International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN : 2456-3307

Web of Things

Vandana C. P., Suman Thapa, Pradip Thapa
Department of Information Science Engineering, New Horizon College of Engineering, Bangalore, Karnataka, India

ABSTRACT

In the vision of the Internet of Things (IoT), an increasing number of embedded devices of all sorts (e.g., sensors, mobile phones, cameras, smart meters, smart home appliances, etc.) are now capable of communicating and sharing data with each other. With the ever-increasing capabilities of computation and communication pose new opportunities, but also new challenges. As IoT becomes an active research area, various points of view have been explored to promote the development and popularity of IoT. One trend is viewing IoT as Web of Things (WoT) where the open Web standards are supported for information sharing and device interoperation. By penetrating smart things into existing Web, the conventional web services are enriched with physical world services. This WoT enables a way of narrowing the barrier between virtual and physical worlds. In this paper, we elaborate the architecture and limitations of WoT. Finally, we point out some open challenging issues that shall be faced and tackled by the research community in the future.

Keywords : IoT, WoT, REST, RESTful API, URI, XML, JSON

I. INTRODUCTION

The Web of Things (WoT) is used to describe approaches, programming patterns and software architectural styles for real-world objects to be part of the World Wide Web. Devices are becoming smart because of advancement in computing technology. IoT is the next big possibility, and the Internet will be no longer just a network of computers but also interconnect many smart things with embedded systems. IoT will increase the size and scope of current Internet, providing new opportunities and challenges. IoT gives the everyday device an IP address and makes them interconnected on the Internet, WoT enables them to speak the same language, so as to communicate and interoperate freely on the Web.**Homes,** cities, electrical grids, retail, and manufacturing are few of many possible applications across a wide range of domains. WoT allows objects to be sensed and controlled with the use of available infrastructure. WoT provides opportunities to integrate the physical world into computer-based systems with improved efficiency, accuracy and economic benefit without any human intervention. The goal is to make interconnection and interoperability with proper security and privacy. The development of WoT is still at an initial stage, and still many issues need to be tackled to realize the full potential of WoT. The primary focus of this paper is to give the overview of WoT, its present status and the potential of WoT, and its limitations.

II. METHODS AND MATERIAL

1.2 Architecture

The focus is to incorporate smart things with current services on the Web and the making of Web applications using smart things. Architecture for the WoT should be flexible, compatible and provide with safety and security. It purposes four main layers that are used as a framework to classify different patterns and protocols involved.

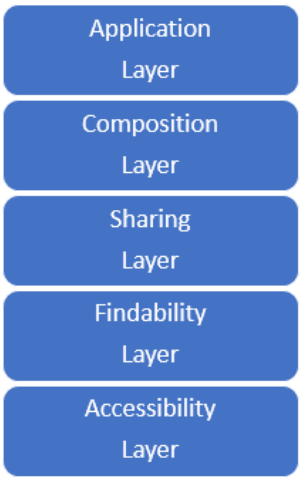


Fig: layers of Web of things architecture

a) Accessibility Layer:

In this layer, we address the access of smart things to the Web and exposure of their services via web API. Smart things are integrated into the core of the Web, such that they are treated as web pages. This is a core layer of the WoT architecture.

Access layer in the WoT follows following patterns

- Services are exposed by RESTful API
- One of the limitations of current internet is Request-Response and is not suitable for the event-driven approach of sensory devices. To overcome this HTML 5 **WebSockets** is used for real-time communication across the internet.

Websocket is bi-directional communication protocol which enables full-duplex message based communication between client and server. A dynamic web page can be created where changes occur in real time after the connection is established. Support of asynchronous changes and number of clients makes WebSocket communication a suitable protocol for IoT.

b) Findability Layer:

This layer focuses on a way to find and locate things once they are accessible on the web. It is strongly influenced by the semantic web (say SOA). Web semantic is used to describe things and services(WSDL). So we can search for things using a search engine.

c) Sharing layer:

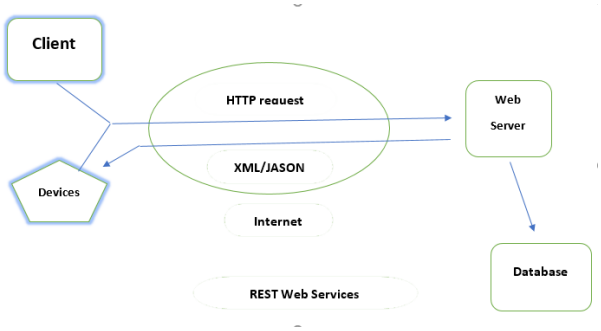
Once smart things are accessible and findable on the Web, to access those we need a common data format. sharing layer ensures that data generated by things can be shared in an efficient and secure manner. Several approaches have been proposed such as the use of social network to build a Social Web of Things.

d) Composition layer:

The last layer integrates the services and data offered for the creation of simpler applications connecting things and virtual Web services.

