Medical IoT Device and Service Discovery

Ms. Supriya Salian, Dhanya, Jayalakshmi, Lavita Maria Monteiro, Meloni Jenisha Dsilva

Abstract—In an existing infrastructure, people lack unified way to discover devices and their associated services and the current approach to IoT is inconsistent with the growing network and software services associated with these devices. Similar to users of a personal computing device, users can define their preferences, install services, and manage the data that is generated and consumed by services. Hence, service request provide a basis for proper service discovery. Users looking for services will not be aware of potential services satisfying their need. This proposed framework provides a mechanism for different providers to register their devices and services, and the user can discover the devices and services of his interest at one place. We have bound the devices and services together to provide the requested service data to the user.

Keywords—Cloud storage, IoT device discovery, Service discovery, Web service.

I. INTRODUCTION

IoT is an emerging technology that enables human being to interact with web services and aims at creating a smart world. IoT is used in all the fields such as making smart city, developing smart transportation system, making smart industry and many other things. The goal of Internet of Things is to connect the people, objects and places together.

A. IoT General Application

The Internet of Things finds various applications in health care, fitness, education, entertainment, social life, energy conservation, environment monitoring, home automation, and transport systems. In all these application areas, IoT is used vastly to reduce human effort and to improve the quality of life. There are varieties of areas in which IoT applications have been developed. All of these applications are not readily available; however, some research indicates the capacity of IoT in improving the quality of life in our society.

B. IoT Medical Application

The Internet of Things is very useful for health applications. Health applications make patients to live independently in serious conditions. There are many IoT sensors that are used to measure and monitor various medical parameters in the human body. These IoT sensors continuously monitor health condition and records measured data. These applications monitor patient's health when they are not in hospital or when they are alone. Subsequently, they may suggest real time feedback to the doctor, relatives or the patient. If it finds any abnormal health condition it transmit warnings and if it is a minor problem then IoT application itself may give a solution to the patient.

An IoT application uses an Electronic Health Record (EHR) that records all medical details of person or patient and it is maintained by the health system. An EHR can be used to record body temperature, heart rate, oxygen saturation level, surges in blood sugar and blood pressure etc.

There are many wearable sensor devices and products available in the market. Wearable health devices are capable of tracking medically useful health information such as pulse, heart rate, body temperature, blood pressure, oxygen saturation and blood glucose levels etc. Some of the wearable devices are smart watches, smart textiles, and wristbands etc.

In any given context, to provide appropriate services to the user of his interest a number of devices will work together. Depending on the Iota(Internet of Things) devices the service information is gathered. The gathered information is raw and not formally organised in a set or conventional pattern in nature; therefore it is required to convert into meaningful services. Users looking for services will not be aware of potential available services satisfying their need. Further, knowing the available services in an environment, user can access those services depending on their preferences and requirements.
For example, retrieving the body temperature value can be used to illustrate the solution to the challenge faced in service discovery. The service requester seeking a service will enter the query. The keywords in the query are used to search for the appropriate service. If the keyword matches data in the database, then list of all the available services related to body temperature will be returned to the service requester where he can choose which service he wants to access.

C. IoT Device Discovery

Device discovery is a crucial step, where user can search for a device in order to accomplish a task. This allows user to discover the available devices which are registered by the different providers. It receives queries from the user and returns all available devices matching the query.

D. IoT Service Discovery

It is a mechanism to search the available services that are registered and stored in the database. This allows user to discover the services of his interest by giving the input as service type using query. After receiving the query or input from the service requester, it returns all the available services matching the query. It also includes web service with URI that are created for each service and using this sensor data can be accessed.

II. LITERATURE SURVEY

In this paper [4], the authors aimed to investigate the existing solutions to solve IoT service discovery problem, therefore they presented a full comparative analysis of the most representative (or outstanding) service discovery approaches in the literature over four perspectives: (1) the IoT service description model, (2) the mechanism of IoT service discovery, (3) the adopted architecture and (4) the context awareness.

The aim of this paper [5] is to present some of the current challenges in device and service discovery. It also discusses current models for resource discovery along with their strengths and weaknesses and how they can contribute to organization based IoT device and service discovery model.

