Secure and Scalable IoT-based Smart Parking System Using MQTT

Adi Tzur decker, Omer Mark

[1] EE, Ben Gurion University, Beer Sheva, Israel
[2] EE, Ben Gurion University, Beer Sheva, Israel

E-mail: aditzur@post.bgu.ac.il, omem@post.bgu.ac.il

Hosts
Dr G. Tel-Zur

[1] EE, Ben Gurion University, Beer Sheva, Israel
E-mail: guycomputing@gmail.com

Abstract

We present a secure and scalable IoT solution for smart parking systems, utilizing MQTT protocol and AWS cloud infrastructure. ESP32 microcontrollers equipped with sensors transmit parking occupancy data securely to an AWS-hosted MQTT broker. Robust encryption and access control mechanisms ensure data security, while AWS scalability enables effortless adaptation to varying parking lot sizes and traffic levels. Integration of Node-RED provides a user-friendly dashboard for real-time monitoring and management. This study highlights the effectiveness of our approach in addressing security and scalability challenges in IoT-based smart parking systems, laying the foundation for future urban infrastructure advancements.

Keywords: IoT, MQTT, ESP32, microcontrollers, Ultra sonic, Parking lot

1. Introduction

In an era marked by rapid urbanization and population growth, the efficient management of urban infrastructure, including parking facilities, has become an increasingly pressing challenge. The proliferation of vehicles coupled with limited parking space availability exacerbates congestion, pollution, and overall urban mobility issues. Traditional parking management systems often struggle to keep pace with the dynamic demands of modern cities, leading to inefficiencies, frustration among motorists, and environmental degradation.

To address these challenges, there is a growing imperative for innovative solutions that leverage emerging technologies such as the Internet of Things (IoT). IoT-based smart parking systems offer the promise of real-time monitoring, optimization, and resource allocation, thereby enhancing the utilization of existing parking infrastructure while minimizing the environmental footprint associated with vehicular congestion.

However, the implementation of IoT-based parking solutions is not without its challenges. Security concerns regarding data privacy and unauthorized access loom large, particularly as sensitive occupancy data is transmitted over networked systems. Moreover, scalability issues pose a significant hurdle, as traditional centralized architectures struggle to accommodate the scalability requirements of dynamic urban environments.

Existing parking management solutions often fall short in addressing these challenges comprehensively. Legacy systems lack the agility and real-time capabilities afforded by IoT technologies, leading to suboptimal utilization of parking resources and exacerbating congestion woes in densely populated urban areas.

Against this backdrop, this study aims to propose a novel IoT-based smart parking system architecture that addresses the twin imperatives of security and scalability. By leveraging MQTT protocol for efficient data transmission and AWS cloud infrastructure for robust scalability, our solution seeks to offer a viable path forward in the quest for sustainable urban mobility solutions tailored to the needs of the modern world. Through empirical evaluation and analysis, we seek to demonstrate the efficacy of our approach in meeting the evolving demands of urban parking management in an over-crowded and overpopulated world.

2. Related Works

In big, populated area, the system proposed has examined through its efficiency for all kinds of attributes considering. In previous works related to this subject, the system purposed was examine back and forth, in all kinds of manners to determine its efficiency. All previous works we looked up to, tried to implement various of systems as such.
Most implementations include same architecture when it comes to physical measurements such as vehicles distances from a parking slot.
All related works used, same as us – HC-SR04 Ultra Sonic distance sensor [1]. In research took place in order to determine the distance sensor's reliability and cost-effort, it was found the sensor was gratefully accurate within its range of detection (2cm to 400cm), where in the close edge of 2-3cm, an object detection was sometimes fails [2].
Since 2-3cm distances from the sensor are not very common among average drivers, we weren't very concerned about it, as were the researchers. Due to the sensor's high accuracy, reliability and affordability, we decided to use it in our own implementation. The integration of cloud services has been identified as a crucial enabler of scalability and management in IoT-based parking solutions. By leveraging cloud platforms such as Soft-Layer (IBM), researchers have demonstrated the ease with which IoT devices can be integrated into scalable, cloud-based architectures. This approach not only facilitates seamless data transmission and storage but also enables efficient resource allocation and management, thereby enhancing the scalability and adaptability of the overall system, including application intended for drivers to feedback [3]. Maintaining a well-functioning system with high security standards is also a great challenge when approaching such a task. Great amount of sensitive personal data transmitted in the medium that must kept safely to avoid it end up in hostile hands. Research in this field has shown that high security standards for IoT-based parking systems are achievable yet require careful configurations. In the research, the implementation relies on a former framework designed by same group, which security and data protection are great deal. The researchers also proven scalability alongside high performance – CPU utilization and power consumption wise, where security standard kept high, especially where ECC security method is taking place instead of RSA [4].
Prior research has proven scalability, cost-effective and secure solutions for a big world problem. Implementation of such solution was also proven easy to do using microcontrollers, MQTT protocol, distance sensors and cloud services.

