

# Design A Model To Analyze Open Source Nodejs Iot Frameworks

Arvind Totey and Dr. Rajeev G. Vishwakarma

Department of Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore  
(M.P.) - 452010

---

## Abstract:

The NodeJS IoT framework has come a long way in the last decade, and various architectures for distributed and networked systems have been developed. Most systems are built utilizing fragmented approaches for specific application areas, making unified solutions difficult. Interms of NodeJS IOT Frameworks, Standardization, on the other hand, is a critical component. Because with Node.js, we now have Full Stack Developers who can independently handle both the server and client sides of a project. Due to its event-driven, non-blocking, and asynchronous methodologies, Node.js is fast and reliable for large files and high network demand apps. Developers can also manage entire development on single pages (SPA) use IoT with Node.js. Design a model to analyze open source node JS IoT frameworks that support a wide range of application layer communication protocols. As a consequence of this research, back-end technologies that can be used to manage access to IoT data streams and micro-services have been examined. The findings indicate that the Internet of Things gateway, which was dependent on an open-source ecosystem, had a framework performance that was consistent and dependable despite a particular information size and concurrent scale. In most sensing contexts, these characteristics satisfy the application criteria of the Internet of Things. To design a model to analyze open source nodeJS IoT frameworks.

**Keywords :** NodeJS , IOT Frameworks , IoT protocols

## 1. INTRODUCTION

When we discuss JavaScript, Node JS reigns at the top. JavaScript has its popularity quotient among many other technologies, and Node JS is among the top. NodeJS [1] is a prominent server-side technology that is based on JavaScript. It comes with a wide variety of frameworks that help in the process of supporting a variety of application instances. The Node.js IoT frameworks have garnered much support from the development community, but it can be difficult for programmers to decide which one to use. These frameworks are essential for developing NodeJS applications more quickly and straightforwardly. Let's take a quick look at the fundamentals of Node.js and its most prominent qualities before we go on to the plethora of Node.js frameworks that are currently available. The Internet of Things (IoT) is a term that refers to a new generation of cyber-physical services that are being developed as a result of the rapid improvements in computer communication technology and the increasing number of

items connected to these networks. These services can improve people's everyday routines and businesses in various sectors, such as health, energy, etc. To increase the number of Node JS based IoT-enabled applications, IoT resources such as devices and software services should be merged into a single pool of publicly available solutions. At the device and platform levels, interoperability, connection, and scalability are all issues. These issues hinder IoT adoption, and new services are delayed. "Interoperability" is a phrase used to describe the ability of diverse platforms, systems, and services to communicate and work together seamlessly in all aspects of their operation. Interoperability between IoT services and platforms is the purpose of the research, which aims to establish a seamless interconnection architecture. Interoperability is essential, and a single technology, such as an industry standard, is considered impossible. Interoperability should be embedded into all system layers, from data to applications/services to middleware and networking, all the way down to the particular device. For interoperable services to be useful, they must be reusable across a wide range of application domains. Ambient assisted living, smart energy management, and smart mobility all rely on an IoT ecosystem's flexibility to be reused, including service reconfiguration capabilities. Internet of Things interoperability relies on middleware, which is a critical component of the stack. This layer's job is to connect various resources and exchange data and messages across many domains and interfaces. In addition, Node JS and IoT middleware must be easy to use and provide means for service connectivity, data analytics, context awareness, resource management, and control via graphical user interfaces. Middleware for the Internet of Things. There are various ways to keep your data safe while allowing you to access it. Application layer protocols (ALPs) are typically implemented as part of middleware. ALPs, which control data flow such as sensor readings and commands, are the primary means through which IoT services [2] and devices communicate. ALPs exist, but none have yet succeeded in overtaking the worldwide IoT market and becoming the most dominating business in the field. ALPs have various serialisation formats and communication patterns, such as publish/subscribe or request/answer, as well as the level of service they offer. Numerous protocols must be implemented to allow for cross-layer communication and the establishment of virtual brokerage systems for ambient intelligence and context-aware apps. IoT ecosystem interoperability is the subject of this essay. Open-source middleware frameworks have been used to build an IoT platform that anyone may use. Synaisthisi is a cloud-based and scalable IoT application-enabler platform that uses ALPs such as REST, HTTP, Websockets, Constrained Application Protocol (CoAP), and Advanced Message Queuing Protocol (AMQP) (AMQP). The IoT stack can also benefit from interconnected dockerized containers. Protocols top this off for identifying who has access to and who owns cloud-hosted data streams and micro-services, including those provided by open-source brokers RabbitMQ and Ponte. Integrates with the platform's other components and supports the oneM2M technical standard via the open-source framework OM2. Cyber-physical applications that are both versatile and complicated require access to a large number of interconnected protocol interfaces. The platform's performance will be evaluated using a variety of ALP procedures. The remainder of this document is organized as follows. In Section II, we explain some of the associated concepts and the research that has been done so far on ALPs and the current Node JS IoT platform design approaches. Section III details the platform's containerized version's internal architecture

proposed methodology. As an example of how to create new Node JS-based IoT-enabled services, Section IV offers a discussion of the IoT marketplace and the design principles for complicated IoT service approaches that can be employed in ambient intelligence use cases. In Section V, we show examples of simulation results, and in Section VI, we summarise our findings.

