A Review on Privacy-Preserving Web Application using Data as a Service (DaaS)

Miss. Ashwini Ingole1 Miss. Ashwini Durge2 Miss. Mrunali Borikar3 Miss. Shradha Aglawe4 Mr. Rahul Shende5

1,2,3,4,5Department of Computer Science & Engineering
1,2,3,4,5RTMN University

Abstract—Now-a-days there is a growing interest in using Web services as a reliable medium for data-sharing among different data providers and users. Recently, enterprises are using service oriented architecture for data sharing in web by putting data sources behind web services instead of creating database applications. These types of web services are called as Data providing (DP) Web services. Data as a Service (DaaS) builds on service-oriented technologies to enable fast access to data resources on the web. In this project, we proposed a mediator-based approach to compose different websites of different doctors with the help of DaaS. Every doctor wants to be upgraded so he/she is always connected to new innovations and other doctors. Doctor having different website created in different language and communicate because of Window Communication Foundation (WCF). But, DaaS technology raises several new privacy concerns that traditional privacy models do not handle. In addition, DaaS composition may reveal privacy-sensitive information. Here we are trying to enhance the privacy while composing the websites. So we will describe a formal privacy model for Web Services that goes beyond traditional data-oriented models. This model extends DaaS descriptions with privacy capabilities. It deals with privacy not only at the data level (i.e., inputs and outputs) but also service level (i.e., service invocation). The algorithms like Privacy Compatibility Matching (PCM) to check the privacy compatibility, negotiation mechanism that makes it possible to dynamically reconcile the privacy compatibilities of services will be used.

Key words: DaaS Composition, WCF Binding

I. INTRODUCTION

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves patient personal information. All of this information must be managed in an efficient and cost wise fashion so that an institution’s resources may be effectively utilized. Web service will automate the management of the patient making it more efficient and error free. It aims at standardizing data, consolidating data, ensuring data integrity and reducing inconsistencies. This project will be helpful to the doctors who want to be upgradeable. In this project we will combine the information of number of disease recovery treatments and new innovations. This project will allow the doctor to keep track of his/her patients. This project will allow front office to enter details of drugs and test into database. Doctors can get visits history of patients. Also, One Doctor can access the data of other doctor through web service.

WCF allows the communication between two different web sites independent to the languages in which the websites are created. The mediator approach use to compose DaaS selects, combine and orchestrates the DaaS services (i.e. gets input from one service and uses it to call another one) to answer received queries. It also carries out all the interactions between the composed services (i.e., relays exchanged data among interconnected services in the composition). The result of the composition process is a composition plan which consists of DaaS that must be executed in a particular order depending on their access patterns (i.e. the ordering of their input and output parameters). In addition, the privacy model proposed in this project allows a service to define a privacy policy and set privacy requirements. We also propose a privacy-preserving DaaS composition approach allowing verifying the compatibility between privacy requirements and policies in DaaS compositions. This project captures all the information of patient regarding disease name, city name, zip code, weather condition areas, hospitals in those areas, doctors available in those hospitals etc. It provides the facility to doctor to access the information of other doctor from the different database. Doctor can display data of medical on his website by accessing the web service. When data from one of the web site is updated, the data on all the websites will be updated simultaneously. We are creating multiple end points for the doctor so that we can apply different types of security for enhancement of web service.

II. LITERATURE SURVEY

Traditionally, most enterprises have used data stored in a self-contained repository, for which software was specifically developed to access and present the data in a human readable form. One result of this paradigm is the bundling of both the data and the software needed to interpret it into a single package, sold as a consumer product. As the number of bundled software/data packages proliferated and required interaction among one another, another layer of interface was required. These interfaces, collectively known as Enterprise Application Integration (EAI), often tended to encourage vendor lock-in, as it is generally easy to integrate applications that are built upon the same foundation technology.

Existing system introduce only method calling to get result from the service. We cannot access data service between two applications with the help of web service. Existing system contain only one end point and that end point has to be distributed to all people. We cannot apply different permission for different client because only one end point exists. Web services introduce lot of burden when we need enhancement in web service and that will be increase burden on programmer. Web service need to modify when the two web service communicate for sharing data resource and the change in programming code will make lot of burden on administrator. One example of this type of service is the medical forums. In a medical field every doctor has to maintain their own patient database.
Doctor also needs to access the data of another doctor. In many hospitals there exist such forums through which one doctor can access the database of another doctor. But the problem with this forum is that they need to maintain another extra database for this forum.

DaaS services used by existing systems provide information very rapidly but they do not provide security to any of the confidential information. They need web server to access information of another doctor and also have to make update in the forum along with their own database.[1]

The term “Web Services” is generally used to describe a collection of protocols and standards that are used to facilitate interoperability between applications. One of the major factors for their success is the fact that they are built upon existing Internet standards such as XML and HTTP. This allows for connecting and communicating different web-sites created at different platforms. A web service is defined by Web Service and Web Method attributes. A web service only supports HTTP protocol. A web service is hosted only by IIS web service only supports security services. A web service supports XML serialization by making use of System.Xml.Serialization.WSDL.EXE tool is used for generating the same for a web service. Web service supports only SOAP or XML. Web service does not support duplex service operations. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. The W3C defines a Web service generally as a software system designed to support interoperable machine-to-machine interaction over a network. The W3C Web Services Architecture Working Group defined a Web Services Architecture, requiring a specific implementation of a “Web service.” In this: [a Web service] has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP (Simple Object Access Protocol) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

There is another technique which is more efficient than web server. It is also used for connecting different web sites. This technique is known as Windows Communication Foundation (WCF). Using WCF we can create web server without using web services. Using WCF, we can send data as asynchronous messages from one service endpoint to another. WCF supports a range of protocols, i.e., HTTP, Named Pipes, TCP, and MSMQ. Various activation mechanisms are there for WCF hosting, i.e., IIS (Internet Information Service), WAS (Windows Activation Service), Self-hosting and Windows Service.

Today, more than ever, modern enterprises are using Web services for data sharing within and across the enterprise's boundaries. We call this kind of Web service as Data Providing Web services. Specifically, data providing services are modeled as RDF Parameterized Views over mediated ontologies. Then, an RDF oriented query rewriting algorithm is used to compose services for answering received queries.

There are many types of medical forums through which one doctor can access the updated database of another doctor. Let there are two doctors, doctor d1 and doctor d2. If both the doctors are connected to this medical forum and doctor d1 has to access the updated data of doctor d2 then he can do so, but if and only if the doctor update that forum’s database otherwise doctor d1 cannot use the database of doctor d2. This means that each and every doctor connected with this forum has to update his database as well as that forums database.

III. CONCLUSION

In this work, we are implementing a Data as a service dynamic privacy model for Web services. The model with privacy at the data and operation levels provide data encryption with WCF binding. The Web Services interface provides a standard framework for performing queries on authenticated dictionaries over the Internet. Additionally, it allows clients to spend less code dealing with the serialization, canonicalization, and communication of data by delegating those tasks to already implemented standards. This, in turn, motivates smaller, simpler clients on many different possible platforms. Also propose a negotiation approach to tackle the incompatibilities between privacy policies and requirements. Although privacy cannot be carelessly negotiated as typical data, it is still possible to negotiate a part of privacy policy for specific purposes. In any case, privacy policies always reflect the usage of private data as specified or agreed upon by service providers. As a future work, we aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator.

REFERENCES