3. System Architecture

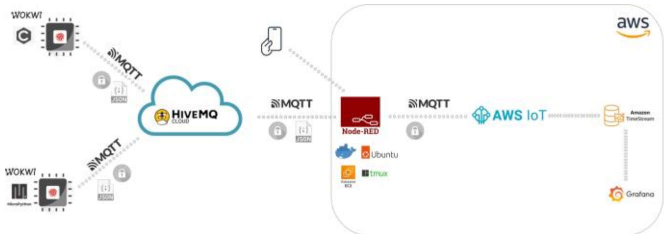
Our System Architecture was designed to perform the following tasks:

1. Physical data measurements such as distance and motions coming from parking cars.
2. Data transmit using MQTT protocol.
3. MQTT broker that receives data from clients.
4. Database to store all streaming data.
5. UI for monitoring system and management.
6. Visualization tools of the parking lot occupancy over time

The system architecture of our IoT project revolves around a streamlined data flow aimed at real-time monitoring and visualization of parking spot occupancy.
The first stop in the data flow there ESP32 microcontrollers equipped with distance and motion sensors, responsible for detecting the parking slots availability. For any parking slot's status change, the ESP32 microcontrollers publish MQTT messages to an MQTT broker hosted on AWS IoT core.
The AWS MQTT broker serves as a central hub for collecting all parking lot occupancy data collected by the ESP32 microcontrollers. Using AWS Datastream service, the broker efficiently manages the incoming data streams, ensuring reliable and low-latency communication between the microcontrollers and cloud infrastructure.
In parallel, we employ Node-RED for visualization and user-friendly interface of the real time parking lot data and statuses, Node-RED is a service running on a docker container, where the whole cluster is hosted on an AWS EC2 instance. Also allow us to publish from the MQTT broker to

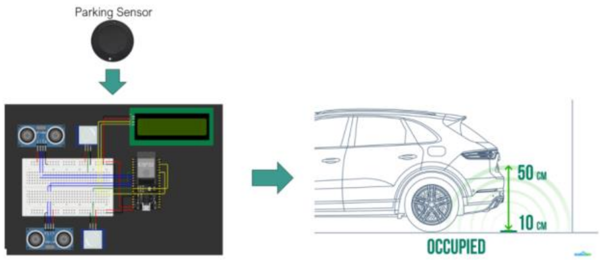
clients – an important and useful ability for our clients with LED sensors.
Finally, the collected data from Datastream service integrated to Grafana, Service hosted on Grafana Cloud and easily integrates with AWS data streams. Grafana provides comprehensive analytics and extra visualization capabilities. We used Grafana to create customized dashboards that enabled us to visualize the parking lot occupancy over time. The Grafana visualization empowers parking lot management, conclusions and decisions making efforts to optimize the parking lot infrastructure.
Overall, the system architecture combines the capabilities of ESP32 microcontrollers, AWS cloud infrastructure, MQTT protocol, Node-RED UI tools and Grafana analytics to create a robust and scalable solution for monitoring and optimizing parking lots management and monitoring.

4. Implementation



We implemented our parking system using the MQTT protocol, employing two MQTT brokers: HiveMQ and AWS IoT thing. The primary broker, HiveMQ, supports three clients. The clients implementation was conducted entirely on the Wokwi simulator platform [5].

HiveMQ Cloud, a managed MQTT cloud service provided by HiveMQ, simplifies the deployment and management of MQTT brokers. With an intuitive GUI and built on AWS infrastructure, it offers a hassle-free experience for setting up and maintaining MQTT brokers. HiveMQ offers two free plans: the "Serverless" plan, which is entirely free, and the "Starter plan," which we opted for during a 14-day trial period [6]. This plan ensures secure communication channels through Certification Authentication (CA), enhancing the overall reliability and security of our parking system.