## 2. RELATED WORK

With the Internet of Things, the future has come early. Today, more than half of all businesses use the Internet of Things to develop or ship products. And because each connected device, sensor, and piece of equipment has its own interface, you need unified IoT software that can handle a lot of data. We should mention that developers can use a wide range of programming languages to build these kinds of solutions. IoT software development often uses Java, Python, C++, and Ruby, all of which do a great job with this task. But none of the languages made just for this kind of programming are as good as JavaScript. Even though it has some problems, it is more than perfect for making IoT apps. Relevant programmers like Node.js as a runtime environment for IoT mobile apps for good reasons. We go over the benefits of IoT and NodeJS and show why they work well together for your future IoT projects.

Using choreography, C. Damian et al. [3] Assertions, validation, and security checks can all be integrated into IoT software systems. Node.JS (java script on servers)-based software systems can benefit from the techniques presented here. You may use it in your Node.js application to report and handle errors and make assertions and other semantic checks. Additionally, the module's GitHub page provides additional examples and a modified version of the functions.

C. Thomás Oliveira et al. [4] an experiment could be conducted to verify the efficacy of the implementation. Sending encrypted requests slows down FIWARE-based IoT apps, but the added security they give more than makes up for it. Security features can be added to several IoT protocols using open source code on GitHub.

H. C. Irwan et al. [5] it is stated in this paper how some persons can monitor their heart rate and interbeat intervals. This prototype used WebSocket, nodeJS, and JavaScript on a Samsung Gear S3. The server nodeJS do signal processing and GUI code on behalf of the user in a system architecture definition. The signals are processed using a JavaScript algorithm.

In this work, R. O. Jacob et al [6], a battery-powered GPS tracking system with an SOS feature is demonstrated. Real-time tracking and an emergency warning are now possible thanks to the Internet of Things. The microcontroller and GPS module provide users and the central server with position data, which is then transmitted to them via the GSM module via AT instructions and SMS messages. The central server will use MongoDB and NodeJS to check the location regularly. Thanks to Open Layers, this data may be shown as a map on an HTML-based website hosted on Amazon Web Services (AWS).

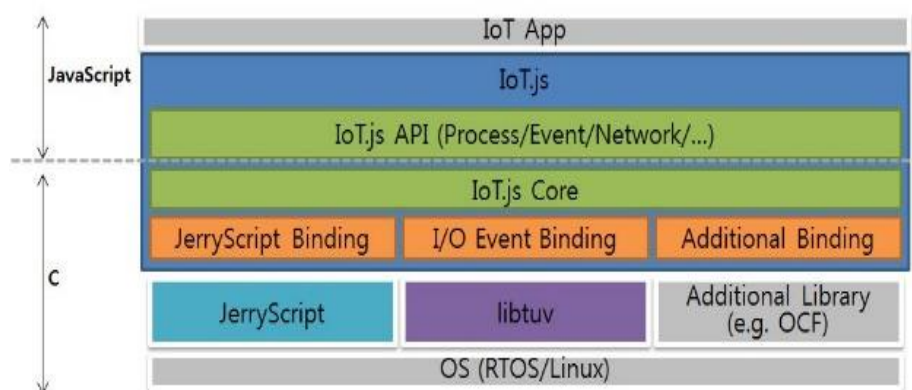
C. -P. Yi et al. [7] this article aims to examine the design of the monitoring and servo terminals, how the server's performance may be changed, and how power calculations can be swiftly performed when the signal has been received. An actual utility signal was used to verify the smart metre platform's ability to handle real-world processing signals, proving its validity and speed.

E. Stark et al. [8] open technologies like Mozilla IoT in JavaScript can help developers create Internet of Things systems that are more flexible and scalable. Open-source platform Things Gateway was utilized in this project. It is free to use, developed by Mozilla and a small number of volunteers. For intelligent devices with Internet of Things capabilities, frameworks and a new application layer are being developed to make the process easier. Among the emerging standards discussed in the article is the Web of Things. The Things Framework may be used to develop an adaptor for any new device running on the Raspberry Pi computer using this information.

