ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Data Access Service through Web Service by Auto-Updation Mechanism

Sneha Kalbande[1], N.M.Tarbani[2]

Student, Dept. of CSE, PRMIT&R, SGBAU, Badnera, Maharashtra, India.[1]

Assistant Professor, Dept. of CSE, PRMIT&R, SGBAU, Badnera, Maharashtra, India.[2]

ABSTRACT: Web service composition is a web technology that chains information from more than one source into a single web application. This technique delivers a special kind of composition application that aims at integrating data from multiple data providers conditional on the user's request. In addition, DaaS (Data as a Service) composition may reveal privacy sensitive information. Capable of both assessing the compatibility and identifying incompatibility of service. To check the privacy compatibility of privacy policy and privacy condition. Sometime, The mediator basically discards any composition plan which is subject to privacy incompatibility from the set response.to resolve this issue Negotiation Mechanism is used. Also provide privacy with WCF having different binding Mechanism like MTOM binding, BasicHttp binding, MexBinding by sending message in SOAP format over Http.

KEYWORDS: service composition; DaaS Services; privacy Techniques; Negotiation; Compability,WCF binding, Hosting options.

I. INTRODUCTION

There is a increasing interest in using Web services as a dependable medium for data sharing between different data providers and users. Newly, enterprises are using service oriented architecture for data sharing in Web by placing data sources behind web services in its place of creating database applications. These types of web services are called as Data-providing (DP) Web services. In DP web services there is a challenge to deliver a broad spectrum of enterprises the capability to adventure the data and information that is generally stored in distributed and heterogeneous information systems. Also introduces a model of web service system that integrates distributed data sources and facilitates sharing of data through web services. The web services are built on top of existing data sources and the system enables the exchange of data through services. We also discuss service selection and query rewriting techniques for processing queries over data providing web systems.

In a Web data services environment, applications may subscribe to and consume information, provide and publish information for others to put away, or both. Applications that can serve as a consumer-subscriber and/or provider-publisher of Web data services include mobile computing, web portals, enterprise portals, online business software, social media, and social networks.[2] Web data services may provision business-to-consumer and business-to-business information-sharing requirements. Increasingly, enterprises are including web data services in their SOA implementations, as they integrate mashup-style user-driven information sharing into business intelligence, business process management, predictive analytics, content management, and other applications, according to industry analysts.
A composition involves numerous steps, which contains in: composing the high-level user goal into subtasks, finding Web services that implement the functionalities of each subtask, and [3] orchestrating the interactions between composed Web services in order to accomplish the high-level goal of the composition and to fulfill user's requirements Several techniques exist to compose Web services, mainly variants of planning such as model checking or situation calculus [10].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

II. RELATED WORK

Two factors intensify the problem of confidentiality in DaaS. First, DaaS services assemble and supply a large amount of private information about users. Second, DaaS services are capable of share this information with other entities. Moreover, the emergence of analysis tools makes it easier to analyse and synthesize huge volumes of information, hence increasing the risk of privacy damage [2]. The extensible Mark up Language (XML) [4] is a requirement offering a option for flexible storage of tree-based data. Due to their elasticity, the XML documents expanded in recent years much popularity. They are currently used for data communication, data packing, or in incident of a Web Service for function appeal calls. Since XML documents often contain confidential and reliable data, the W3C consortium has established standards that describe the XML syntax for applying cryptographic primitives to arbitrary XML data. The resulting standards have become XML Encryption [6] and XML Signature [7]. Using XML Encryption to XML data ensures its confidentiality. In parallel, XML Signature guarantees data integrity and authenticity. Both can be functional to arbitrary data in the document. While planning the Web Service every effort should be made that all the information necessary by the said web service should be delivered at one place under single umbrella that is all the functionality be defined under the functions defined in once class and less number of inner classes, unspecified inner classes, abstract classes and nesting of classes should be used. A Web service should be accessible to one and all and it should be designed in such a manner that it strictly conforms to the Web Content Accessibility Guidelines 1.0 and the standards for defined for Web Accessibility Initiative [3]. The first issue is the need for custom SOAP serializes in Java. We found that the internal object structure for STMS queries and responses could be expressed in a simpler and more efficient manner by using custom written serializes that simply plug in to the Apache SOAP package. This also contributed us a actual acceptable particle of switch over the plan of the messages, which made it easier to build a .NET client. The second issue that aided interoperability is the fact that the .NET client was written from scratch; we were free to build the object hierarchy in C# around the format of the SOAP communications, thus agreeing us to custom the constructed in SOAP serialization with the aid of code attributes.

