Leveraging Representational State Transfer (REST) API And Middleware In Ensuring The Security Safeguards Of Internet Of Things (IOT) Devices While Securely Connecting The Dots[1]

Vanshika Goel
Vishwakarma University, Pune, Maharashtra, India

ABSTRACT

The Internet of Things, or IoT, is a technologically disruptive force that is growing, impacting, and having capabilities that are unimaginable.

We cover some basic IoT ideas and the use of REST API in IoT systems, whose technology may count and record anything. We also emphasize the idea of middleware, which serves as a bridge between these devices and the cloud. The emergence of novel Internet of Things applications on cloud platforms has resulted in fresh risks to data security and privacy. Consequently, it is necessary to implement a secure Internet of Things system that prevents hackers from entering the network through IoT devices and secures data while it is being transferred from IoT devices to cloud storage. We describe in detail how Representational State Transfer (REST) API enables connected devices to be safely exposed to cloud apps and users. The main functions of middleware in the suggested approach are to conceal details, expose device data over REST, and serve as a user interface for interacting with sensor data.

INTRODUCTION

Through the Internet, electronic devices embedded in common things are connected and may exchange data thanks to the Internet of Things. There are two benefits: first, we can enable our computers to collect environmental data without human assistance, and second, by analyzing the data, we can lower costs, losses, and luxury. There is contact between the digital and physical worlds thanks to the Internet of Things. Actuators and sensors allow the digital and physical worlds to communicate with one another. These sensors gather data that needs to be processed and saved. Processing of data can occur at the network's edge, on a distant server, or in the cloud. An Internet of Things object's storage and processing capacities are limited by the resources at its disposal, which are limited by factors including size, energy, power, and computational capacity. For these systems to have the necessary functionality, IoT middleware is required.

[1] *How to cite the article:* Goel V., (September 2023) Leveraging Representational State Transfer (REST) API And Middleware In Ensuring The Security Safeguards Of Internet Of Things (IOT) Devices While Securely Connecting The Dots; *International Journal of Advanced Engineering*, Jul-Sep 2023, Vol 6, Issue 3, 37-48

A. RESTful machine-to-machine and IoT communication

APIs make it possible to securely expose the connected device to users. In the current web, RESTful APIs are frequently utilized. Typically, JSON or XML is used for data transport over HTTP. It's a useful model for systems that are heterogeneous. The device information is readily available thanks to REST API. They can agree on a common method for generating, reading, editing, and erasing this data. The REST query requests will receive input from all of these actions. Authorization can be managed and delegated using REST APIs. To thwart man-in-the-middle attacks, the server may authenticate to the API and the API can authenticate to the server.

B. The architecture of IoT middleware

Having a middleware platform that acts as a link between objects and cloud apps is one method to manage such heterogeneous applications. Application programming interfaces (APIs) for communication, data processing, computing, privacy, and security are provided by middleware, which also bundles and abstracts hardware.

An overview of middleware's function in the Internet of Things is shown in Figure 1.The items themselves, the local network, which may include a gateway, middleware, and the cloud (for user access management, business data analysis, etc.) are the four primary parts of an IoT system in the broad category.

SECURITY DIFFICULTIES WITH IOT

As IoT applications become more prevalent in daily life, sensitive and private data about individual users is being acquired.

In each of these settings, privacy and security concerns need to be taken into consideration. Health data is extremely important for maintaining personal privacy, much like in the healthcare industry, thus it shouldn't be accessed by unauthorized parties.The October 2016 massive denial-of-service (DDoS) attacks on DYN's servers, which knocked down numerous well-known US online businesses, showed us what may happen when attackers are able to use up to 150,000 insecure Internet of Things devices as malicious terminals [1].

Understanding the security requirements for each of the major IoT system components—devices, users, middleware/IoT gateway, communication channel, and cloud applications—is essential to comprehending the overall approach to data protection.

A. Security Challenges in Constrained IoT Devices:

• Devices that are part of IoT systems have limited resources.

• Public key infrastructure is not appropriate for IoT contexts because high key sizes make ciphertext calculation computationally demanding.

• Why Updating the system becomes challenging once it has been compromised. Furthermore, it is not advisable to restart or reinstall software, replace parts or subsystems, or shut down possibly compromised systems for many IoT devices since this could cause significant disruption and loss of revenue. Installing a secure firewall on IoT devices is also neither viable or practicable. [2].