### 3. PROPOSED METHODOLOGY

Open-source and platform-independent, NodeJS is a backend JavaScript runtime environment. To run JavaScript code outside of a browser, it uses the V8 engine. V8 is the engine that powers Node.js. NodeJS can generate dynamic page content, collect form data, add, delete, alter database data, and create, read, write, delete, and close server-side files. As a result, it is often regarded as the best tool for designing applications such as those video streaming, social media feeds, online apps, and games. NodeJS may be used to build applications using the MVC, Full-Stack MVC, and REST API frameworks. There are a range of frameworks that web application developers may use to support a variety of different use cases during the application's development lifecycle. Node.js frameworks often fall into one of three categories: MVC, Full-Stack MVC, or REST API.

The MVC frameworks: The controllers, models, and views of a program can be divided into three distinct pieces using these frameworks: controllers, models, and views. Maintaining and scaling an application is considerably more accessible if you separate development concerns from the rest of the program. Model-view-controller (MVC)[9] framework Express.js has proven itself time and time again.



**Figure 1: IoT JavaScript Framework [17]**

Frameworks for building full-featured MVC applications: Full-Stack MVC frameworks have a lot to offer developers regarding building real-time applications, including libraries, template engines, and a variety of other options. In addition, they can manage the frontend and backend development of apps. The "representational state transfer" (REST) API and frameworks [11]. By providing a built-in REST API experience, Node.js frameworks are a trusted reference for fast constructing applications. Since network programs use so many different architectural

styles, you won't have to worry about them. Developing internet-dependent software applications is made much easier thanks to the large majority of these frameworks already including an application programming interface (API). Node.js, The Internet of Things [12], was once a futuristic concept, but it is now a reality. The Internet of Things is now included in the delivery or creation of more than half of all businesses today. IoT software must handle massive amounts of data to manage linked devices and sensors, each of which has its own interface[13]. Developers have a wide range of programming languages at their disposal, allowing them to develop solutions like this. Java, Python, C++, and Ruby are all widely used in Internet of Things (IoT) software development and do an excellent job when put to work on this task.

On the other hand, JavaScript [14] is the only programming language specifically designed for this development. Despite a few flaws, it's an excellent platform for building internet of things apps. For various convincing reasons, Relevant's programmers choose Node.js as the runtime environment for the Internet of Things mobile apps. We'll go over the advantages of using Node.js and the Internet of Things together[15] and why that's a good idea for your future Internet of Things applications.

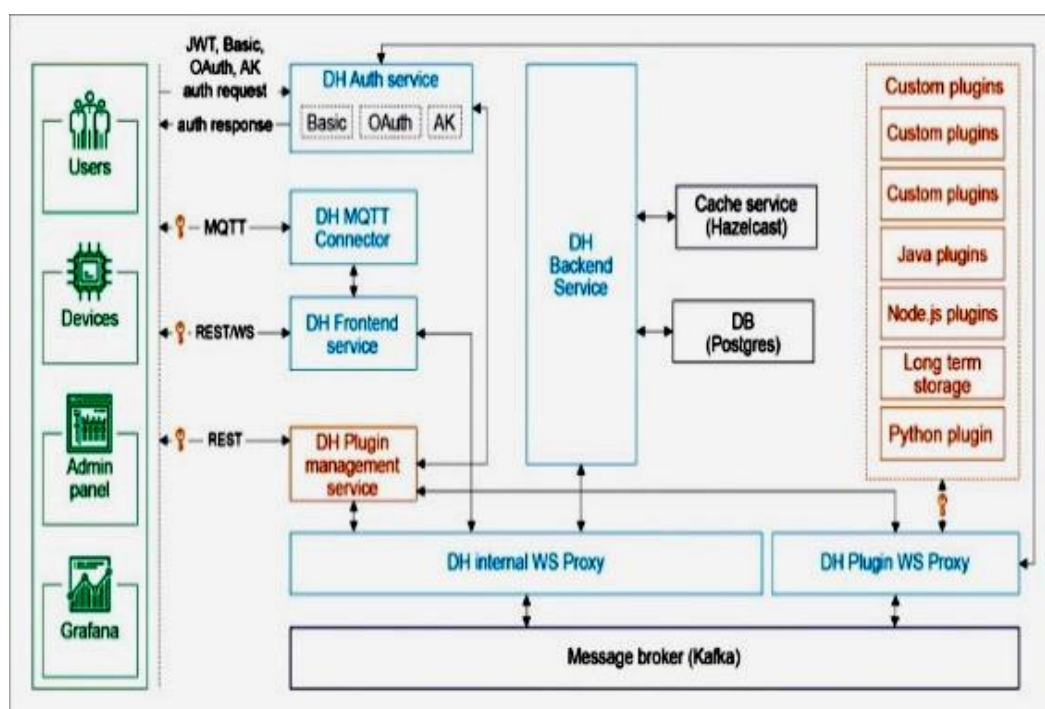


Figure 2: Analyze Open Source NodeJS IoT Frameworks [6]