III. PROPOSED ALGORITHM

In this segment, we present the concept of compatibility between privacy policies and requirements. Then, we express the conception of privacy subsumption and existing our cost model-based privacy matching mechanism.

Algorithm 1: *PCM*

input : $\mathtt{PR}^S$=$\{(A_j(R_i, rs_k)),\ j \leqslant |\mathtt{PR}^S|,\ i \leqslant |RS|,\ k \leqslant |P_c|,\ rs_k \in P_c,\ R_i \in RS\}$ (assertion of privacy requirements)

input : $\mathtt{PP}^{S'}$=$\{(A_{j'}(R_i, rs'_k)),\ j' \leqslant |\mathtt{PP}^{S'}|,\ i \leqslant |RS|,\ k \leqslant |P_p|,\ rs'_k \in P_p,\ R_i \in RS\}$ (assertion of privacy policy)

output: InC (The set of incompatible assertion couple);

```
1 foreach rs_k = rs'_k do
2     for i = 1, i ⩽ |RS| do
3         for j = 1, j ⩽ |PR^S| do
4             for j' = 1, j' ⩽ |PP^S'| do
5                 if (A_j'(R_i, rs'_k) ⊑ (A_j(R_i, rs_k) then
6                     A_j(R_i, rs_k) is compatible with
                      A_j'(R_i, rs'_k)
7                 else InC ← (A_j(R_i, rs_k),
                  A_j'(R_i, rs'_k))
```

Algorithm 1. Privacy Compability Matching.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

The intermediary mostly discards any composition plan which is subject to privacy incompatibility from the set answer. The main idea behind escaping empty responses is to reach a compatible through a privacy-aware negotiation mechanism. Enthusiastically resolve the privacy capabilities of services when incompatibilities ascend in a composition.