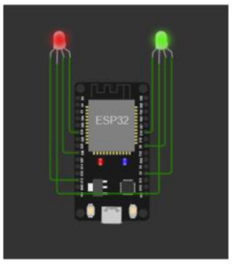


Parking Sensor:

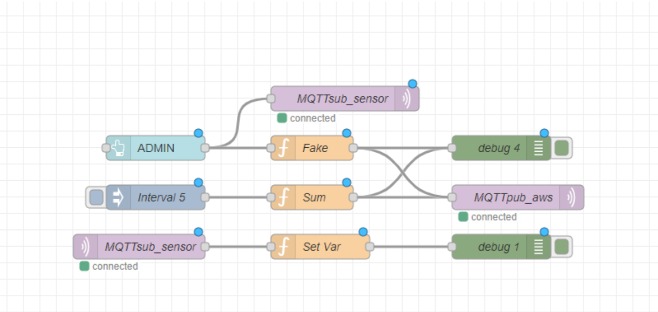
Implemented on Wokwi using ESP32 microcontrollers, the parking sensor system covers two parking spots with each ESP. Programmed in C language, the system is secured with CA certification and requires MQTT username and password for authentication, The data passes through secure port 8883. Each ESP setup consists of two sets of sensors: a motion sensor (PIR motion sensor [4]) and a distance sensor (Ultrasonic Distance Sensor - HC-SR04).
When motion is detected by the PIR motion sensor and the distance measured by the Ultrasonic Distance Sensor is less than 100 cm, it signals that one of the parking spots is occupied. Subsequently, the system publishes an MQTT message in JSON format to indicate that the spot is taken.
Conversely, when the parking spot becomes free, the sensors detect the change, and the system publishes another MQTT message, indicating that the spot is available again.

Json client publish example:

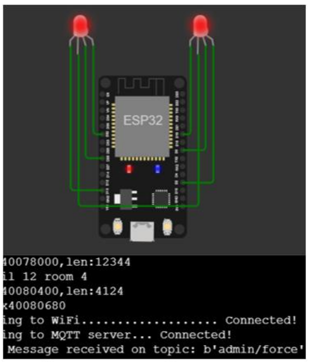
{
"clientid": "ESP8266Client-1341",
"sensor_number": "1“
"spot": "1",
"status": "taken",
}



Led Sensor:
Implemented on Wokwi using ESP32 microcontrollers, written in MicroPython, and MQTT subscribed to the topic "data/parking" published by the Parking Sensor. This system also consists of 2 LEDs, each covering 2 parking spots.
The ESP32 fetches the payload from the subscribed MQTT topic and interprets it as JSON. When the status of a parking spot changes (either taken or free), the system updates the LED color accordingly: red indicating the spot is taken and green indicating it is free.



Node-RED:
our third client, operating on an AWS EC2 instance running Ubuntu 22.4, Node-RED's editor port is secured using IP restrictions, aiming to simulate a VPN environment.
Docker manages Node-RED's container within a tmux session, facilitating live monitoring of the logs for debugging purposes.
The primary purpose of utilizing Node-RED is to handle scalability, simplify flow, and manipulate JSON messages. Additionally, it sends data at intervals of 5 seconds to AWS IoT for data visualization and database storage, aiming to minimize message overhead over the network.
Furthermore, we've implemented a feature in Node-RED granting a "corrupt admin" privilege. This privilege allows for the remote activation of a red light on a parking spot by publishing an MQTT message to the topic "admin/force".



AWS Section:

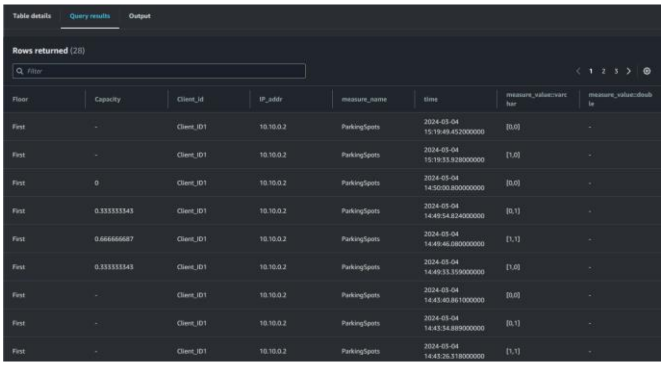


Table details | Query results | Output

Rows returned (28)

Floor	Capacity	Client_id	IP_addr	measure_name	time	measure_value::varchar	measure_value::double
First	-	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 15:19:49.452000000	[0,0]	-
First	-	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 15:19:33.928000000	[1,0]	-
First	0	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:50:00.800000000	[0,0]	-
First	0.333333343	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:49:54.824000000	[0,1]	-
First	0.666666687	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:49:46.080000000	[1,1]	-
First	0.333333343	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:49:33.359000000	[1,0]	-
First	-	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:43:40.861000000	[0,0]	-
First	-	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:43:34.889000000	[0,1]	-
First	-	Client_ID1	10.10.0.2	ParkingSpots	2024-03-04 14:43:26.318000000	[1,1]	-

Firstly, we utilize the IoT Thing service, which employs an end-to-end architecture to connect to our Node-RED instance. It subscribes to the topic "data/parking" and receives MQTT messages in JSON format, sent at 5-second intervals. These messages are then streamed to the Timestream Table. Secondly, in the Timestream Table, we manipulate the data to determine if a parking spot is taken. Lastly, we connect this information to Grafana, displaying a visual representation of which parking spots were occupied at different times of the day.

