IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

Middleware to Address Heterogeneity Problem in IoT

Chaitali Parmar[1], Atharva Todankar[2], Shrihari Wayal[3], Prof. Jagat Gaydhane[4]
Students, Department of Information Technology[1,2,3]
Faculty, Department of Information Technology[4]
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India

Abstract: *The Internet of Things (IoT) envisions a future in which digital and physical things or products (e.g., smartphones, TVs, cars) can be linked through the use of appropriate data. to enable a variety of applications using information and communication technology as well as products and services The characteristics of the Internet of Things include an ultra-large scale network of things, as well as device and network-level heterogeneity and a high number of unintentionally created occurrences things like this will make developing a wide range of applications and services a difficult endeavor. In general, middleware can make a process easier. Integration of heterogeneous computing and development process promoting interoperability within telecommunication equipment a wide range of applications and services. In this paper, we will discuss Smart System for Device Interoperability was offered as a solution. It serves as a hub, allowing disparate devices to communicate with one another. Regardless of their differences, they can communicate through it. Protocols and other forms of communication Detailed. The proposed system's architecture is presented, as well as each of its component is well explored.*

Keywords: Heterogeneity Problem.

I. INTRODUCTION

Smart home devices have been around for decades and have been regarded as promising realizations of the Internet of Things (IoT) wherein, users easily perform tasks involving diverse sets of devices in their home without the need for painstaking configuration and custom programming. While modern homes have many network-capable devices, applications that coordinate them for cross-device tasks have yet to appear in any significant numbers and also, the lack of dominating standards for IoT communication, control, and data management results in highly fragmented smart home systems consisting of proprietary solutions provided by each device vendor. Hence, users are required to use different control interfaces, e.g., mobile apps, to interact with smart appliances in their homes, or are forced to use devices sold by a single vendor to achieve the best user experience. The project aims to solve this issue by introducing interoperability among the various different vendors and making the heterogeneous devices work together via a single user interface

The core idea behind the Internet of Items (IoT) is to link things through the internet, which can include sensors, actuators, tags, and mobile phones. The Internet of Things can be used in all aspects of human existence, including health, smart homes, smart cities, energy, and logistics. The Internet of Things ushers in a new era of physical contact in which everything around us may be connected to the internet at any time and from anywhere. According to a recent projection by the International Data Center (IDC), the Internet of Things and its associated ecosystems would be worth $7 trillion by 2020.

It is now achievable thanks to recent advancements in communication technology, such as lower power usage, simple data connectivity, and lower device costs.

The objective of the Internet of Objects is to connect all things in such a way that they can gather, process, and share data with one another. Regardless of their communication technology, all devices must be able to communicate and interact with one another.

The high degree of heterogeneous heterogeneity is one of the primary challenges that IoT faces. Different communication protocols, technology, and hardware are used by different devices. In fact, one of the primary problems to be addressed while establishing and integrating new IoT ecosystems is interoperability [3]. Interoperability issues can reduce the IoT's benefits by up to 40%. Interoperability in the Internet of Things refers to the capacity of two components or systems to share and utilise data with one another.

Interoperability can be created at multiple levels in the IoT context, such as protocol interoperability and data interoperability. Protocol interoperability refers to the ability to connect multiple network technologies directly.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

The syntax and semantics of data are linked to data interoperability. To process and understand data, the communications transferred between two linked devices must also be interoperable. Data grammar and encoding technologies are unique to each communication protocol.

By resolving the interoperability issue, the Internet of Things ecosystem will be able to grow in the right direction and achieve the true meaning of IoT, which is hidden in data transmission and understanding. In this paper, we will look upon We presented a Smart System for Device Interoperability in the Internet of Things (SI2oT) that facilitates network, syntactic, and semantic interoperability amongst heterogeneous devices. The decision-making components that leaned through Naive Bayesian Classifier decide on protocol conversion, ontology, and transmitting routes. The data is then converted into the destined protocol and transferred to the destination device.

JSON (JavaScript Object Notation) is a language-independent data-exchange text format that is simple for humans to write and read and simple for machines to generate and parse. It employs programming concepts that are known to programmers who work with the C family of languages. It is made up of name/value pairs that are represented as objects and can be studied and used by humans or computers quickly and easily, making it preferable to the web in terms of data exchange.

The format for data shared between the Raspberry Pi and the mobile application in this experiment was JSON.

{'led':1}

The JSON string seen above is an example of what the Raspberry Pi can send or receive. It is made up of pairs of attribute names and their values.