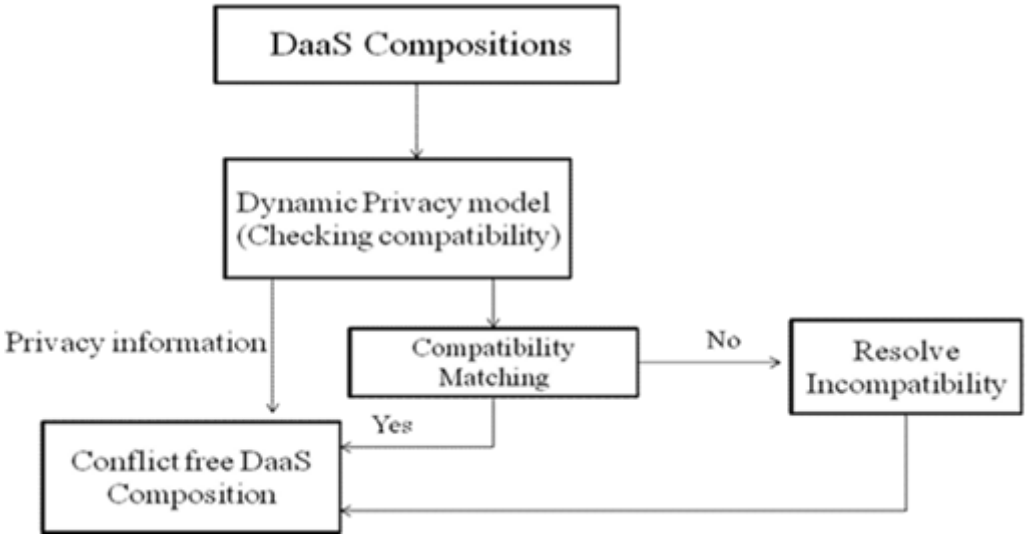

Fig.1:Compatability Checking

IV. SYSTEM ANALYSIS AND DESIGN

Two factors exacerbate the problem of privacy in DaaS. First, DaaS services collect and store a large amount of private information about users. Second, DaaS services are able to share this information with other entities. Besides, the emergence of analysis tools makes it easier to analyze and synthesize huge volumes of information, hence increasing the risk of privacy violation [2]. When such a service is executed, it accepts from a user an input data of a specified format ("typed data") and returns back to the user some information as an output. DaaS services are modeled by RDF views. Figure 3 summarizes the architecture of this project.

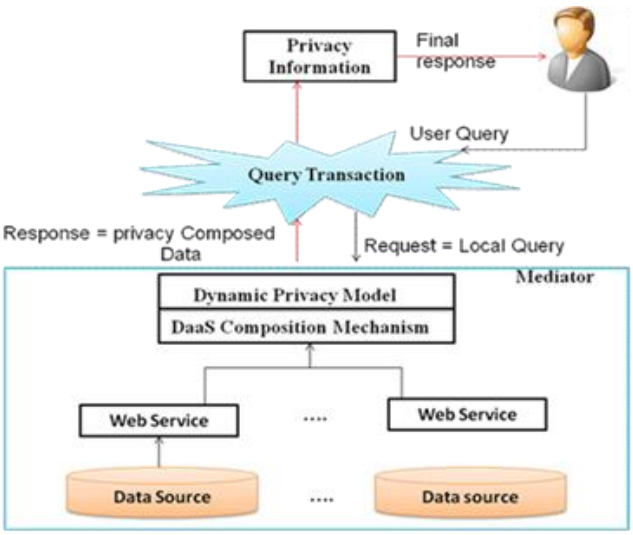

Fig 2:System Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

The Multi-Peer Query Processing component is in charge of answering the global user query. The latter has to be split local queries (i.e., sub-queries) and has to determine which peer is able to solve a local query. Each sub-query is expressed in SPARQL. Each peer handles a Mediator equipped with a Local Query Processing Engine component. The mediator exploits the defined RDF views within WSDL files to select the services that can be combined to answer the local query using an RDF a query rewriting algorithm [2]. Then, it carries out all the interactions between the composed services and generates a set of composition plans to provide the requested data.

XML Security: The Extensible Markup Language (XML) is a specification offering a possibility for flexible storage of tree-based data. Due to their flexibility, the XML documents gained in recent years much popularity. They are currently used for data transmission, data storage, or in case of a Web Service for function invocation calls.Since XML documents often contain confidential and reliable data, the W3C consortium has developed standards that describe the XML syntax for applying cryptographic primitives to arbitrary XML data. The resulting standards have become XML Encryption and XML Signature. Using XML Encryption to XML data ensures its confidentiality. In parallel, XML Signature guarantees data integrity and authenticity. Both can be applied to arbitrary data in the document.
MTOM Binding: In any of these BasicHttpBinding security scenarios you can also optimize support for large messages using MTOM an interoperable encoding format for SOAP messages that reduces message size and parsing overhead when dealing with binary data in a SOAP message. I won't get into the details of the format here, but you can configure the binding to use MTOM instead of Text encoding (the default) with the message encoding setting:

```
<basicHttpBinding>
<binding name="Soap11UserNameMTOM" messageEncoding="Mtom"/>
</basicHttpBinding>
```