### a) IoT development with NodeJS

To begin, let's investigate the role that Node.js plays in functioning the Internet of Things. The Internet of Things (IoT) is a concept that refers to the act of connecting physical objects to a network through the use of technologies such as embedded sensors and software as well as other wireless technologies. The platform for the Internet of Things acts as a go-between in the data collection, tracking, and exchange process amongst various devices. The framework needs to be able to deal with data coming in at a constant rate in real-time while yet making consistent forward progress. The ability for users to automate their environments is made feasible by IoT

systems since they handle data visualization and administrative responsibilities. Even if there are a lot of different technologies being used, frameworks are still an essential aspect of the process of developing intelligently linked systems. Because this industry is still in its infancy, there is currently no clear winner in this field.

On the other hand, most software developers and hosting providers believe that developing Internet of Things apps with Node.js is the most reliable option. And we will explain in great depth why it is that this is the case. Why ought you to make use of Node.js?

#### **b) Advantages of Node.js in IoT**

As a first step, the Node.js runtime environment is a cross-platform, open-source platform for building quick and scalable applications. Because of its versatility and good performance [16], it is popular among programmers. Node.js has several advantages over its competitors, including IoT libraries and a strong developer community. Because of the following reasons, Node.js development can improve IoT systems and apps.

#### **c) Using Node.js, IoT protocols may be simply integrated.**

Integration is essential to the functioning of the Internet of Things. Node.js makes integrating into a wide variety of ecosystems easier by supporting the MQTT protocol. This protocol is frequently utilised by Internet of Things applications in order to connect to third-party services. It is not the frontend that does the integration; rather, it is the backend. MQTT.js serves as the basis for all Node.js Internet of Things software development kits produced by industry-leading cloud service providers[17]. An excellent example of this is provided by AWS IoT Node.js. MQTT allows for the reliable processing of millions of messages and the routing of those messages to AWS endpoints and other devices.

#### **d) Node.js is a performant and quick web framework.**

Any gadget that makes use of the Internet of Things works with data that is constantly shifting. As a result, it requires a robust Internet of Things infrastructure capable of managing enormous data streams in real-time. Node.js comes with something called Node Packaged Modules (NPM), essentially sophisticated repositories that cater to the needs of the internet of things. Node.js is capable of compiling JavaScript into native machine code and providing it with high-speed execution, which is what Node.js is capable of integrating with Google's V8 engines. The NPM repository includes[18] more than 80 packs for the Arduino controller as well as several different packages for the Pebble and Fitbit wearable.

Additionally, it suggests incorporating Internet of Things frameworks for Bluetooth devices and sensors. Using the flow-based programming tool Node-RED, you may construct Internet of Things applications with minimal coding. In the internet of things (IoT) age, having the ability to construct C ++ modules is quite helpful. C drivers are currently available for most of the hardware, allowing developers to integrate it into Node easily.

#### **e) Scalability and resource efficiency are two of Node.js's strong suits.**

An application created in the Node.js framework consumes fewer resources than one written in Java or Ruby because of its faster execution time. Each thread requires 2 megabytes of

memory. Therefore, an 8-gigabyte device can manage 4,000 connections at once. Using Node.js, the same device can handle one million simultaneous connections.

**f) Using Node.js, you can ensure the security of your data.**

The Internet of Things is a network of connected devices that share data for a particular purpose. For a successful implementation, all of the devices must communicate with each other. Internet of Things (IoT) devices need to be secure in three ways: access rights, device, and user authentication, and well-designed code architecture.

**d) Node.js can meet these requirements.**

Procedures for secure identification

Other options for Node.js application authentication include using tokens, auth0, and jwt. Each has advantages and downsides when viewed through the lens of the internet of things.

Tokens are useful because of their efficiency but don't provide complete protection[19]. They are an effective way of authentication since they allow you to identify a particular user and decide whether or not to provide them access.

Even encrypted tokens or passwords can be stored in the firmware of an IoT device. Tokens, jwt tokens, and auth0 tokens can all be stolen if the device they're kept on is broken into physically.

Using the Node.js framework, you'll be able to use all of the server-side authentication options available. Passport, auth0, and jwt may all be installed manually with Node Package Manager. Integrating cloud services, such as AWS-IoT or Azure IoT Hub Node.js, into your application is possible. It is also possible to conduct secure HTTP requests using this method.