1.2.1 REST (Representational State Transfer)

REST is an **architecture** style for designing the networked applications and one way of providing interoperability between computer systems on the Internet. API following the REST principles does not require the client to know anything about the structure of the API. Rather, the server needs to provide whatever information the client needs to interact with the service. In REST architecture, a REST Server just provides access to resources and the REST client accesses and presents the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource like Text, JSON, and XML. JSON is the most popular format used in Web Services.



Commonly used HTTP methods in a REST based architecture are.

- **GET** − Provides read-only access to a resource.
- **PUT** − Used to create a new resource.
- **DELETE** − Used to remove a resource.
- **POST** − Used to update an existing resource or create a new resource.

- **OPTIONS** − Used to get the supported operations on a resource.

The constraints on RESTful system restrict the ways the server may process and respond to client requests. The constraints that define a RESTful system are

a) Uniform Interface

It is fundamental to the design of any REST service which defines the interface between client and server. It simplifies and decouples the architecture and central to the RESTful design. URLS are used to identified Individual resources. The resources (database) are themselves different from the representation (XML, JSON, HTML) that sent to the client. Enough information on how to process the message each message. The hypermedia (i.e., Hyperlinks and hypertext) act as the engine for state transfer.

b) Stateless Interactions

The server contains no client state. Each request includes enough context to process the message(self-descriptive). Any session state is held on the client.

c) Ccheable

Clients can cache the responses from the server. Responses can be Implicit, explicit or negotiable to not prevent clients from reusing stale or inappropriate data in response to further requests.

d) Client-Server

Clients do not have direct access to the server. The client is not concerned with the data storage thus the portability of the client code is improved while on the server side. The server is also not concerned with the client interference. Thus the server is simpler and easy to scale. Uniform interface is the link between the client and server.

e) Layered System

The client cannot assume a direct connection to the server. Software and hardware are intermediaries between client and server. Intermediary improves system scalability by providing shared caches and by enabling load balancing. It also enforces security policies.

f) Code on Demand

The server can temporarily extend the functionality of a client by the transfer of executable code. Example: Java applets, Java Scripts. This method is the only optional constraint.

1.2.2 service-oriented architecture (SOA)

SOA is an architecture approach for defining, linking, and integrating reusable business services that have clear boundaries and are self-contained with their own functionalities. These services communicate with each other. The communication can involve either simple data passing, or it could involve two or more services coordinating some activity. Services are the main focus on SOA applications.

A service has four properties according to one of many definitions of SOA

- A logical representation of business activity with a specified outcome.
- Self-Contained.
- Black box for its consumers.
- It may consist of other underlying services.

New services can be created from an existing IT infrastructure of systems in SOA as it allows the reuse of existing assets. Example any big application which uses Amazon Web Services.

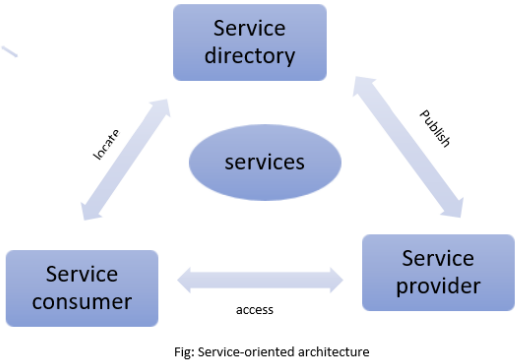


Fig: Service-oriented architecture

Service provider: Provider of services whose invocation contract and location are published
Service consumer: Consumer of services matching his or her business need found in a service directory
Service directory: Directory for publishing and listing available services for consumers

When SOA is the best fit

- Centralized business functions used by multiple entities
- Integration with partner
- The existence of old technologies that are still working

When SOA is a liability

- A homogeneous IT environment
- When true real-time performance is critical
- When things don't change
- When tight coupling is not an inconvenience

1.2.3 JSON (JavaScript Object Notation)

JSON is a way to store information in an organized, easy-to-access manner. Data can be accessed in a logical manner. JSON is a language-independent data format. JSON is widely used to support the communication between multiple APIs.

When exchanging data between a browser and a server, the data can only be text. JSON can easily be sent to and from a server and used as a data format for any programming language. We can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects. This way we can work with the data as JavaScript objects, with no complicated parsing and translations.

