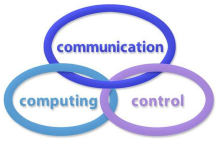
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 3, Month: June, Year: 2025
Article Number: 7077, https://doi.org/10.15837/ijccc.2025.3.7077



CCC Publications

Laravel and Vue.js as tools to control IoT devices over the internet. Current state-of-the-art.

D. Popa, I. Buciu

Daniel Popa
Dept. of Electronics and Telecommunications
Faculty of Electrical Engineering and Information Technology
University of Oradea
410087, Oradea, Romania
popa.daniel@gmail.com

Ioan Buciu*
Dept. of Electronics and Telecommunications
Faculty of Electrical Engineering and Information Technology
University of Oradea
410087, Oradea, Romania
ibuciu@uoradea.ro *Corresponding author: ibuciu@uoradea.ro

Abstract

Looking around us, we can notice that the Internet of Things (IoT) technologies are becoming indispensable. In an era where communication via the Internet is driving our lives, and the control of IoT equipment is increasingly popular, the development of web applications to allow the control of this equipment is in a continuous evolution. The evolution and diversification of web applications has created a favorable environment for the development and diversification of web servers and for the development of server-side and client-side programming technologies. Another important aspect is the security and protection of data transmitted over the Internet. This study presents an analysis of the current status of web applications development for controlling IoT devices using Laravel and Vue.js technologies, the evolution of web servers used to run web applications, respectively the evolution of technologies used to develop application programming server side and client side. The work focuses on the presentation of IoT-specific communication protocols, on the presentation of the possibilities of connecting Laravel to IoT equipment and on the transmission of this data in the user interface. Possibilities for securing connections and encrypting transmitted data are analyzed and presented.

Keywords: Internet of Things, Laravel, Vue.js.

1 Introduction

The idea of being able to control, open or close, the garage door through a device (smartphone, computer, ecc.), or to program the central heating system remotely has become a necessity. Laravel [1]