We may either do it yourself (with Node.js, it's not difficult) or use hosting that already has HTTPS certificates configured for you.

**The development of APIs is made easier by the use of Node.js.**

Another advantage of utilizing Node.js is the ability to create APIs. First, a mobile API was built utilizing tools like Node.js by LinkedIn developers, who were pioneers in this area. Web service and API development are supported by Node.js frameworks integrated within the platform. A corporation using Node.js can quickly build an API[20] because only a few lines of code are needed.

If you need to process requests from a large number of Internet of Things sensors and wearable devices all at once, you should consider utilizing Node.js.

Even while Node is a valuable piece of technology in and of itself, it must be utilized by the appropriate individuals. You may count on Relevant to provide the most qualified Node.js developers to assist you in developing IoT software by utilizing this adaptable technology. '

Node.js is an excellent tool for developing new applications that are compliant with the security rules for the Internet of Things.

We collaborate with well-known businesses operating in various markets to bring our Internet of Things solutions closer to commercial viability.

Real-time data alerts and web-based monitoring [21] are made possible by the Internet of Things platform known as Sensor Innovation.

Users can access and analyse data regarding the air quality generated by sensors using the Airthings dashboard.

Our Internet of Things development services includes everything from prototyping to delivery and maintenance of the finished product [22].

We have experience in various fields, such as UI/UX, Big Data, and DevOps, amongst others; however, this list is not exhaustive.

Because of the protocol's method, the Node JS Solid platform is handy for use cases involving the Internet of Things (IoT). Instead of being stored locally on a single device, coding data is now kept in the cloud, where it may be retrieved whenever desired and by whoever needs it (which may be vulnerable to data loss or have limited storage or availability). During the product bake-in phase, developers can debug and gain greater visibility into processes. This contributes significantly to the product's overall value and saves a substantial amount of resources when utilized for quality assurance. It can be challenging to acquire forensic data such as CPU profiling or memory snapshots, for example, despite the significance of these data in comprehending the performance of a programme. These artefacts can be obtained through the use of Node.js and other open-source software, but there is a potential security risk associated with the special issues of delivering information about memory or crashing problems in order to analyze them. Scripts modified or enhanced using open source software can also present security risks, produce erroneous results, or have other negative consequences. Using Node Source Node JS [23]Solid makes it easy and safe to perform tasks like as profiling, taking snapshots, and troubleshooting the latency of event loops. Ephemeral artifacts can be moved to a permanent position at the user's discretion, to the system's nature. Even if the process is stopped, the artefacts continue to exist for a significant amount of time. The open-source community does not have any solutions for debugging a non-responsive programme that has wholly halted a production operation. Because it is situated within the event loop, the NodeSource agent is in a position to retrieve the essential data that is being sought. In addition to encryption, it employs cryptographic keys and curves, which helps to ensure that it is a valuable addition to any protocol. Even if stringent development methods and testing in a staging environment are beneficial, the extreme loads, one-of-a-kind use cases, and long-tail anomalies that only arise in a production environment are where the real action takes place. Visibility, comprehension, and the capacity to debug issues after they have occurred are essential for effective and speedy triage and problem resolution when addressing defects in internal processes, apps, and APIs that have an impact on customers and business partners, utilising NodeSource results in significant time, financial, and resource savings.

**Table 1: Comparative Analysis Deliberations and NodeJS IOT Frameworks**

Deliberations	NodeJS IOT FRAMEWORKS
Ensuring the security of essential files and services	N Solid's built-in security policies can be customized to protect your apps better. As a result of always-on vulnerability scanning, you will be informed to any new threats that may arise.



Performance that is quick and dependable for end-users	The entire team can obtain an understanding of the application's behaviour thanks to N Solid's application profiling tools.
Observance of legal and personal requirements	With low overhead, Node Source Certified Modules provides licence enforcement and quantifiable trust in third-party Node.js modules for developers who wish to use open-source code while minimizing risk.
The ability to quickly identify and fix problems and minimize or eliminate downtime	An alert system will notify your team whenever an issue is detected, so they can take action before it gets out of hand and reduce your average time to resolution (MTTR).
To keep one's competitive edge, one must speed up innovation.	Node.js development can be swiftly and simply implemented throughout many teams in your organization using the Node Source platform.

According to a survey conducted, eighty percent of the Internet of Things devices that are the most extensively utilised have major security flaws. As more people become interested in the Internet of Things, there is an increased demand for innovative testing and verification tools for IoT systems. Combining logs, exceptions, assertions, and checks is made possible by the NodeJS IOT Frameworks module[24], which was developed as part of the Private SKY research. This module is compatible with any software system that employs Node.js and executable choreographies.