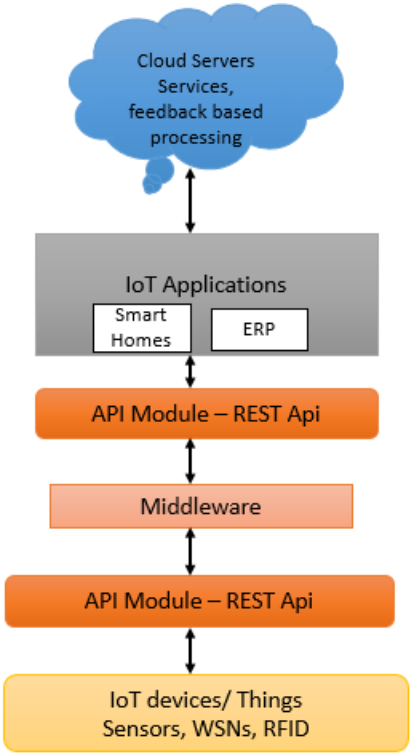

Figure 1: Architecture of IoT Middleware

B. The Need of Reliable Devices

In order to verify and allow devices to share data, IoT middleware must maintain a trust relationship with those devices. It must impose authentication before communicating with any device so that data origin verification is possible. The unique identities given to these devices prevent the sharing of security credentials amongst them.

C. Linking Middleware to IoT devices.

Connecting a lot of different types of smart devices is necessary, such as wearables, linked cars, smart cities, etc. Usually, an IP stack is used to link Internet of Things devices to the Internet. These days, this stack is extremely complicated and demands a lot of processing power and memory from linked devices.

D. Communication channel security.

To maintain data integrity, Internet of Things data must be protected both in transit and at rest. Security solutions are put into practice with the intention of identifying unauthorized access points and thwarting malevolent attacks on the communication layer. defending against assaults such as guessing an ID offline, replay attacks, unauthorized logins, user anonymity, and anonymity of sensor nodes. An overview of the security specifications for the various IoT system pillars is provided in Figure 2.

CONNECTED WORK

To meet the various security needs of an Internet of Things system, there are numerous options available. In the resource-constrained setting of an IoT system, mutual authentication between the IoT device and gateway is the primary requirement.

Many IoT authentication solutions have been proposed recently. For IoT devices, the author of [3] proposed a strong anonymity-preserving authentication protocol that allows tag and reader mutual authentication via the server. Elliptic Curve Cryptography (ECC) is the authentication method used in this technique.

Cryptographic algorithms based on ECC have already been proven to be secure. The challenge of resolving these two issues is what ensures ECC security:

•Let E be an elliptic curve over a finite field. This is the Elliptic Curve Discrete Logarithm (ECDL) issue. Let P and Q represent Zq's points (modulo q). Compute the special integer α that belongs to Zq such that Q = α P is a challenging task.

• Decisional Diffie-Hellman (ECDDH) problem with an elliptic curve: There are three points in G: aP, bP, and cP. Checking if abP = cP is difficult.

In most cases, the user must authenticate through the gateway in order to gain access to the sensor or sensor data. The user, the gateway, and the sensor will be the three key parties involved. A verifiable, proven, and privacy-preserving user authentication strategy for wireless sensor networks (WSN) is proposed by the authors in [4]. The authors highlight the insecurity of the Hsieh and Leu's method [13], citing a number of security flaws such as insider attacks, offline password guessing attacks, user forging attacks, and sensor capture attacks. A brand-new two-

factor authentication system for WSN that is likewise based on ECC is introduced. The authorization, confidentiality, and integrity requirements for Internet of Things security are met by this approach.

For user authentication, several multi-factor authentication systems are recommended. A combination of factors, including smartcards, biometrics, and passwords, is employed in the identity authentication process. Similar to [5], a three-factor authentication system was suggested for WSN (2018). Passwords, smartcards, and fingerprint identification are the three factors. In this instance, the gateway is meant to act as a reliable intermediary between the user and the sensor. This technique authenticates the user to access the sensor, successfully registers the user and sensor on the gateway, and gives the user the ability to change their password.

Additionally, a few incredibly light-weight authentication systems that just require basic bit-wise operations on tags (such as XOR, AND, OR, etc.) have been developed by researchers. As a result, it has relatively low costs associated with communication and storage [6].

In order to accommodate devices with limited resources for authentication, Razouk et al. proposed a novel Security Middleware Architecture based on Fog Computing and Cloud [7]. Proposed middleware gives IoT-constrained devices access to extra processing power and improves their capacity to conduct secure communications. For ease of use, this model is built on well-known technologies like REST API and "Constrained Application Protocol (CoAP)".