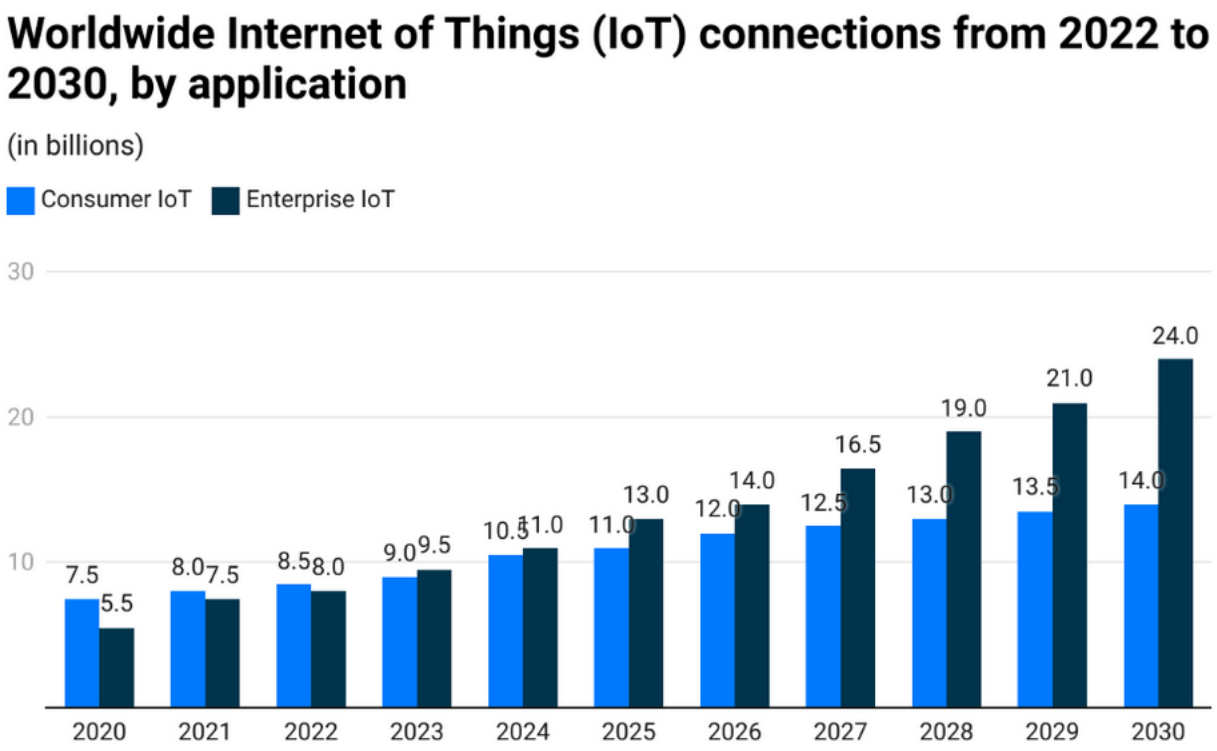


Figure 1: The number of IoT devices connected to the Internet [5]; Evolution over time.

and Vue.js [2] technologies are one of the most used technologies that can perform versatile IoT functions. Today, most manufacturers implement applications to facilitate the use of things. But, in order to benefit from them, it was necessary for the equipment to be able to communicate with each other, and a network is needed in which all these equipment is connected, and the Internet is such a network. As the website wikipedia.org [3] shows, from the beginning until now the Internet has developed spectacularly. From showcase sites to online stores or online stock trading sites, etc. , today, apart from traditional servers or computers, a lot of equipment is connected to the Internet: web cameras, surveillance systems, cars, etc. [3]. All these: "smart switch"or "automatic access gate", are more than simple devices, they can communicate with each other, they have various sensors. The Internet of Things is [4] a terminology that refers to equipment with sensors, processing capacity, software and various other technologies that connect and exchange data, via the Internet, with other devices and/or systems also connected to the Internet [4]. Society reacted positively to the appearance of these technologies so that their evolution had an upward trend. The website enterpriseappstoday.com presents statistics and estimate, made by statista.com [5], of the use of IoT in the coming years as depicted in Figure 1.

Over time, the control interfaces of these equipments have evolved, adapted to market requirements and we have the possibility to control them remotely through a web application on any device that has a web browser installed. Britannica defines a web application as [6] a computer program stored on a remote server and run by its users through a Web browser. A Web application is an efficient form of software because the use of browsers allows the application to be compatible with most computers and standard operating systems [6]. Web applications have had a rapid and spectacular evolution [7]. If at the beginning of the 1990s they contained only text in static HTML (HyperText Markup Language) pages, later it became possible to add images, video and audio files. In 1995, the development of the Javascript programming language allowed the addition of interactive elements, including vector graphics. Later, in 2005, web applications received a new approach through the introduction of dynamic pages, and from 2015 they received the functionality of a native application [7].

But why use a web app over native apps for IoT control? Linda Rosencrance in her article "Considerations to make an IoT web app" published on techtarget.com shows that [8] IoT web apps provide users with an intuitive interface to help them control these devices much more efficiently. IoT

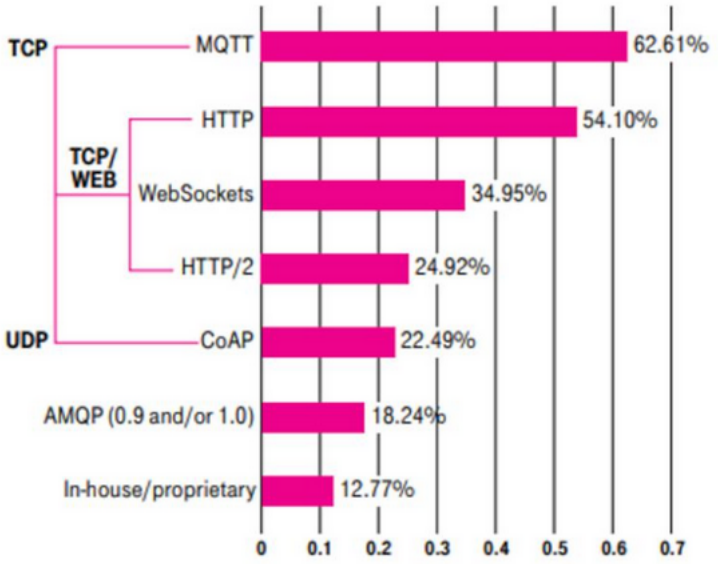


Figure 2: Popularity of IoT communications protocols [9]