MTOM encoding is orthogonal to security settings, so this can be used in conjunction with any of the other binding and behavioral settings for the service. WSHttpBinding and WSFederationHttpBinding also support MTOM encoding. For this to work, the client must also be configured for MTOM. Because MTOM is included in the WSDL for the service, the generated client configuration will automatically include the correct setting.SOAP Over HTTP;A matching threshold is set up by services to cater for partial and total privacy compatibility. The result of a composition is a set of component DaaS services which must be composed in a particular order depending on their access patterns (i.e., the ordering of their inputs and outputs parameters), to check the privacy compatibility within composite services. In a web services environment a provider supplies a set of services to consumers . The Simple Object Access Protocol (SOAP) is used for exchanging XML-based messages between the consumer and the provider over the network (usually using HTTP). In a typical web services interaction the consumer (client) sends a request SOAP message to the provider (the server). After processing the request, the server sends a response message to the client with the results. A service may include several operations and is described using WSDL (Web services Description Language) . As shown in Figure 1, the core software components supporting this scenario and allowing end-to-end communication between clients and servers are: 1) a web services framework (such as Apache Axis or Metro); 2) a web server (e.g., Apache Tomcat), in a typical interaction, a client sends a SOAP message via HTTP. When it reaches the server, the HTTP connector handles and processes the incoming HTTP request, retrieves the SOAP message and delivers it to the web service framework. The framework then processes and delivers the SOAP message to the actual service implementation (i.e., the application). In short, the framework validates each message and transforms it in an object that can be handled by the application. After this object is processed by the application, the reverse path is taken, with the return object being serialized into a SOAP response that is sent via HTTP to the client . A study is presented in with the goal of characterizing the performance of SOAP frameworks. The work uses distinct arrays to study the cost of the serialization and deserialization processes of XML parsers. Memory footprint is seen as crucial factor for deploying web services.

V. SYSTEM IMPLEMENTATION

Windows Communication Foundation (WCF) is a framework for structure service-oriented applications. Consuming WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a constantly accessible service hosted by IIS, or it can be a service hosted in an application. An endpoint can be

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as composite as a stream of binary data. A few sample states include:
While creating such applications was imaginable prior to the reality of WCF, WCF makes the progress of endpoints relaxed than ever. In immediate, WCF is designed to offer a practicable method to creating Web services and Web service clients.

- Hosting Services

To become active, a service must be hosted within a run-time environment that creates it and controls its context and lifetime. Windows Communication Foundation (WCF) services are designed to run in any Windows process that supports managed code.WCF provides a unified programming model for building service-oriented applications. This programming model residues consistent and is independent of the run-time environment in which the service is deployed. In practice, this means that the code for your services looks much the same whatever the hosting option.

- Hosting Options

WCF services can be hosted in any accomplished application. This is the most stretchy option since it involves the least infrastructure to deploy. You insert the code for the service inside the managed application code and then create and open an instance of the Service Host to make the service accessible. For more information, see How to: Host a WCF Service in a Managed Application.

- Using WCF Client Objects

A client application is a managed application that uses a WCF client to communicate with another application.
To create a client application for a WCF service requires the following steps:

- Obtain the service agreement, bindings, and address evidence for a service endpoint.
- Create a WCF client using that data.
- Call processes.
- Close the WCF client object.

.The basic steps for creating a WCF client include the following:

- Compile the service code.
- Generate the WCF client proxy.
- Instantiate the WCF client proxy.

The WCF client proxy can be generated manually, The WCF client proxy can also be generated within Visual Studio using the Add Service Reference feature. To generate the WCF client proxy using either method the service must be running. If the service is self-hosted you must run the host. If the service is hosted in IIS/WAS you do not need to do anything else.

VI. CONCLUSION AND FUTURE WORK

In this work 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. In any case, privacy policies always reflect the usage of private data as indicated or decided upon by service providers. The Web Services interface provides a standard framework for execution queries on authenticated dictionaries over the Internet. Moreover, 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. By developing prototype implementations in .NET, Also implementation of interoperability and platform-independence of proof verification.

In this work , presented literature review considering the area of web services supply chains and the need for QoS optimization in such supply chains. The gaps in various dimensions such as conceptual gap, QoS gap and the method gap are identified and pointed out. The current methods used, the QoS attributes considered and various other dimensions of the literature are classified and presented clearly for understanding the need for considering this new area of research.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Also proposed 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.