5. Results

The results of our IoT project underscore the efficacy of the tools and methodologies employed in achieving a robust, scalable, and secure smart-IoT based parking system. The seamless integration of simulated ESP32 microcontrollers, MQTT protocol, AWS cloud infrastructure, HiveMQ broker service, Node-RED visualization tools, and Grafana analytics demonstrates the ease of implementation and interoperability offered by these technologies.
The utilization of ESP32 microcontrollers equipped with sensors proved to be highly effective in accurately detecting parking spot occupancy, while the MQTT protocol facilitated efficient and reliable communication between the microcontrollers and the cloud brokers. Leveraging AWS's Datastream service ensured the scalability of the system, enabling seamless handling of varying levels of parking lot sizes and traffic volumes.
The integration of Node-RED and Grafana provided intuitive and customizable interfaces for real-time monitoring and visualization of parking spot occupancy data. Node-RED's visual programming capabilities enabled the rapid development of user-friendly dashboards, while Grafana's advanced analytics tools facilitated comprehensive data analysis and visualization.
Moreover, the implementation of robust security measures at various layers of the system ensured the protection of sensitive data against unauthorized access or manipulation. By employing encryption, access control, and other security best practices, we safeguarded the integrity and confidentiality of parking occupancy data without compromising system performance.
Overall, the results highlight the successful implementation of a secure, scalable, and user-friendly smart parking system, underscoring the potential of the tools and methodologies utilized in addressing the challenges of modern urban mobility and infrastructure management. The seamless integration of IoT technologies, cloud infrastructure, and

visualization tools paves the way for future advancements in smart city initiatives, offering stakeholders actionable insights for optimizing parking resources and improving urban mobility.

6. Conclusion

In conclusion, our IoT project aimed to address the pressing challenges associated with parking management in urban environments, including congestion, inefficient resource utilization, and lack of real-time monitoring capabilities. Through the integration of ESP32 microcontrollers, MQTT protocol, AWS cloud infrastructure and cloud brokers, Node-RED visualization tools, and Grafana analytics, we developed a robust and scalable smart parking system that offers tangible benefits in terms of efficiency, security, and user experience.

Our solution effectively overcame several key aspects of the main issue. Firstly, the real-time monitoring capabilities provided by the system enable parking administrators to make informed decisions regarding resource allocation, thereby alleviating congestion and improving overall urban mobility. Additionally, the implementation of robust security measures ensures the integrity and confidentiality of parking occupancy data, addressing concerns regarding data privacy and unauthorized access.

While our solution successfully addresses many aspects of the parking management challenge, there are areas where further refinement is needed. For instance, the scalability of the system could be further optimized to accommodate larger parking facilities or even broader smart city initiatives. Additionally, while our focus was primarily on parking spot occupancy monitoring, there is potential for the integration of additional sensors and data sources to provide a more comprehensive view of urban mobility patterns.

Nevertheless, the advantages of our solution are evident. The ease of integration and interoperability of the tools and technologies used in our system make it readily adaptable to other smart city or smart world solutions. The modular nature of the architecture allows for seamless integration with existing infrastructure, or the development of new applications tailored to specific urban challenges.

Looking ahead, our system serves as a blueprint for the development of future smart city initiatives, offering a scalable, secure, and user-friendly solution for real-time monitoring and optimization of urban infrastructure. By leveraging IoT technologies, cloud computing, and data analytics, our solution lays the groundwork for a more efficient, sustainable, and connected urban environment, ultimately enhancing the quality of life for residents and visitors alike.

Acknowledgements

We thank for our supervisor Dr G. Tel Zur fot intruduce us to the IoT world and it's various tools;

References

[1] HC-SR04 datasheet.
[2] Joni Welman Simatupang, Aida Mahdalena Lubis, " IoT-Based Smart Parking Management System Using ESP32 Microcontroller",2022, IEEE.
[3] Ashhwath C, Rohitram V and Sumathi G, " Smart Parking System using MQTT Communication Protocol and IBM Cloud", 2021, Journal of Physics: Conference Series.
[4] PIR motion sensor datasheet.
[5] Wokwi simulator.
[6] HiveMQ