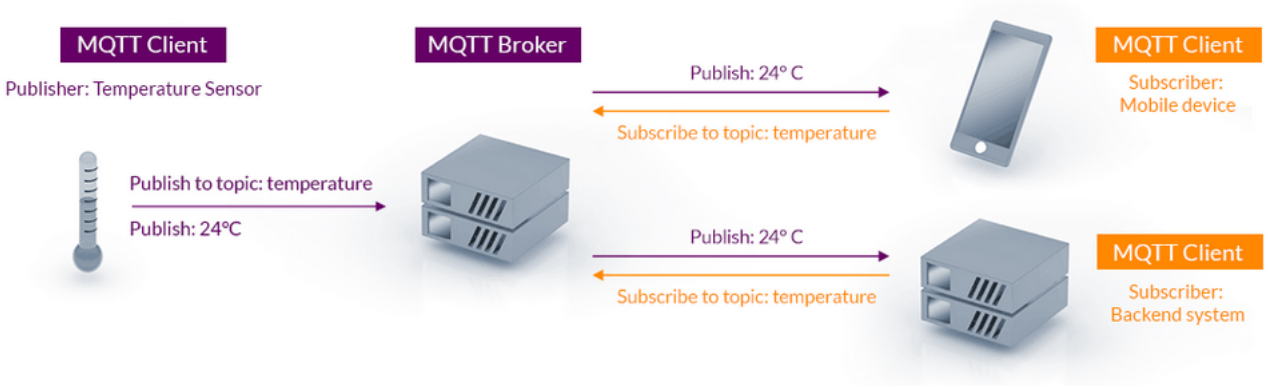


Figure 3: MQTT Publish / Subscribe Architecture [11]

mobile apps must be built and tuned specifically for app stores and mobile phones such as Android or Apple. Mobile apps are best when they are designed for a specific use and purpose. For example, a mobile application in a closed loop with a medical device attached to a person to monitor blood sugar and deliver insulin must work reliably without depending on networks or being subject to security breaches. An IoT web application can be flexible for applications that need to be updated regularly and are contextually enriched with enterprise data. An IoT web application can also have business process benefits for the many technology professionals who interact with it to generate value [8].

2 IoT communications

In order to control or supervise the status of a IoT equipment, it is necessary to communicate with them. Today, there are several IoT communication protocols, but we will focus only on those that are used more frequently used [9] as depicted in Figure 2 and indicated by The website www.iot.daviteq.com in its article "What are the most popular IoT protocols? Characteristics and applications of them", updated on Jul 3, 2023 [9].

MQTT (**Message Queuing Telemetry Transport**) is an OASIS standard messaging protocol for IoT. It is designed as an extremely lightweight publish/subscribe messaging transport, ideal for connecting remote devices with a small code footprint and minimal network bandwidth [10]. Its architecture is shown in Figure 3.

On the Cloudflare website HTTP (**Hypertext Transfer Protocol**) is presented as [12] the foun-

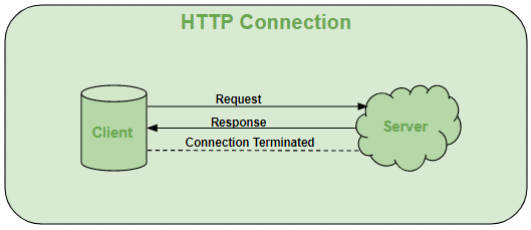


Figure 4: Hypertext Transfer Protocol [12]

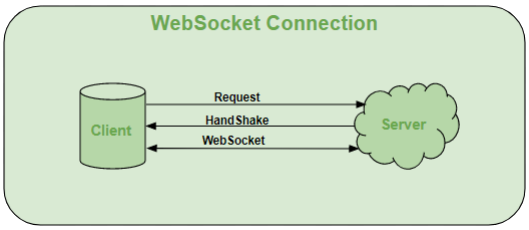


Figure 5: WebSocket Connection [13]