### **Features of the IoT Application Development Platform with Node.js**

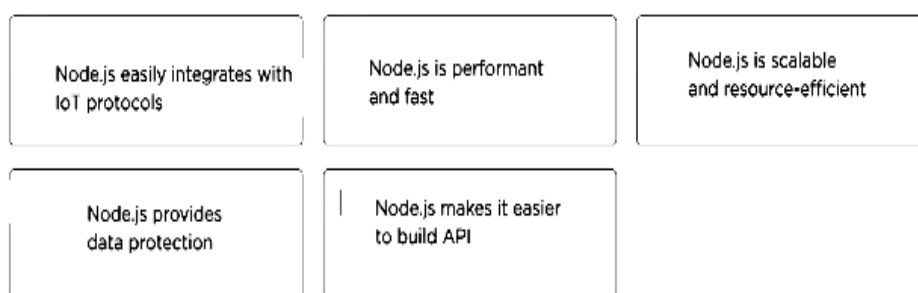
An interoperability module that works on a gateway to connect devices, a persistence and analytics module that stores and processes data, a context module that provides relevant information to services and users, a semantic operation module, and a resource and event module that manages the device's capability and generates events based on capability and context were all proposed in an IoT middleware reference model. This proposal's system architecture[25] differs from this one in the following ways:

1. Cloud-fog computing mode Fog computing can be installed near IoT sensors. Data can be processed locally without being transferred to a cloud platform. The processed data can be transferred to intelligent processing units in the cloud to enable cloud and fog computing.
2. For system scalability, the northbound interface shields the underlying heterogeneity and supports many device types. The user application also has access to a southbound interface for development. Web Service, the open source Platform API, RESTful style interface and SDK are all supported by the southbound interfaces.
3. Integrate open-source software. Smart city subsystems may employ different open source or self-development platforms. The system uses Web of Things to facilitate platform-to-platform service exchanges.

4. The architecture that is being proposed here allows for more innovative application processing by supporting context processing through application layer business logic. This could include more complex event processing.
5. administration of distributed resources and billing for resources based on blockchain technology.

#### 4. RESULTS & DISCUSSION

The results of the survey told us some things about how Node.js is used. In the next section, we'll talk about the good things we learned from the literature review and the survey. Since Node.js came out, the work of full-stack developers has become like something out of a storybook. Because Node.js wasn't available, it was hard for a developer to learn a lot of different languages and environments to handle the whole system on both the server and the client sides. Since Node.js was made, businesses and software developers can now make applications that can handle a lot of traffic and run faster. By using Single Page Applications (SPA), the number of server calls can be cut down. This makes applications that are both easier to use and run faster. Node.js made it easy to do heavy-load tasks like visual processing and Internet GIS very quickly [26], and it can be used reliably in any industry where file sizes are large or network traffic is high. Some examples of these actions are With Node.js, these kinds of tasks can be done faster and with less bandwidth. Members of the community like that the same language is also used on the server side, while JavaScript has been used on the client side for a very long time. Node.js faces a number of problems when it comes to being used in the community and being chosen by developers and organisations over other programming languages.



**Figure 3: Node.js for Your IoT Development**

These problems can be put into two groups: how they are used and how people accept them. Without a doubt, Node.js has a lot of benefits for the community, but it also has several problems. One of these problems is that developers might use the language that is most often misused. A software enthusiast used Node.js on Raspberry Zero to make a backdoor in software. Even if the target computer has a password, it can still build backdoors into that computer and the network it is connected to. Even though there are ways to close these backdoors, some of them don't seem very likely. For example, when the user logged off, they would have to block the USB ports and close the web browser completely. And other methods, like using a secure layer on your website (https) and turning on secure flags on your cookies,

which most people might not know about, aren't used by most people, and most people might not even know about them.

There are some good things about this, but the study results show that most people in the community find it hard to get good at JavaScript for Node.js. Also, developers already familiar with other programming languages have more problems when they try to use Node.js. Even installing a server to help them with their programming is not a problem. Researchers came to this conclusion based on what they learned from the poll. Event-driven programming, input/output that doesn't stop, and asynchronous functions were some of the system's most important strengths. But the survey results show that features like event-driven programming, nonblocking input/output, and asynchronous processing are a barrier. As a result of the survey results, one problem that has come to light is that most companies are not ready to use new technologies like Node.js instead of their old ones like PHP,.Net, etc. Also, organizations don't know enough about the market, a barrier that keeps them from using Node.js for implementation.