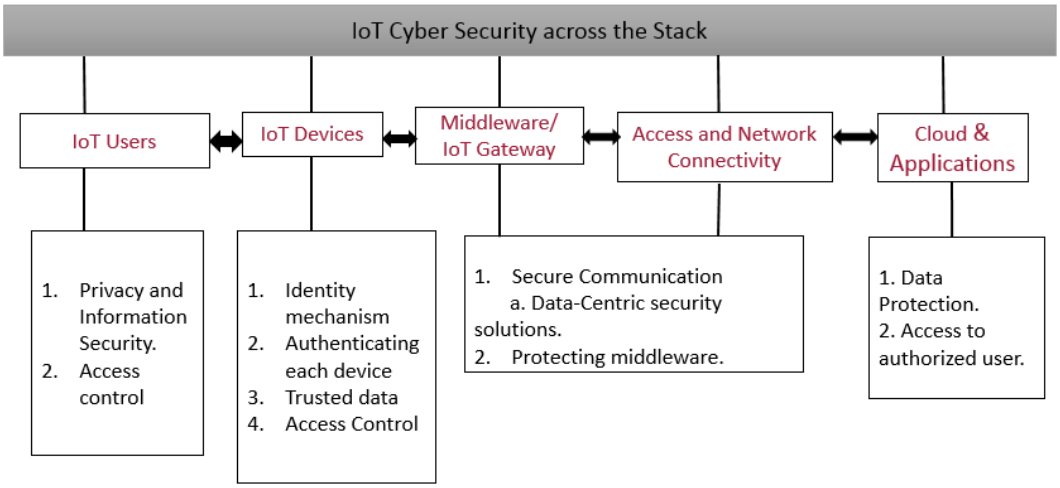

Figure 2 shows the Internet of Things security model.

RECOMMENDED MODEL

IP stacks are commonly used by Internet of Things devices to connect to the Internet. The connected IoT devices must have a lot of power and memory because this stack is extremely

complicated. The suggested model involves the localization of IoT devices over non-IP networks and their connection to the Internet via an intelligent gateway. For IoT devices, this gateway serves as an interface to the internet. Because of the smart gateway, devices don't need inbound ports and can be concealed in an enterprise behind several firewalls.

A middleware architecture that can adjust to the needs of an application is provided by the suggested approach. Database management, device identification, and registration are handled by middleware.

Additionally, it guarantees data confidentiality and privacy. Following authorization and authentication through REST API, the stored data is made public. The general scenario is summarized in Figure 5.

Numerous protocols for authorization are available. OAuth, for instance, is an open authorization protocol that enables middleware resource access via tokens, password, and username. This onboard flow's architecture is described in Figures 3 and 4.

Step 1: To register a device, an authorized user must first create an online account using middleware.

Step 2: The gateway accesses the publicly accessible REST API to validate the device request and payload whenever an authentication request originates from an IoT device.

With its own credentials, the gateway will now request access to the exposed API.The input parameters for the request will be the secret key and the gateway ID. The gateway will be authorized and authenticated by the API, which will also validate the request. In a REST API, authentication is required for every method.

A response in encrypted form with the device details is sent back to the gateway once it has been authorized.

Following verification, the device receives an access token from the gateway, enabling it to submit real-time data to the gateway.