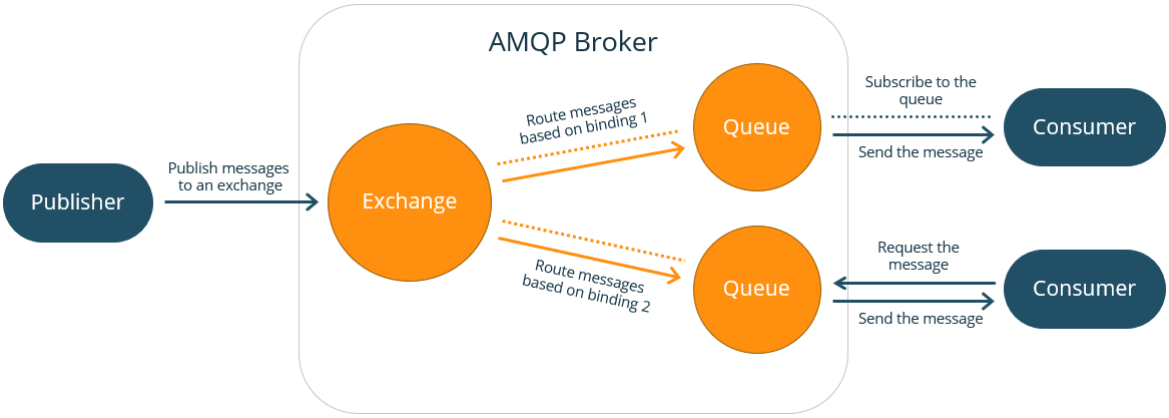


Figure 6: AMQP Clent-Sever Message Pattern.

dation of World Wide Web (WWW) and is used to load web pages using hypertext links (see Figure 4). HTTP is a protocol designed to transfer information between network devices. A typical HTTP flow involves a client machine making a request to a server, which then sends a response and the connection between the two is closed [12].

Unlike HTTP, **WebSocket** is a stateful, bidirectional protocol (see Figure 5), meaning that the connection between client and server remains active until terminated by either party [13].

Constrained Application Protocol (CoAP) is a transfer protocol, a specialized web for use with constrained nodes (eg, low-power, lossy networks). The protocol is designed for machine-to-machine (M2M) applications. CoAP is like a limited HTTP designed for constrained IoT devices. Similar to HTTP, it is client-server based, where the client makes a request and the server sends a response [14].

Similar to MQTT is the **Advanced Message Queuing Protocol** (AMQP), which has revolutionized the way applications communicate over the Internet [15]. This protocol enables reliable communication between devices. It supports various message types, including text, binary, and streaming data. Many industries such as finance, healthcare, transportation and logistics have adopted AMQP [16]. The AMQP Client-Server Message Pattern is illustrated in Figure 6).