Smartphones and microcontrollers are used to control the majority of smart home systems. Using wireless connection technology, a smartphone application is utilised to control and monitor home functions. We look into it. Integration of IoT services into a smart house concept. Smart sensors and actuators can be networked by incorporating intelligence into them. Items with the appropriate technology, allowing for easier interactions with smart devices things that are easy to access in several areas, increasing. Data exchange efficiency, compute power, and storage space.

II. Difficulties In Heterogeneity

In this section, we discuss the various types of heterogeneity among IoT devices on the market, which poses obstacles in the development of truly linked smart homes.

Communication interface: The protocol and architecture of the communication interface for smart appliances differs greatly. Although most smart appliances provide remote control through HTTP (or HTTPS) over WiFi, control commands are sent in different ways for each item. Many devices use JSON messaging formats for RESTful APIs, whereas others, such as WeMo devices, solely use SOAP messaging to exchange XML messages. Communication architecture is another facet of heterogeneity. There are basically three possible communication designs. Appliances in the home network are accessible directly via local IP address; appliances require a dedicated "hub" to communicate.

Venstar and Radio Thermostat thermostats, for example, can be controlled from devices connected to the same WiFi network. Only a gateway hub that implements a WiFi interface for remote control may operate Wink, SmartThings, and Philips HUE lighting. Nest's gadgets and Ecobee's thermostat are, on the other hand, operated remotely via their own cloud services. Security and user authentication systems are two further examples of communication heterogeneity. Users of Nest and Ecobee thermostats must use an OAuth2 authorisation token, whereas Philips HUE relies on a username/password combination.

III. Background

The world's first Internet of Things device was created at Carnegie Melon University in the early 1980s. A group of university students devised a method to have their campus Coca-Cola vending machine report on its contents over a network, saving them the trip if the machine was out of Coke. Microswitches were put in the machine to report on how many Coke cans were available and if they were cold.

What is the Internet of Things (IoT)?

The term "Internet of Things" refers to devices that can communicate with the network but are not often assumed to have an internet connection. As a result, the Internet of Things (IoT) is a network of interconnected devices. internet-connected

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

items that can collect and transmit data over a wireless network without the use of a person. Wearable fitness trackers (like Fitbits) and IoT healthcare applications, voice assistants (Siri and Alexa), smart cars (Tesla), and smart appliances are all instances of IoT in action (iRobot).

A system that monitors and/or controls house features such as lighting, climate, entertainment systems, and appliances is referred to as smart home automation. Wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface that can be accessed off-site over the Internet are all options for controlling the system. It essentially simplifies a task that we would have had to complete manually.

Heterogeneity refers to the fact that we communicate with our IoT devices through a variety of service providers, but each request demands a different application. To avoid future issues, all devices or defined devices should be incorporated into a single protocol type, allowing them to be controlled by any Android app. To communicate with IoT devices, they merely need to use our API provider. The proposed middleware model should provide data compatibility, bandwidth management, ensure connectivity between heterogeneous devices, and solve security problems in order to efficiently exploit the capabilities of current communication technologies and provide more flexible, reconfigurable, and efficient networks.

To control the many electrical devices connected to the system created for the home automation project outlined in this paper, a control unit, a computer, is required. The Raspberry Pi is a credit-card-sized computer that can be hooked into a monitor and uses a conventional keyboard and mouse to teach people of all ages how to programme.

Because this project focuses on IoT-enabled smart home automation, the smart house concept must first be grasped. To be able to operate it, smart houses combine typical electronics found in homes. Originally, the technology was designed to regulate environmental systems, but nowadays, practically any electrical component may be integrated into a smart home system.

IV. Related Work

Multiple solutions, such as middleware, gateways, and hub-based systems, have been proposed to address the issue of heterogeneity. The creators Wireless Sensor Networks (WSNs) and mobile communication networks are proposed as IoT gateways. These gateways are in charge of protocol conversion in order to make data transmission and device management easier. For specialised domains, some researchers offered gateway solutions. Gateways for e-health and for the home, for example, are offered. Through the usage of WebSockets, the e-health gateway allows interoperability between WiFi, Bluetooth, and the underlying protocols.