The survey results show that there is a problem at the developer level: it is hard for them to learn how a database works and how to use an environment that uses JavaScript. This is not easy. According to the survey results, many people who answered questions about important features like event-driven, non-blocking I/O and asynchronous processing are deciding this. Also, it seems like the developers don't know enough about this subject. Training and workshops can be used to show new and experienced programmers how to use Node.js and explain the benefits of its features, such as event-driven programming, non-blocking I/O, and asynchronous processing.

This will make it easy for the community to switch to the Node.js platform. The community needs to be kept up-to-date on how Node.js is faster at doing network-related tasks, especially when handling large files over the network and multiple call backs in other languages like PHP,.NET, etc., which cause the server to be overloaded or need more memory. Also, the company should be told about the benefits of hiring Full Stack Developers based on Node.js and how they will reduce costs related to server bandwidths and hiring developers, as well as speed up Node.js applications. When working on NodeJS IOT Frameworks, it's vital to remember the importance of a logging system, standard error handling, automated testing, and continuous integration. From the beginning, it is recommended that certain policies are in place. It can be difficult to make changes to the code if the exceptions, assertions, and logging code is not structured in a way that can be managed from the beginning.

A controlled environment is preferable to a production environment for early crushing and detecting other issues (e.g., security holes, bugs). A smart city-automation system can use NodeJS IOT Frameworks. Client applications request server-side code from a central server, which receives IoT device data from sensors on the network and processes it. Web sites or native mobile applications may be clients of the system, depending on their requirements (iOS or Android). The depicts both wireless (WiFi, ZigBee, etc.) and wired IoT sensors. As detailed in this paper, a double-check module, local server programs, and methods necessary to receive and transfer data all work together to ensure data security and privacy.

It is possible to preserve minimal documentation of the called primitives by using doublecheck functions, which can be used to verify, synchronize, or test (assure functionality) (logging

functionality). Node.JS module NodeJS IOT Frameworks tries to ensure that all processes are working together in the most efficient manner possible. All of these qualities can be created and manipulated from this point onwards. It is possible to build logging infrastructures, extensible exception handling mechanisms, and run-time validation during development and production using the "specific language/API" provided by this module. Private SKY work can utilize this module as an experimental feature (but does not have other dependencies). We developed a framework for logs, exceptions, assertions, checks, and other semantic checks in the double-check module because of the distributed nature of Private Sky projects. To develop and manage all of these aspects, this module is essential. See how we've used them in our experiments.

## 5. CONCLUSION

The Internet of Things (IoT) is increasingly part of daily life. However, the development of IoT applications still faces many problems, such as heterogeneity, complex management, and other difficulties. In this paper, first, the open-source technologies NodeJS IOT Frameworks of IoT was using. We compare these technologies from different levels of technical requirements, such as device management, data management, communication, intelligent data processing, security and privacy protection; we also look at requirements of application development and deployment. Second, an IoT integrated development platform architecture for IoT applications based on NodeJS open-source ecosystem is proposed and evaluated in an industrial setting. The results show that the IoT gateway based on an open-source ecosystem had a stable and reliable system performance with specific data size and concurrency scale. Due to the fact that these conditions meet the NodeJS IoT application criteria, they can be used in nearly all sensing scenarios. For embedded devices, JavaScript is best suited because of its natural ability to support asynchronous and event-driven operations, which is why Node.js has been accepted by developers and academics working on the Internet of Things (IoT). Using JavaScript in conjunction with HTML5 to create front-end user interfaces is a popular choice for many web developers. Aside from this, Node.js has a programming style that is ideal for embedded devices as well as servers, and the Internet of Things has already adopted it (IOT). Microsoft has also approved the usage of Node.js in the development of Internet of Things apps.

## REFERENCE

1. By Hezbollah Shah & Tariq Rahim Soomro, Node.js Challenges in Implementation ,Global Journal of Computer Science and Technology: Volume 17, Issue 2, 2017.
2. Akasiadis, Charilaos, Vassilis Pitsilis, and Constantine D. Spyropoulos. 2019. "A Multi-Protocol IoT Platform Based on Open-Source Frameworks" Sensors 19, no. 19: 4217.
3. C. Damian, C. Ursache, S. Alboaie, R. Maștaleru, A. Panu and L. Alboaie, "Using Double-Check Mechanism in IoT Software Environments," 2018 International Conference and Exposition on Electrical And Power Engineering (EPE), 2018, pp. 0816-0821.
4. C. Thomás Oliveira, R. Moreira, F. de Oliveira Silva, R. Sanches Miani and P. Frosi Rosa, "Improving Security on IoT Applications Based on the FIWARE Platform," 2018 IEEE