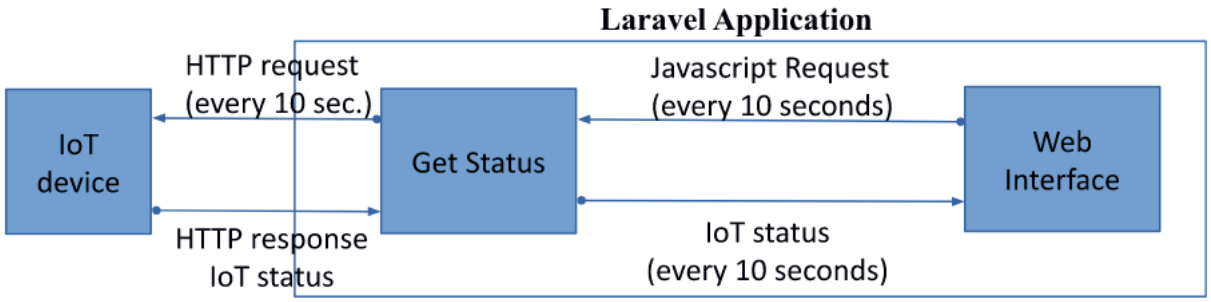


Figure 7: Get IoT device status

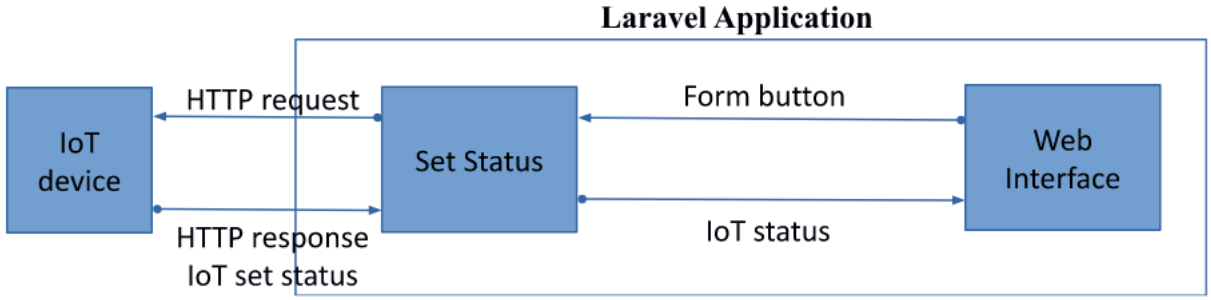


Figure 8: Set IoT status

3 Laravel to control IoT devices

Laravel is a framework for creating web applications with expressive and elegant syntax [17]. It provides a structure and a starting point for creating any type of application. It provides the ability to build robust, full-stack applications in PHP using Laravel and Livewire. A monolithic front-end based on React or Vue by pairing Laravel with Inertia can be easily designed.

Out of the box, Laravel has elegant solutions for the common features needed by all modern web applications without wasting time looking for various packages and reinventing the wheel [18]. Solutions for implementing authentication in a web application, or for interacting with the database, etc are available [17].

However, going through the official Laravel documentation, surprisingly, to date, the Laravel community has not developed a package (out of the box) for interacting with IoT devices.

Let's see what solutions there are to interact with these devices. Considering the popularity of communication protocols and the fact that web applications mainly use the HTTP protocol, we will first see the possibilities of controlling and displaying the status of an IoT device using this protocol.

Let's recall that an HTTP connection is made up of two important sequences: the client's request to the web server and the web server's response sent to the client, after which the connection is closed. Then we consider the fact that in general we need to give commands to a device, or we want to see what its state is (if a garage door is closed or open, for example). Considering the above, Laravel allows the implementation of a function/method in a controller, the method that will be called through API routes (Application Programming Interface) of Laravel as shown in Figure 7.

In a similar way, the status of the equipment can be changed, specifying that the request will not be called at a predefined time interval. It will be called upon pressing a button in the web interface, for example (see Figure 8).

In the situation where we want to keep the history of all status changes, a database can be used and then we can find out the current status from the database.

To implement control over IoT devices using the MQTT protocol, several libraries created by various contributors may be found. We will only present the package created by user salmanzafar949 from github.com website.

The package contains [19] a library for connecting/publishing/subscribing to an MQTT broker.

The package also offers various possibilities that we will not insist on. All these can be accessed at the address https://github.com/salmanzafar949/MQTT-Laravel [19]. The methods to be implemented for subscription and publication are shown below:

```
1. use Mqtt;
2.
3. public function SubscribetoTopic($topic)
4. {
5.         Mqtt::ConnectAndSubscribe($topic, function($topic, $msg){
6.             echo “Msg Received: \n”;
7.             echo “Topic: {$topic}\n \n”;
8.             echo “\ t$msg\n \”;
9. },$client_id);
10. }
```

It can be seen that the class from the library line 1 is used, the class that allows instantiation, calling, connecting and subscribing to a topic, on line 5, through the **ConnectAndPublish()** method. When the equipment receives an order, it will send the information on lines 6, 7 and 8.

To publish messages on a topic (or sending a command to the device), the creator of the library proposes the following method:

```
1. use Mqtt;
2.
3. public function SendMsgViaMqtt($topic, $message)
4. {
5.         $client_id = Auth::user()->id;
6.         $output = Mqtt::ConnectAndPublish($topic, $message, $client_id);
7.         if ($output === true)
8. {
9.             return “published”;
10. }
11.         return “Failed”;
12. }
```

Similar to what was previously, the Mqtt class is used, line 1; then connection and sending the message through the **ConnectAndPublish()** method, line 6. If the publication was successful, the word "published" will be displayed, otherwise “Failed”.