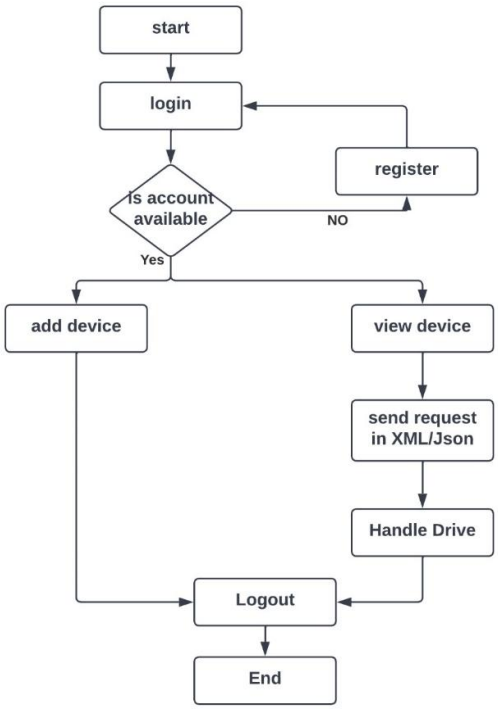
A recent and comprehensive survey on IoT middleware is presented in. Functional, non-functional, and architectural needs are the three basic areas of IoT middleware requirements. For a better understanding, the functional, non-functional, and architectural requirements for IoT middleware are explored in detail. Then, based on their design, Middleware solutions are classified. All current solutions are evaluated in light of the requirements. The existing issues are then explored in order to prepare future researchers. The authors looked at the level of heterogeneity in existing middleware as an architectural need for IoT middleware. In order to make heterogeneous devices interoperable in an IoT ecosystem, a smart middleware solution is required.

V. Design Goals

After deciding on this issue, the first thing to consider was the system architecture. How will appliances and gadgets interact with a web/mobile application, receive commands, and report their status? Should it be a server-client protocol, with a direct link between the user's devices and the programme they're using? How will a direct internet connection be established? Should the client and server connect via a medium channel through which the messages are relayed before reaching the client or server? These inquiries led to the investigation of various ways and services. To connect multiple sensors and gadgets together as well as to the internet, the Raspberry Pi was chosen as the main control unit.

We want to create a framework for controlling, monitoring, and deploying smart IoT devices as well as designing linked smart-home applications. For users and application developers, the framework should enable a vendor-neutral, device-agnostic ecosystem. The following are the design principles.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252



s

5.1 Smartphone-Based Implementation

One of the important design ideas is to take use of the widespread availability of smartphones (or tablet devices) in practically all homes, as well as users' familiarity with such devices, to make the transition from traditional to smart homes as painless as possible. Furthermore, we rely solely on the out-of-the-box capabilities of commercial-off-the-shelf (COTS) cellphones, with no kernel-level tweaking or rooting required. We present a generic approach for defining the features and communication interfaces of smart appliances in a unified structure known as the driver. According to the control interfaces supplied by each device vendor, a driver defines each variable and configuration option, including access control settings for each variable, the range of permitted values for each variable, the details of communication protocols, and authentication procedures. The framework should also include a mechanism for dynamically loading required drivers at runtime, either from local storage or from the cloud, so that applications built on top of it can operate devices with ease.

VI. IMPLEMENTATION

Smart Home Automation is a term used to describe a system that monitors and/or controls several aspects of a home, including lighting, climate, entertainment systems, and appliances. Wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface that is also available off-site through the Internet are all options for controlling the system. It essentially simplifies a task that we would otherwise have had to complete manually.

6.1 What is the Internet of Things?

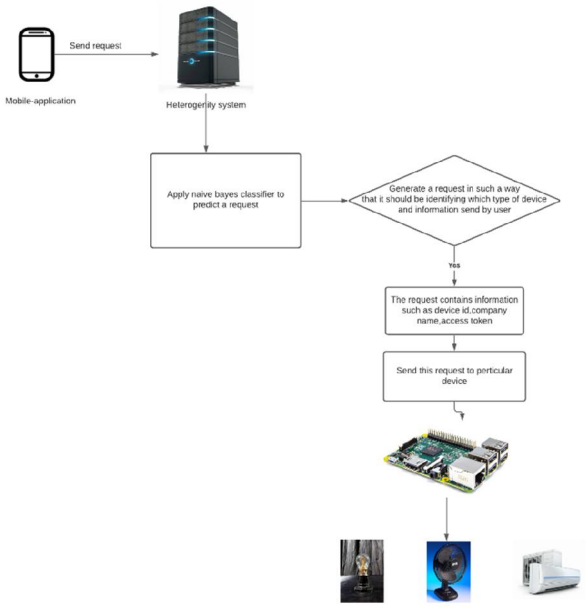
The term "internet of things" refers to gadgets that can communicate with the internet but aren't typically assume have one. So, the Internet of Things (IoT) is a network of interconnected devices. We will use SOAP and REST architecture to solve this. REST transfers data in Json format, whereas SOAP sends data in XML format.

