing.

An ethical review will be submitted via the Lincoln Ethical Approval System (LEAS). The project will comply with the University of Lincoln’s data handling and cybersecurity guidelines to avoid misuse or data leakage during the experiment.

1. Project Plan and Risk Analysis

Gantt chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | June | July | August | September |
| 1. Topic Finalization & Approval |  |  |  |  |
| 2. Introduction & Scope Definition |  |  |  |  |
| 3. Ethical Review Preparation & Submission |  |  |  |  |
| 4. Literature Review |  |  |  |  |
| 5. Aims & Objectives Finalization |  |  |  |  |
| 6. Methodology Design |  |  |  |  |
| 7. IoT Device & Firebase Setup |  |  |  |  |
| 8. API Gateway Design & Development |  |  |  |  |
| 9. Security Features Implementation |  |  |  |  |
| 10. Testing & Performance Evaluation |  |  |  |  |
| 11. Results & Discussion Writing |  |  |  |  |
| 12. Final Documentation |  |  |  |  |
| 13. Supervisor Feedback & Corrections |  |  |  |  |
| 14. Draft Submission (Required) |  |  |  |  |
| 15. Final Submission |  |  |  |  |

**Risk Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Likelihood | Impact | Mitigation Strategy |
| ESP32 hardware failure | Medium | Medium | Have a backup board or switch to software simulation |
| Firebase connectivity issues | Low | Medium | Implement retries and offline logging |
| JWT or HTTPS configuration delays | Medium | High | Start security implementation early and test incrementally |
| Overambitious scope | High | High | Stick strictly to one device and one cloud service; avoid feature creep |

References

Chataut, R., Phoummalayvane, A. and Akl, R., 2023. Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. *Sensors*, *23*(16), p.7194.

Diab, S. and Haile, N., 2024. Security of IoT devices: DoS attacks: Security Threats on IoT layers.

Katal, A., Prasanna, P., Birla, R. and Kunal, 2025. Evolution from Monolithic to Microservices Architecture: A New Era in Software Architecture. In Advancements in Optimization and Nature-Inspired Computing for Solutions in Contemporary Engineering Challenges (pp. 235-279). Singapore: Springer Nature Singapore.

Patera, L., Garbugli, A., Bujari, A., Scotece, D. and Corradi, A., 2021. A layered middleware for ot/it convergence to empower industry 5.0 applications. *Sensors*, *22*(1), p.190.

Pawar, S., Jadhav, D.B., Lokhande, M., Raskar, P. and Patil, M., 2024. Evaluation of quality of service parameters for MQTT communication in IoT application by using deep neural network. *International Journal of Information Technology*, *16*(2), pp.1123-1136.

Sharma, S., 2021. *Modern API Development with Spring and Spring Boot: Design highly scalable and maintainable APIs with REST, gRPC, GraphQL, and the reactive paradigm*. Packt Publishing Ltd.

Word Count: 848 words