We will not substantiate the implementation of communication through the AMQP and COAP protocols. Notable, the reader may find on www.github.com packages developed by contributors that allow interaction with IoT device. For example, in [20] this package contains an AMQP framework for Laravel and Lumen to publish and consume messages. It offers possibilities to add the message

to the queues or to follow the queues. The installation and configuration methods are described at https://github.com/bschmitt/laravel-amqp [20].

Let's see now the case where a permanent connection between the web interface and the IoT equipment is necessary. In this case, we found the possibility of using the Laravel Reverb package [21]. This package brings extremely fast and scalable real-time WebSocket communication directly into your Laravel application [21]. Lyron Foster in the article "Building Real-Time Applications with Laravel and WebSockets" on the website https://medium.com, presents a way to implement real-time web applications using Laravel and WebSockets [22].

Laravel is a popular framework with which real-time functions can be implemented relatively easily using WebSockets. Real-time communication allows devices to exchange data instantly, ensuring quick responses and timely actions. With real-time data updates, interactive dashboards become more informative and engaging for users [23].

4 Vue.js frontend for IoT

Vue.js is a progressive Javascript framework, affordable, performant and versatile for building web interfaces for users [24]. It's based on standard HTML, CSS, and JavaScript and offers a declarative, component-based programming model that helps you efficiently develop user interfaces of any complexity.

Considering the fact that Vue.js is a Javascript framework used in particular to create web interfaces for users, it does not have "out of the box" packages that allow the control of IoT equipment. Let's see what possibilities we have to implement code sequences that allow us to have access to the state of an IoT device, or to send a command to change the state of an IoT.

A simple way is the one in which the device allows communication through the HTTP protocol. Then we can send requests to the device, or to a procedure accessed by a controller through a route (as was presented in previous section), and display the response in the user interface. The modality is similar to the one presented in the LARAVEL section.

Similarly, to modify the state of a device, a request will be sent, also to the device directly or to a method.

In addition to those listed above, the Vue.js community has developed a promise-based client, AXIOS, for node.js and the browser [25]. It is isomorphic, so it can run in browser and nodejs with the same code base. On the server side it uses the native node.js http module, while on the client (browser) it uses XMLHttpRequests.

For communication with IoT devices using the MQTT protocol we found, for example the PAHO plugin, which allows you to connect to an MQTT broker and subscribe to topics in the Vue application [26]. It uses paho-mqtt to connect to the broker and provides several useful features such as automatic reconnection, message handling and error notifications. For details, it can be accessed https://github.com/kaandesu/vue-paho-mqtt [26].

Abiola Farounbi, a frontend developer, in his article "How to build a real-time Vue app with WebSockets" published on the website https://blog.logrocket.com/ in March 30, 2023, presented a way of using websockets for the creation of real time applications [27]. "In this article, we explored WebSockets by building a Vue chat application that provides a seamless and fast user experience, allowing users to communicate with each other in real time. You can also build on this example to add any changes or new features" [27].

5 Privacy and Security

We know that the most secure communication network is the "isolated" one, without internet connection, but without internet access it is impossible to remotely control IoT equipment.

The increase in the number of IoT devices in a network causes a decrease in the security of that network and in the future it is expected that the risk of cyber attacks based on artificial intelligence will increase. This scenario requires taking strict security measures [28].

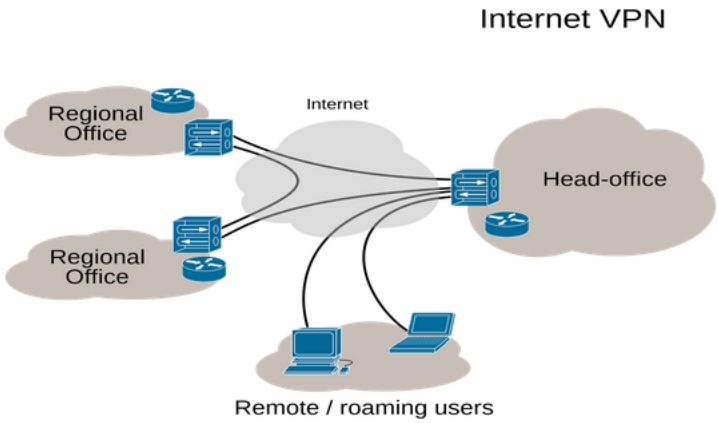


Figure 9: VPN conectivity overiew

Today, in order to increase the security of communications with IoT equipment in a network connected to the Internet, several security options are considered. Below are the basic security principles that we believe we should consider when implementing control through a web interface on a network of IoT devices.