Our project uses a unique XML format to define and assemble device specifications and control interfaces (i.e., device drivers). Users can build their own smart-home system with ease, without having to worry about heterogeneity or interoperability between multi-vendor IoT devices.

JSON (JavaScript Object Notation) is a language-independent data-exchange text format that is simple for humans to write and read and simple for machines to generate and parse.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

It employs programming concepts that are known to programmers who work with the C family of languages6. It is made up of name/value pairs that are represented as objects and can be examined and used by humans or machines. As a result, it is superior in terms of data sharing via the internet.



The format for data shared between the Raspberry Pi and the mobile application in this experiment was JSON.

{'led':1} .

The JSON string seen above is an example of what the Raspberry Pi can send or receive. It is made up of pairs of attribute names and their values.

To set values on the appliance (update the appliance configuration), the project system sends an HTTP request to one or more fields, perhaps with additional headers, and possibly with a body message, using HTTP POST / PUT methods as stated. Similar to the read action, if the new values for the variables are set by a user from the user interface or by the app, project system converts the variable values to the accepted format and range provided for each variable in the XML driver. It then produces the message and transmits it to the target device using the settings given in the publish action unit. Upon receiving a command acknowledgment from the device, the value of the variable is updated in the database if necessary.

The Raspberry Pi and the mobile application are both data transmitters and receivers. A mobile application written with Ionic was used to operate the LEDs, which were connected to the Raspberry Pi through GPIO pins and integrated in the model. The application has a "Light" tab with a button that toggles between "ON" and "OFF" when clicked. The button's default value displays the current state of the LED lights. If the button displays the value "OFF," the LEDs are now turned off. When the button is pressed, however, the value toggles to "ON," sending a JSON string with the value 1 for the "led" property, 'led':1. This channel's JSON string is subsequently transmitted to its destination.

To comprehend the project's operation, it explains the entire working structure as well as the integration of various equipment. The arrows depict the progression of initiatives, beginning with user smartphones and ending with modifying the condition of electronic gadgets. The user can communicate with our Raspberry in one of two network modes. If the user is within their home (on their local network), they will be able to utilise all of the IoT services without having to connect to the internet cloud. Because everything happens locally, this will also result in speedier device-to-user communication.

If the user is located outside of the home, anywhere in the world, the second network mode is employed. The user then connects to the internet for the first time. The completed request is forwarded to the appropriate Raspberry for processing. Each user's services are handled based on the credentials used to initiate the request. If the user is not connected to the home network, APIs are called from the cloud. If the user is connected to the home network, the same APIs are stored on the Raspberry Pi server.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

JSON is used to share data between the application and the server database. Multiple hashing algorithms are used to secure APIs. The Raspberry Pi GPIO [44] pins are used to change the status of any device. The Raspberry Pi gets the server's request. Raspberry Pi responds to devices in accordance with the user's desire. At servers, a database of each request generated by a specific user is kept. On his smartphone, the user can view the whole history of requests that have been processed. Sensors put throughout the house continuously update its status and respond to changes on the Raspberry Pi server. As a result, the Raspberry Pi server syncs all data to the database and updates the values on the mobile app.

VII. Conclusion

One of the most essential IoT applications is home automation. It makes life easier and more enjoyable for everyone. The created model was used to implement and evaluate a technique for developing an IoT software-based smart home automation system in this project. It focuses on the safety and security aspects of home automation by utilizing some of the most latest technology available. The following technologies were utilized to complete this project: The Raspberry Pi, a credit-card-sized computer, served as the project's main control unit, connecting various devices and sensors. In which JSON was used as the data exchange format.

A complete assessment of these middleware systems has been published based on these needs, with a focus on current, state-of-the-art research. Finally, open research questions, problems, and suggested future research directions are discussed.

We present a centralized real-time event processing application that can coordinate and handle enormous data flows in a balanced and effective manner, leveraging the strengths of each component. We go over the advantages and benefits of each individual component, as well as potential complements that can be achieved by combining it with other components, resulting in new benefits from the compound system as a whole. Because these components are still in the early stages of development, their integration may change, resulting in a stable paradigm that leads to the development of a new generation of infrastructure and applications. We will continue to consider additional components to be added as we track each component's progress and impact on the integrated compound, resulting in new service models.