- 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018, pp. 686-693.
5. H. C. Irawan and T. Juhana, "Heart rate monitoring using IoT wearable for ambulatory patient," 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017, pp. 1-4.
  6. R. O. Jacob, S. M. Alyaan, K. Nikitha and H. S. Niranjana Murthy, "IoT Based GPS tracking system with SOS Capabilities," 2022 International Mobile and Embedded Technology Conference (MECON), 2022, pp. 72-75.
  7. C. -P. Yi, H. -H. Chen and Y. -C. Chen, "A Smart Meter Design Implemented with IOT Technology," 2018 International Symposium on Computer, Consumer and Control (IS3C), 2018, pp. 360-363.
  8. E. Stark, F. Schindler, E. Kučera, O. Haffner and A. Kozáková, "Adapter Implementation into Mozilla WebThings IoT Platform Using JavaScript," 2020 Cybernetics & Informatics (K&I), 2020, pp. 1-7.
  9. F. Bellotti et al., "Designing an IoT Framework for Automated Driving Impact Analysis," 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1111-1117.
  10. K. Gupta, K. Kishore and S. C. Jain, "Quality assessment and drift analysis of IoT enabled ammonia sensor," 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2017, pp. 171-176.
  11. A. Kassem, M. Hamad, C. El Moucary, E. Nawfal and A. Aoun, "Med Bed: Smart medical bed," 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME), 2017, pp. 1-4.
  12. Halim, D.K., Hutagalung, S. Towards data sharing economy on Internet of Things: a semantic for telemetry data. *J Big Data* 9, 1 (2022).
  13. Luperto, M., Monroy, J., Renoux, J. et al. Integrating Social Assistive Robots, IoT, Virtual Communities and Smart Objects to Assist at-Home Independently Living Elders: the MoveCare Project. *Int J of Soc Robotics* (2022).
  14. Konstantinidis, E.I., Billis, A.S., Dupre, R. et al. IoT of active and healthy ageing: cases from indoor location analytics in the wild. *Health Technol.* 7, 41–49 (2017).
  15. Mylonas, G., Paganelli, F., Cuffaro, G. et al. Using gamification and IoT-based educational tools towards energy savings - some experiences from two schools in Italy and Greece. *J Ambient Intell Human Comput* (2021).
  16. Ceccarini, C., Mirri, S., Salomoni, P. et al. On exploiting Data Visualization and IoT for Increasing Sustainability and Safety in a Smart Campus. *Mobile Netw Appl* 26, 2066–2075 (2021).
  17. [https://wiki.tizen.org/Tizen\\_RT\\_Long-term\\_Goals](https://wiki.tizen.org/Tizen_RT_Long-term_Goals)
  18. Corno, F., De Russis, L. & Montanaro, T. XDN: cross-device framework for custom notifications management. *Computing* 101, 1735–1761 (2019).
  19. Bosse, S., Lehmus, D. Material-integrated cluster computing in self-adaptive robotic materials using mobile multi-agent systems. *Cluster Comput* 22, 1017–1037 (2019).
  20. Zhang, X. The use of ethereum blockchain using internet of things technology in information and fund management of financial poverty alleviation system. *Int J Syst Assur Eng Manag* (2022).

21. Hung, LP., Yang, DY., Wu, ZJ. et al. Constructing a Search Mechanism for Dementia Patient Based on Multi-Hop Transmission Path Planning and Clustering Method. *Mobile Netw Appl* (2022).
22. Rajawat, A.S., Rawat, R., Mahor, V., Shaw, R.N., Ghosh, A. (2021). Suspicious Big Text Data Analysis for Prediction—On Darkweb User Activity Using Computational Intelligence Model. In: Mekhilef, S., Favorskaya, M., Pandey, R.K., Shaw, R.N. (eds) *Innovations in Electrical and Electronic Engineering. Lecture Notes in Electrical Engineering*, vol 756. Springer, Singapore.
23. ITU-T Recommendation Y, 2066. Common Requirements of the Internet of Things. Available online: <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12169&lang=en> (accessed on 18 July 2017)
24. Fit IOT-LAB. Available online: <https://www.IoT-lab.info/> (accessed on 18 June 2018).
25. Da Cruz, M.A.A.; Rodrigues, J.J.P.C.; Al-Muhtadi, J.; Korotaev, V.; Albuquerque, V.H.C. A Reference Model for Internet of Things Middleware. *IEEE Internet Things J.* 2018, 99, 871–883.
26. Yoon, C.; Choi, H.; Cho, J.; Kim, Y.W. CoAP over BLE-GATT for OCF. In *Proceedings of the International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC, Jeju, Korea, 18–20 October 2017*; pp. 32–34.