First of all, it is advisable, we would say mandatory, to create a VPN (Virtual Private Network) for all IoT equipment [29] (see Figure 9). A VPN is a mechanism for creating a secure connection between a computing device and a computer network, or between two networks, using an insecure communication medium such as the public Internet [29].

Then the Transport Layer Security (TLS) encryption protocol can be considered [30]. It is popular in IoT environments and increases the security, confidentiality, integrity and authenticity of information.

Securing the application is another important aspect that must be taken into account and Laravel allows the implementation of authentication and authorization "out of the box".

According to the article "A Beginner's Guide to Secure MQTT in IoT: Safeguard Your Connected Devices Seamlessly", published on the website https://bytebeam.io a series of security measures should be considered when creating a control system for IoT equipment on the MQTT protocol. It will be easily noticed that many of them are applicable in general.

Among those approaches presented in the article, we would list the following measures to secure IoT: [31]

- **Certificates**: Use a public-private key pair to authenticate devices/users, eliminating the need for passwords and relying on cryptographic keys for identity verification
- **Limiting client privileges**: Assign appropriate privileges to clients based on their roles to reduce the risk of unauthorized access
- **Monitoring and logging**: Implement comprehensive monitoring and logging to detect suspicious behavior and track activities
- **Session Management**: Configure the broker/server to terminate inactive or unresponsive client connections with appropriate session timeouts, preventing resource exhaustion and unauthorized access
- **Connection Timeout**: Define client connection timeouts to prevent potential DoS attacks and improve network stability
- **Secure Gateway Authentication**: Ensure that the web gateway supports secure authentication methods to verify the identity of clients
- **Protection against Cross-Site Scripting (XSS) attacks**: Implement measures to prevent Cross-Site Scripting attacks, which could compromise the web gateway and expose sensitive

information to attackers. Use input validation and output encoding techniques to mitigate XSS risks

- **Traffic Analysis**: Monitor traffic to identify unusual patterns or unexpected spikes in data transmissions. Unique traffic patterns can indicate potential security threats or unauthorized access attempts.

6 Conclusions

Several communication protocols with IoT devices have been created, each one characterized by certain advantages. We believe that, depending on the situation, it is preferable to use one or the other, and the development of web systems and applications for IoT control should be adapted to the use of protocols.

Laravel is a powerful framework for developing web applications on the server side (backend). We found that it offers the possibility to develop web applications for IoT control, using various communication protocols. Although, for now, Laravel has not developed an "out of the box" package, it is probably only a matter of time until such a package appears.

Also, Vue.js, although it has developed certain libraries that allow the control of IoT equipment, remains a framework recommended only for creating web interfaces (frontend). Control remaining to be implemented through server-side programming languages.

Used together, Laravel and Vue.js, I consider that they offer a lot of possibilities for the complete development of a web application used in any field.

Regarding ensuring security and confidentiality in the operation of a web application, there would be a lot to specify. We conclude by saying that although it will be a continuous challenge, the careful implementation of the principles of security and confidentiality, respectively the continuous supervision of the system, can lead to its stability and safety.

Acknowledgment

The research has been funded by the University of Oradea, within the Grants Competition "Scientific Weak - May 21, Research Grant", Project No. 122/25.06.2021

References

[1] [Online]. Available: Laravel *The PHP Framework for Web Artisans*, https://laravel.com/, Accesed on December 21, 2023.

[2] [Online]. Available: Vue.js *The ProgressiveJavaScript Framework* https://vuejs.org/ Accessed on December 21, 2023.

[3] [Online]. Available: *Wikipedia* https://en.wikipedia.org/wiki/Internet/ Accessed on December 21, 2023.

[4] Buyya, R.; Dastjerdi, A. V. (2016). *Internet of Things - Principles and Paradigms*, Elsevier, 2016.