In conclusion, there is much more future work to be done in this area. More extensive performance evaluations are required in order to optimize the algorithm and compare approaches. Also, there are currently only two implementations; there exist many more XML Signature and SOAP toolkits for which clients and transforms can be written. The current implementation only validates a single proof per document; it would be significantly more efficient to be able to handle the verification of multiple proofs from the same repository with a single XML Signature validation. And also we aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator.

REFERENCES

1. M. Alrifai, D. Skoutas, and T. Risse, ‘‘Selecting Skyline Services QOS-Based Web Service Composition,’’ in Proc. 19th Int’l Conf. WWW, 2010, pp. 11-20.
2. B. Medjahed, B. Benatallah, A. Bouguettaya, A.H.H. Ngu, and A.K. Elmagarmid, ‘‘Business-to-Business Interactions: Issues and Enabling Technologies,’’ VLDB J., vol. 12, no. 1, pp. 59-85, May 2003.
3. N. Mohammed, B.C.M. Fung, K. Wang, and P.C.K. Hung,‘‘Privacy-Preserving Data Mashup,’’ in Proc. 12th Int’l Conf. EDBT, 2009, pp. 228-239.
4. M. Barhamgi, D. Benslimane, and B. Medjahed, ‘‘A Query Rewriting Approach for Web Service Composition,’’ IEEE Trans. Serv. Comput., vol. 3, no. 3, pp. 206-222, July-Sept. 2010.
5. H. Kargupta, K. Das, and K. Liu, ‘‘Multi-party, Privacy- Preserving Distributed Data Mining Using a Game Theoretic Framework,’’ in Proc. 11th Eur. Conf. Principles PKDD, 2007, pp. 523-531.
6. J. Kawamoto and M. Yoshikawa, ‘‘Security of Social Information from Query Analysis in DaaS,’’ in Proc. EDBT/ICDT Workshops,2009, pp. 148-152.
7. Y. Gil and C. Fritz, ‘‘Reasoning About the Appropriate Use of Private Data Through ComputationalWorkflows,’’ in Proc. Intell. Inf. Privacy Manage., Mar. 2010, pp. 69-74, Papers from the AAAI Spring Symposium.
8. B.C.M. Fung, T. Trojer, P.C.K. Hung, L. Xiong, K. Al-Hussaeni, and R. Dssouli, ‘‘Service oriented Architecture for High-Dimensional Private Data Mashup,’’ IEEE Trans. Serv. Comput., vol. 5, no. 3, pp. 373-386, 2012.
9. W. Ford, P. Hallam-Baker, B. Fox, B. Dillaway, B. Lamacchia, J. Epstein and J.Lapp.XML Key Management Specificatio.Http://www.w3.org/TR/2001/notexkms-20010330/. W3C Note, March 2001.
10. Anagnostopoulos, M. T. Goodrich, and R. Tamassia. Persistent authenticated Dictionaries and their applications. In Proc. Information Security Conference (ISC 2001), volume 2200 of LNCS, pages 379–393. Springer-Verlag, 2001.
11. M. Mrissa, S.-E. Tbahriti, and H.-L. Truong, ‘‘Privacy Model and Annotation for DaaS,’’ in Proc. ECOWS, G.A.P. Antonio Brogi and C. Pautasso, Eds., Dec. 2010, pp. 3-10.
12. L. Motiwalla and X.B. Li, ‘‘Value Added Privacy Services for Healthcare Data,’’ in Proc. IEEE Congr. Serv., 2010, pp. 64-71.
13. Okkyung Choi, Seokhyun Yoon ,Myeongeun Oh. Sanygoung Han, “Semantic web Search Model for information retrieval of the semantic data”, The Second HSI Conference, June, 2003, pp. 588-593.

BIOGRAPHY

Sneha Kalbande student in computer science & Engineering, college of PRMIT&R, Badnera, SGBA .Pursuing master of Engineering. Intrested in Research with Networking Technology.

Prof.N.M.Tarbani Asistant Professor in computer science & Engineering, college of PRMIT&R, Badnera, SGBAU.