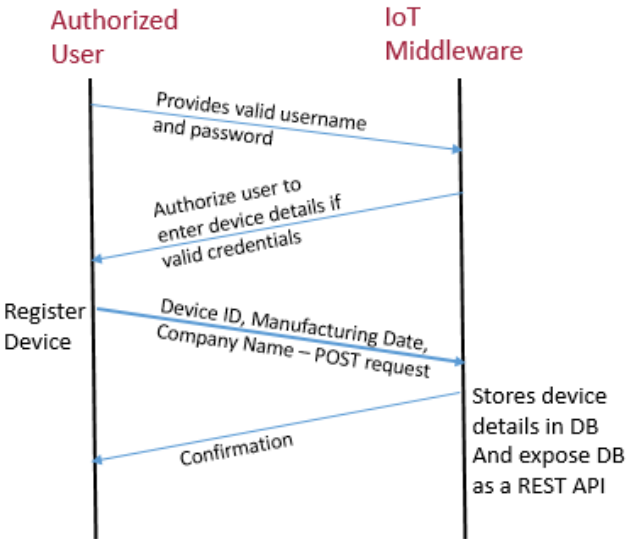

Figure 3: Device Registration

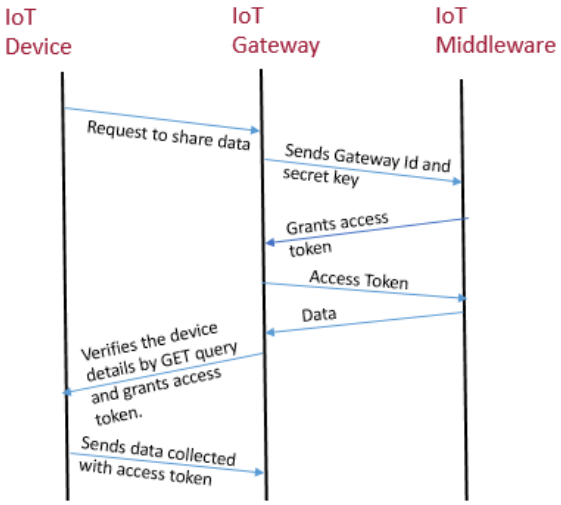

Figure 4: Data sharing

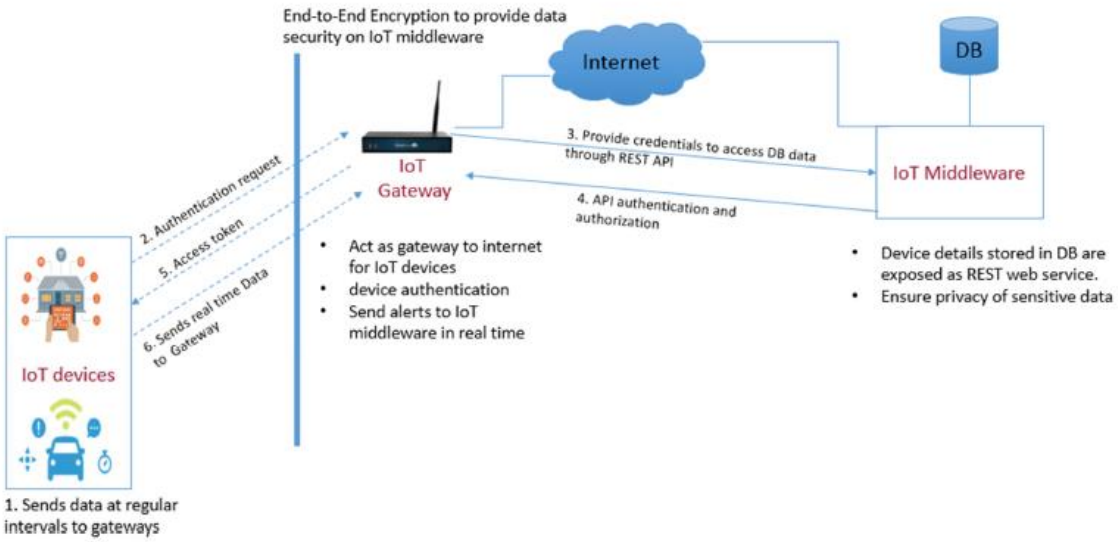

Figure 5: Proposed System Model.

A. Outlining the necessary REST endpoints.

In an Internet of Things environment, moving data safely, swiftly, and efficiently is really what's needed. The heart of the Internet of Things is an API, which makes this easier. HTTP is used by RESTful web services to transfer machine-readable file formats like JSON and XML as well as for machine-to-machine communication. For various operations, we have developed distinct REST endpoints.

B. Getting to know REST endpoints

IoT REST API-Inbound has authorization and authentication enabled. The gateway is granted access to the URLs once its operations have been authenticated correctly. Gateway will now create a jQuery client in order to use a RESTful web service. Every service request will be viewed at a specific URL. In response, the service will provide a JSON representation of the device's information.

• RESTful resource to obtain Specifics of the Device:- a. Obtain an access token from the middleware to access API-inbound.

REST URL (Access token URL)	https://middleware-0ff5f6bf8404dsdca4012sa1c0426f689905.identity.c9dev1.oc9qadev.com:443/oauth2/v1/gateway-id
Request	POST
Response	{ "access_token":" sdjdwkwsjdjnjsncjdcnhwdnjsnjnjdncjddjiwdjjdnjndjcxmsmcxsncwenddscdcnwdeidmd

	mlkwkqnjqwdnjwdndkwefjuiefjrnfjfnjefhhyrfbrhfb", "token-type":"bearer", "expires_in":"3600" }
Mandatory fields	Gateway-id, secret key

b. To view device information following permission

The sample code to use the REST API is shown in Figure 6. The provided module is a straightforward JavaScript function that accesses a REST service at a given URL by using jQuery's $.ajax() method. Should it is successful, the received JSON will be assigned to the variable data.