The Internet of Things is mainly about data, different devices from different places and the connectivity between them. The goal of this paper [7] is to find a way to interact with the devices and their data, according to the customer’s requirements, in the IoT context, all this supported by the use of Web services. They have presented broker based architecture for service selection which facilitates devices to specify both functional and non-functional requirements. In this paper they have developed a device discovery and recommendation mechanism based on a proposed web service similarity metric.

In this paper [6], the authors analysed the role of context and relations with entities in IoT, then combined the characteristics of data in IoT, a common ontology based context model with the ability to handle uncertainty and worldly aspects of context is proposed and Dynamic Bayesian networks (DBN) is adopted to reason about these contexts for supporting to sense current situation. Finally, based on the context model and reasoning approach they have presented context-aware service discovery architecture.

Internet of Things is a network of physical devices that are accessible through the internet. In order to deliver appropriate services to the user and to access those services there is a need for device and service discovery mechanism. This framework [3] focuses on discovery of existing services in the user environment and an implementation of mDNS and DNS-SD for resource constrained devices. An evolution on using these devices in networks in suitable way is also presented.

One of the main problem in cloud computing is service discovery. Another issue, which has received less attention is service composition based on formal methods such as automata and formal languages. In this paper [1], behaviour composition and a complete controller synthesis technique, is adapted to provide a formal approach for resource matching and service composition.

People lack unified way to discover IoT devices and software services in an existing infrastructure, and the current approach to IoT is inconsistent with the growing network and software services associated with these devices. Similar to users of a personal computing device, users can define their preferences, install services, and manage the data that is generated and consumed by services. Hence, in this paper [2] service preferences provide a basis for proper service discovery.

III. PROBLEM STATEMENT

When the user request for a specific service as an input which includes certain keywords that describes his demand, matching services are returned to the user where he can select the service that best matches his request. Then the corresponding service data is returned to the user.
IV. PROPOSED METHODOLOGY

In the figure 1, IoT devices of different providers are registered in the IoT device store, which is present in the IoT service broker. The corresponding services provided by those devices are also registered in the IoT service store. When the user request for a specific service, matching services are returned to the user where he can select the service that best matches his request. Then the corresponding service data is returned to the user.

A broker is the backbone of the architecture and responsible for managing the registration, device and service discovery and retrieving the resources of service. It ensures that broker resources of IoT devices and services available to the user. It includes an API that enables extractions of resource description of IoT devices and services registered in the AWS cloud. The descriptions such as endpoint, device details, service details and web service for each service are stored in the database configured within the AWS cloud.

The details of the IoT device and service registration, discovery of devices and services and binding phases are described within the broker. The broker houses several functionalities such as device discovery and binding. These functionalities are developed and exposed using web services.

Initially Home page function is called to show the broker architecture from Windows Form Application. This architecture includes following modules.

A. Device Registration

The device details such as serial no, device name, provider, description and device type are entered into the windows form created, which will be stored in the database. A device id is auto-generated for each device that gets registered. It will then send that id to the broker which is then acknowledged back to the front end.

Algorithm:
Input: Serial no, Device name, Provider, Description and Device type
Output: A device id signifying the successful device registration
Steps:
1: Setup connection to the database instance in RDS.
2: Enter the device details in the device registration form.
3a: If the entered details of the device already exist in the database, then it requests the administrator to enter new device details. Go to step 2.
3b: If the device name with the same provider already exists, then it requests the administrator to enter device name with the different provider. Go to step 2.
3c: If the entered details of the device don’t exist in the database, the device gets registered successfully. Go to step 4.
4: On successful registration of the device, a device id is auto-generated for current device and it displays a popup message as “Device registered successfully with device id=____”.

B. Service Registration

The service details such as service type, URI, provider, description and device id associated with that service are entered into the windows form created, which will be stored in the database. A service id is auto-generated for each service that gets registered. It will then send that id to the broker which is then acknowledged back to the front end.