[5] [Online]. Available: *Enterprize Apps Today* https://www.enterpriseappstoday.com/stats/internet-of-things-statistics.html#Number_of_Active_IoT_Devices/ Accessed on April 4, 2024.

[6] [Online]. Available: *Britannica - Science & Tech* https://www.britannica.com/topic/QuickTime/ Accessed on February 17, 2024.

[7] [Online]. Available: *Medium* https://oleg-uryutin.medium.com/a-brief-history-of-web-app-50d188f30d/ Accessed on February 20, 2024.

[8] Rosencrance, L. (2021). Considerations to make an IoT web app, *Techtarget, December, 21th, 2021*, 2021.

[9] Davitech (2023). What are the most popular IoT protocols? Characteristics and applications of them *Davitech, September, 21th, 2023*, 2023.

[10] S. Quincozes, T. Emilio and J. Kazienko. (2019). MQTT Protocol: Fundamentals, Tools and Future Directions, *IEEE Latin America Transactions*, 17(09), 1439–1448, 2019.

[11] [Online]. Available: https://mqtt.org/ Accessed on February 14, 2024.

[12] [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/ Accessed on April 20, 2024.

[13] [Online]. Available: https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/ Accessed on April 20, 2024.

[14] [Online]. Available: https://academy.nordicsemi.com/courses/cellular-iot-fundamentals/lessons/lesson-5-cellular-fundamentals/topic/lesson-5-coap-protocol/ Accessed on April 22, 2024.

[15] Natale, J. (2024). What is Advanced Message Queuing Protocol (AMQP)?, *Nonamesecurity, December, 21th, 2024*, 2024.

[16] [Online]. Available: https://support.smartbear.com/readyapi/docs/testing/amqp.html/ Accessed on May 24, 2024.

[17] [Online]. Available: https://laravel.com/docs/11.x, Accesed on May 24, 2024.

[18] N. Yadav, D. S. Rajpoot and S. K. Dhakad. (2019). LARAVEL: A PHP Framework for E-Commerce Website, *2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India*, 2019.

[19] [Online]. Available: https://github.com/salmanzafar949/MQTT-Laravel, Accesed on May 25, 2024.

[20] [Online]. Available: https://github.com/bschmitt/laravel-amqp, Accesed on May 25, 2024.

[21] [Online]. Available: https://laravel.com/docs/11.x/reverb, Accesed on May 27, 2024.

[22] Foster, L. (2023). Building Real-Time Applications with Laravel and WebSockets, *Medium, July, 12th, 2023*, 2023.

[23] Ram, M. (2023). Laravel-Powered IoT Applications: Connecting Devices and Data Management, *Medium, August, 5th, 2023*, 2023.

[24] [Online]. Available: Vue.js https://vuejs.org/, Accesed on June 02, 2024.

[25] [Online]. Available: https://axios-http.com/, Accesed on June 02, 2024.

[26] [Online]. Available: https://github.com/kaandesu/vue-paho-mqtt/, Accesed on June 02, 2024.

[27] Farounbi, A. (2023). How to build a real-time Vue app with WebSockets, *LogRocket – Frontend Analytics, March, 30th, 2023*, 2023.

[28] T. Sasi, A. Habibi Lashkari, R. Lu, P. Xiong, S. Iqbal (2005). A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges, *Journal of Information and Intelligence*, December, 2023.

[29] Juma, M., Monem, A. A., and Shaalan, K. (2020). Hybrid End-to-End VPN Security Approach for Smart IoT Objects, *Journal of Network and Computer Applications*, 158, 102598, 2020.

[30] E. Rescorla. (2018). The Transport Layer Security (TLS) Protocol Version 1.3., *IETF TLS workgroup*, 2018.

[31] Team Bytebeam. (2023). A Beginner's Guide to Secure MQTT in IoT: Safeguard Your Connected Devices Seamlessly, *Team Bytebeam – Bytebeam, September, 25th, 2023*, 2023.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Popa, D.; Buciu, I. (2025). Laravel and Vue.js as tools to control IoT devices over the internet. Current state-of-the-art., *International Journal of Computers Communications & Control*, 20(3), 7077, 2025.

https://doi.org/10.15837/ijccc.2025.3.7077