REST URL	http://localhost:8080/test/webresources/com.mycompany.test.devicedetails/{id}
Request	GET (application/ json)

Response In JSON	{
	"deviceIdentifier":"a","
	deviceManufacturingDate":"2018-12-16T18:30:00Z[UTC]",
	"uid":"abcd12344"
	}

Mandatory fields	Device Identifier

Figure 6: An example of an AJAX request sent via URL

REVIEW OF THE SUGGESTED MODEL

Our study presents a safe Internet of Things infrastructure that guarantees end-to-end security from IoT devices to IoT applications. By examining the integrity of each component, we can assess the system's security.

• Internet of Things devices don't communicate or interact with the outside world. They are linked to a gateway, which serves as the Internet of Things devices' interface. Because of the smart gateway, devices don't need any inbound ports and can be concealed in an enterprise behind several firewalls. Thus, there is no possibility of an attacker getting access to these devices.

• An IoT gateway serves as a middleman between middleware and IoT devices. Gateways have the ability to securely communicate all data and call REST APIs.

• Using conventional cryptographic techniques ensures secure communication between the middleware and IoT gateway. Since there are no resource constraints for either party, lightweight algorithms are not required.

• REST APIs handle authorization and authentication, which simplifies the process and ensures compliance with industry standards.

```
$.ajax({
        type: "POST",
    url:  "http://localhost:8080/test/webresources
/com.mycompany.test.devicedetails",
    data: JSON.stringify({
    deviceidentifier : self.newDeviceId(),
    devicemanufacturingdate : self.newMDate(),
    uid : self.newUid()
            }),
    headers: {
    'Content-Type': 'application/json'
            },
    success: function() {
                console.log(data);
            },
    error: function(err) {
                console.log("AJAX Error: " + err);
            }
          });
```

CONCLUSION

We suggested a middleware design that offers contributors who upload sensing data an end-to-end security solution. With this method, data can be encrypted from beginning to finish to protect it while it's in transit. All IoT system restrictions are taken into account in the suggested middleware solution. Data transmission and communication are facilitated by REST API. By exposing REST API, giving users an interface to register their IoT devices, and enabling safe access to the data collected by the devices, middleware effectively supports IoT development.

REFERENCES

[1] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84.

[2] Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. IEEE Internet Computing, 21(2), 34-42.

[3] Tewari, A., & Gupta, B. B. (2018, January). A robust anonymity preserving authentication protocol for IoT devices. In Consumer Electronics (ICCE), 2018 IEEE International Conference on (pp. 1-5). IEEE.

[4] Wu, F., Xu, L., Kumari, S., & Li, X. (2017). "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security". Journal of Ambient Intelligence and Humanized Computing, 8(1), 101-116.

[5] Li, X., Niu, J., Kumari, S., Wu, F., Sangaiah, A. K., & Choo, K. K. R. (2018). "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments". Journal of Network and Computer Applications, 103, 194-204.

[6] Chien, H. Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE transactions on dependable and secure computing, 4(4), 337-340.

[7] Razouk, W., Sgandurra, D., & Sakurai, K. (2017, October). "A new security middleware architecture based on fog computing and cloud to support IoT constrained devices".In Proceedings of the 1st International Conference on Internet of Things and Machine Learning (p. 35). ACM.

[8] "REST API for Oracle Internet of Things Cloud Service", docs.oracle.com/en/cloud/paas/iot-cloud/iotrq/QuickStart.html.

[9] Sfar, A. R., Natalizio, E., Challal, Y., & Chtourou, Z. (2018). "A roadmap for security challenges in the Internet of Things. Digital Communications and Networks", 4(2), 118-1375.

[10] Fremantle, Paul & Scott, Philip. (2015). A security survey of middleware for the Internet of Things.10.7287/PEERJ.PREPRINTS.1241

[11] Ayoade, G., El-Ghamry, A., Karande, V., Khan, L., Alrahmawy, M., & Rashad, M. Z. (2018). Secure data processing for IoT middleware systems. The Journal of Supercomputing, 1-26.

[12] He, D., Chen, J., & Zhang, R. (2012). An efficient and provablysecure certificateless signature scheme without bilinear pairings. International Journal of Communication Systems, 25(11), 1432-1442.

[13] Hsieh, W. B., & Leu, J. S. (2014). A Robust ser Authentication Scheme sing Dynamic Identity in Wireless Sensor Networks. Wireless personal communications, 77(2), 979-989.

[14] Choi, Y., Lee, D., Kim, J., Jung, J., Nam, J., & Won, D. (2014). Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors, 14(6), 10081- 10106.