A complete assessment of these middleware systems has been published based on these needs, with a focus on current, state-of-the-art research. Finally, open research questions, problems, and suggested future research directions are discussed.

VIII. Future Scope

The creation of diverse IoT applications and services necessitates the use of middleware. Many projects have been launched to address this issue. The ideas are broad, encompassing a wide range of middleware design methodologies and addressing a wide range of requirements. This paper contextualizes these initiatives and provides a broad overview of the topic.

The existing middlewares are classified into the following categories based on their design approaches: event-based, service-oriented, agent-based, tuple-space, VM-based, database-oriented, and application-specific. Each category has a number of middleware proposals, which are presented in the appropriate order. The majority of these proposals (Tables I–III) have been assessed and described in terms of their supported functional, nonfunctional, and architectural requirements. The summaries demonstrate that each middleware meets two or more of the stated needs from each requirement category fully or partially (for example, PRISMA partially provides code management using code allocation). None of the requirements are supported by any of the options.

Middleware is required for the development of a wide range of IoT applications and services. To address this problem, a slew of initiatives have been developed. The concepts are broad, spanning a variety of middleware design techniques and addressing a variety of needs. This paper places these activities in context and gives a general overview of the subject.
Although existing middleware solutions address many of the requirements associated with middleware in IoTs, some requirements and related research issues, such as scalable and dynamic resource discovery and composition, system-wide scalability, reliability, security and privacy, interoperability, intelligence integration, and context-awareness, remain relatively unexplored. Future research in these areas has a lot of potential.

Predetermined and deterministic composition mechanisms, on the other hand, will not scale effectively in ultra-large, dynamic IoT contexts. Because of its mobile and distributed nature, the agent-based design approach is good at resource and code management, but it makes security and privacy solutions problematic. Due to their data redundancy properties,

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

middlewares based on tuple-spaces are distributed and substantially more dependable than others. Tuple-space-based middlewares, like agent-based alternatives, will have security and privacy issues.

REFERENCES

[1]. R.A. Ramlee, M.A. Othman, M.H. Leong, M.M. Ismail, S.S.S. Ranjit, "Smart home system using android application," International Conference of Information and Communication Technology (ICoICT), August 2013

[2]. Colin Dixon, Ratul Mahajan, "An Operating System for the Home", International Research Journal of Engineering and Technology ISSN:2395, Volume-5

[3]. M.M. Moazzami, G. Xing; D. Mashima; W.P. Chen, U. Herberg, "SPOT: A smartphone-based platform to tackle heterogeneity in smart-home IoT systems", 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Volume-116, February 2017

[4]. R. Gosalia, D. Gala, Ami Munshi, "Android Based Home Automation System", International Journal of Scientific & Engineering Research Volume 9, Issue 11, November 2018

[5]. H.B.Shinde, Abhay C., P. Chaure, Mayur C., P Waghmare, "Smart Home Automation System using Android Application", International Research Journal of Engineering and Technology, Volume 4 - Issue 4, April 2017

[6]. https://gizmodo.com/are-smart-locks-secure-or-just-dumb-511093690

[7]. https://www.myq.com/smart-lock

[8]. F. Marino, L. Maggiani, L. Nao, P. Pagano, M. Petracca Towards softwarization in the IoT: Integration and evaluation of t-res in the oneM2M architecture Presented at the Proceedings of The 3rd IEEE Conference on Network Softwarization (NetSoft), IEEE5 (2017)

[9]. Rahmani, A.-M., Thanigaivelan, N.K., Gia, T.N., Granados, J., Negash, B., Liljeberg, P., Tenhunen, H.: Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems. In: Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE, pp. 826-834. IEEE, (2015).

[10]. Kim, J.E., Boulos, G., Yackovich, J., Barth, T., Beckel, C., Mosse, D.: Seamless integration of heterogeneous devices and access control in smart homes. In: Intelligent Environments (IE), 2012 8th International Conference on, pp. 206-213. IEEE, (2012).

[11]. Razzaque, M.A., Milojevic-Jevric, M., Palade, A., Clarke, S.: Middleware for internet of things: a survey. IEEE Internet of Things Journal 3, 70-95 (2016).

[12]. Fortino, Giancarlo, Claudio Savaglio, Carlos E. Palau, Jara Puga, Maria Ganzha, Marcin Paprzycki, Miguel Montesinos, Antonio Liotta, and Miguel Llop. "Towards multi-layer interoperability of heterogeneous IoT platforms: the INTER-IoT approach." In Integration, Interoperability of IoT Systems, pp. 199-232.Springer, Cham, (2017