Example, information about a person might be written as

```
Var jason = {
        "age”: "20",
        "hometown”: "Bangalore, BLR",
        "gender”: "male"
};
```

The distribution of services over the Internet has grown in the past year’s web-based APIs has popped up allowing developers to extend their services to the ones developed by third parties. HTTP-based RESTful services have become one of the most relevant ways to implement distributed web services and JSON has emerged has the data interoperability standard that enables transparent data transfer between different implementation technologies.

2) Security threats of Web of Things(WoT)

Internet of Things (IoT) devices typically incorporates sensors, switches and logging capabilities that collect and transmit data across the Internet. Some devices may be used for monitoring, using the internet to provide real-time status updates. IoT devices need to be protected against a wider range of *active* and passive threats.

2.1 Active threats

Poorly secured smart devices are a serious threat to the security of your network. This allows attackers to use a compromised IoT device to bypass security settings and launch attacks against other network equipment as if it was from the inside.

Some of the threats includes, In a masquerade attack, the intruder pretends to be a user of a system to gain access or to gain greater privileges than they are authorized for.

In a session replay attack, a hacker steals an authorized user’s login information by taking the session ID.

In a denial of service (DoS) attack, the network layer protocols are targeted and users are denied access to their network and web resource.

2.2 Passive threats

The goal is to get detailed information about the target rather than data manipulation. However, passive attacks are often the foundation for active attacks.

Passive threats emerge due to manufacturers collecting and storing private user data. Because IoT devices are merely network sensors, they rely on manufacturer servers to do processing and analysis of data. As end users freely share critical info such as credit information to intimate personal details. With manufacturers accumulating a massive amount of data, long-term risks and threats need to be understood. Data storage by third parties is also a significant concern. These issues associated with data collection has recently come to light. Exploiting a single manufacturer’s devices anyone can access millions of people’s details.

3) Security mechanism of Web of Things(WoT)

The question of security in the connected environment remains urgent. Some of the measures to improve current IoT issues includes:

- **Secure booting:** While running IoT device for the first time, they should undergo digital verification of software to make sure other programs won't run on that device in the future;
- **Access control:** IoT applications must have mandatory controls embedded into their operating system to restrict their functions so that they can only access the resources needed for their operations;
- **Authentication:** Secure machine authentication mechanisms must be implemented to connect embedded computers to the network safely;
- **Local firewalls:** Though an IoT gadget doesn't need to filter the traffic coming to the parent network, it must have local filters to analyze the data it is going to process.
- **Data Encryption:** Secure Sockets Layer protocol or SSL certification to encrypt and protect the user's data online. Encryption of data is necessary while it is being transferred wirelessly. Sensitive data (ex: locations) need to be available to the concerned user and no one else.

III. RESULTS AND DISCUSSION

Problems/limitations of Web of Things(WoT)

Many issues still need to be overcome to explore the full potential of WoT. In this section, we evaluate some of the problems/limitations.

4.1. Heterogeneity and Scalability:

WoT is a good approach to handle the heterogeneity problem still some minor problems related to it still exist. As a large number of devices are to be integrated which are diverse regarding communication means and abilities (e.g., protocol, data rate, reliability, etc.), calculation and storing capacity, energy consumption, adaptability, flexibility, etc. At the application layer communication channel is standardized. Without the support from lower layers, WoT is not possible. Though this issue is under study, it is hard to find a solution for all the devices that may come in future. Consumers data are heterogeneous: some might use real-time information or archived data and also data quality and sampling rate also differs. Different applications use different data processing or filtering. WoT should support all these applications with various characteristics and requirements make the design of an integrated framework and the protocols a very challenging task. Because of a large distributed environment, management of WoT becomes tough, and solutions to resolve the complexity need to be found. An efficient management mechanism should be designed to overcome performance degradation. A control mechanism should distinguish the functionalities and capabilities of devices to provide user application requirements. In WoT a single device is accessed by a small number of applications and handle some requests, especially when it provides public services. A mechanism that can organize the smart things based on their capabilities and user application requirements to keep usage of the resources scalable and avoid unnecessarily denied accesses to some applications. Also, scalability can be improved by more advanced embedded web server techniques

4.2. Security and Privacy:

Government agencies, data collectors, and hackers can access the information on the devices with ease. It poses a security threat to the users and violates their privacy. Thus, gained information might get misused and cause a security breach. The interconnection of various devices makes easier for malware or worm to spread throughout the interconnected system. Many interconnected devices designed are without measures to overcome limitations of IoT. Hence, they can easily become a victim of security attack and alter the operation of any appliance/home appliance and cause damage to itself and its environment. The adoption of technology designed to prevent these concerns is essential to the development of internet of things. Also, security mechanisms should ensure the secure transfer of the transmitted data and protect against interference or misuse of the information traveling across the network.