Algorithm:
Input: Device id, Service type, URI, Provider and Description
Output: A service id signifying the successful service registration
Steps:
1: Setup connection to the database instance in RDS.
2: Enter the service details in the service registration form.
3a: If the entered details of the service already exist in the database, then it requests the administrator to enter new service details. Go to step 2.
3b: If the service name with the same provider already exists, then it requests the administrator to enter service name with the different provider. Go to step 2.
3c: If the entered details of the service don’t exist in the database, the service gets registered successfully. Go to step 4.
4: On successful registration of the service, a service id is auto-generated for current service and it displays a popup message as “Service registered successfully with service id=____”.

C. Device Discovery

When the service requester enters query to find particular device, the keywords used in the query is checked with device description present in the database. If the keywords match with the device description, then it returns the device id and provider information to the service requester.

Algorithm:
Input: Device description
Output: Device id, Provider and Device data location
Steps:
1: Setup connection to the database instance in RDS.
2: Enter the device description
3: If the description matches with any of the device description present in the database, then it returns the matched device’s device id, provider details to the service requester.
4: If the description does not match, then the requested device is not present in the database.

D. Service Discovery

When the service requester enters query to find particular service, the keywords used in the query is checked with service description present in the database. If the keywords match with the service description, then it
returns the service id and provider information and URI to the service requester.

Algorithm:
Input: Service description
Output: Service id, Provider and URI
Steps:
1: Setup the connection to the database instance in RDS.
2: Enter the service description
3: If the description matches with any of the service description present in the database, then it returns the matched service’s service id, provider details and URI to the service requester.
4: If the description does not match, then the requested service is not present in the database.

Algorithm:
Input: Device id, Service URI and Timestamp (optional)
Output: Sensor data
Steps:
1: Setup the connection to the database instance in RDS.
2: Enter the request using device id, service URI and an optional timestamp
3: If the requested service with the device id is present in the database, the data corresponding to the requested device and service is retrieved from the cloud and returned to the service requester along with the service level(normal or abnormal).
4: If the service does not match with the device id provided or data is not present in the database for particular timestamp, then the request fails.

V. IMPLEMENTATION

Initially, a web service is created for all the services using Microsoft visual studio. In Amazon RDS, a DB instance is created which collects the sensor data sent from the web service. RDS consist of separate table for each service.

We have implemented this by creating a broker architecture consisting five modules i.e., device registration, service registration, device discovery, service discovery and binding. In this project, we have considered five medical devices such as thermocouple, glucometer, heart rate monitor, sphygmomanometer and pulse oximeter. Each of these devices measure body temperature, glucose level, heart rate, blood pressure and oxygen saturation level respectively. For each of these devices, a value is generated by the web service and sent to the cloud along with the timestamp.

The service requester will first find if the device and the service is available using the device and service discovery module. If present, he will request for that particular service and value generated for that service is returned to him signifying whether the value is normal or not.
VI. CONCLUSION

We have proposed unified service discovery architecture for IoT services and described abstractly the properties and requirements of an IoT Device and Service Discovery Framework. This proposed framework provides a mechanism for different providers to register their devices and services, and the user can discover the devices and services of his interest at one place. We have bound the devices and services together to provide the requested service data to the user.

VII. FUTURE WORK

In the future, requested services can be ranked as per the preferences set by the user. So that it returns only best services from the discovery results. Complex service request can be made by the user which can be resolved using service composition i.e., combing the different service data to get the final result.

References


[4] Meriem Aziez, Saber Benharzallah & Hammadi Bennoui, “A Comparative Analysis of Service Discovery Approaches for the Internet of Things”, meriemlifemaker@yahoo.com, sbharz@yahoo.fr, bennoui@gmail.com

[5] Edmore CHINDENGA, Caroline GURAJENA, Mamello THINYANE, “Towards an Adaptive Ontology Based Model for Interoperability in Internet of Things (IoT)”, echindenga@ufh.ac.za; cgurajena@ufh.ac.za, mthinyane@ufh.ac.za

[6] Qiang Wei, Zhi Jin, “Service discovery for Internet of Things: A contextawareness perspective”; weiqiang09@mails.gucas.ac.cn, zhijing@sei.pku.edu.cn