4.3. Compatibility

Interconnectivity among the devices is very low and International standard for device compatibility is not

available. Devices from one brand don't accept the connectivity with other devices from a rival brand. Even the connectivity among the laptops and washing machines is not efficient. The products don't have to talk to each other directly as long as everyone agrees on a single platform for developing the products.

4.4. Cost

IOT provides countless benefits, but a significant amount of money is necessary to make use of these advantages. Connecting applications with the internet and with each other require resources, including money and a reliable internet connection. Also, sensors are not exactly cheap. So, buying the devices enabled with IOT (ex: smart bulb) may not be feasible just for switching off and on the bulb as regular bulb are way cheap and using the switch is pretty easy. But for IoT devices knowledge of the application is required.

4.5. Fault tolerance

Since a large number of devices will make use of the various services than the current internet. With an increase in a number of devices, IoT will be more vulnerable to attack. Fault tolerance is essential to assure the reliability of the service with the dedicated and lightweight solution. The first step towards fault tolerance may be making devices secure by default. Further, the devices should have the ability to know the current state of the network and the services with measures to fend off network failure and attacks.

IV.CONCLUSION

IoT is the next big possibility and challenge of the Internet. It does not merely concern the connectivity of smart things, but more about the interaction or interoperation between things and between things and people. This requires that all the smart things can speak the same language to communicate freely with each other. The paper illustrates the implementation and application of IoT platform based on various architecture and outlooks the future of such kind platform. It gives frameworks for the Web of Things and the growth of the IoT through Web technology standards using existing standards, simplifies application development, and enable interoperation across platforms.

V. REFERENCES

[1]. "An architecture for the Web of Things" Dave Raggett W3C/ERCIM 2004, route des Lucioles Sophia Antipolis - France dsr@w3.org

[2]. "Access Control Framework for API-Enabled Devices in Smart Buildings" Syafril Bandara1, Takeshi Yashiro2, Noboru Koshizuka1, 2, and Ken Sakamura1, 2 Interfaculty Initiative in Information Studies, The University of Tokyo, Tokyo, Japan 2YRP Ubiquitous Networking Laboratory, Tokyo, Japan.

[3]. "Securing the Internet of Things "Rodrigo Roman, Pablo Najera, and Javier Lopez University of Malaga, Spain

[4]. "The Internet of Things: Challenges & Security Issues" Gurpreet Singh Matharu Department of Information Technology Amity University Uttar Pradesh Noida, India mtech.gurpreet@gmail.com Priyanka Upadhyay Department of Information Technology Amity University Uttar Pradesh Noida, India priyanka.upadhyay0991@gmail.com Lalita Chaudhary Department of Information Technology Amity University Uttar Pradesh Noida, India lalita.chaudhary19@gmail.com

[5]. "Secure Javascript Object Notation (SecJSON) Enabling granular confidentiality and integrity of JSON documents" Tiago Santos, Carlos Serrao ISCTE - Instituto Universitario de Lisboa Ed. ISCTE, Av. das Fon;as Annadas, 1649-026, Lisbon, Portugal tfpss 1 @iscte. pt, car I os. serrao@iscte. Pt

[6]. "A Web of Things Application Architecture Integrating the Real-World into the Web" Guinard, Dominique (2011)

[7]. "The Web of Things: A Survey "Deze Zeng, Song Guo, and Zixue Cheng School of Computer Science and Engineering, The University of Aizu, Japan Email: {d8112106, sguo, z-cheng}@u-aizu.ac.jpD

[8]. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[9]. R. Khan, S.U. Khan, R. Zaheer and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges" in Proceedings of the 10th International Conference on Frontiers of

Information Technology, December 17-19, 2012, pp. 257-260.
[10]. "Survey on Restful Web Services Using Open Authorization (Oauth)" K. V. Kanmani, P. S. Smitha
[11]. "Web Services Protocol: SOAP vs REST "Vibha Kumari
[12]. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions" Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia
[13]. "Architecting a Mashable Open World Wide Web of Things" Dominique Guinard, Vlad Trifa Institute for Pervasive Computing, ETH Zurich and SAP Research CEC Zurich 8092 Zurich, Switzerland, Erik Wilde School of Information, UC Berkeley Berkeley CA 94720, USA