A Formal Approach for Identity Management in Federated Web Services

S. Chandrasekaran¹, C.Dinesh², Kartic Ramesh³ and Al.Murugappan⁴

¹Professor  ²,³,⁴Student

¹Research and Development, R.M.D Engineering College, ²Department of Information Technology, M.N.M. Jain Engineering College, ³Department of Computer Science and Engineering, Easwari Engineering College, ⁴Department of Computer Science and Engineering, K.C.G College of Technology, Anna University, Chennai, India.

¹chandrasekaran_s@msn.com

Abstract

The objective of the proposed work is to formally specify and implement a federated web service model in which the identity management problem and security features are analyzed. The identity provider assigns equal priority to all registered services of different capability and security domains on mutual trust. The service requester may contact any other service provider inside and outside of its own capability and security domain. The service Id and the service session time are the two important factors on which the federated system security during any session are focused. The federated system with heterogeneous services is formally specified using Temporal Logic of Actions (TLA) and verified. The model is implemented using different types of clients and the performance in terms of latency is measured.

Introduction

The identity-management problem in internet computing becomes more crucial in finding out correct application in an inter-organizational setup [1]. However, the widespread adoption of identity in which users receive trusted access to more than their home network’s resources without having to sign on to each new site separately is reaching critical mass [2]. Web Service standards have formal mechanisms to specify the format of XML data structures and service interfaces [3]. However, they generally do not have methods of precisely specifying the complete behavior of a service. They instead employ informal descriptions of the service that can be imprecise, ambiguous, or incomplete; they often fail to consider unusual cases. The need for precision and completeness in a standard naturally suggests the use of formal methods. Such methods use a well-defined language with a precise semantics for writing formal specifications of a protocol's allowed behaviors. TLA is a language for writing high-level specifications of concurrent systems and provides a mathematical foundation for describing systems [4].

Unlike most specification languages, it is based on mathematics rather than programming-language constructs [5]. A high-level specification of the protocol's requirements can also be used to check the correctness of a protocol's clients. TLA+ is a particularly good language for writing standards because it is based on ordinary mathematics [6]. This makes its semantics particularly simple, since the semantics of a specification language is, by definition, a translation into mathematics of specifications written in the language. The correct unambiguous specification of the web services federation is essential in any web services security model that defines mechanisms to allow different security domains to federate by allowing and serving the registered trust of entities so as to maximize the performance of the federation.

The authentication between participating peer services can be made passive or active at the time of partial or full composition of all participating services. The requests from various web services are monitored under the control of
the identity provider for security purposes. The core idea is to predict the maximum time needed for a service gets completed and communicated by other services in a federated environment. Single sign on technique is playing an important role for earlier completion of the total business process composed of the number of web services. The work also focuses the need and importance of an alert service and the corresponding broadcast time. The introduction of a watchdog timer and a proxy service within the federation enables the uninterrupted composition of the services in a business process execution. The maximum communication time between the participating services within the limited federal life time is also achieved. The Service tokens are treated as dynamic parameters with a user-defined life time and all the participating services are to adhere to the policies of the federation. Single sign on technique can also be used within each federation and these federations have to accept the global policies for achieving inter-federation business processes.

**Formal Specification**

The formal the specification has an action by one process enabled by a predicate on the state of another process, without describing how the first process learns about the second process's state. The Identity Management problem in federated web services is formally specified in three different service levels, 1) service – level 2) Federation – level 3) Global – level written in TLA is shown in Figure 1, 2 and 3 respectively.

The variable parameters are:

- **Status** For each node n, the value of status[n] will be the string “Up” or “Down”, denoting whether or not the node is fully functional.
- **The problem explained here is based on which of the nodes are fully functional and vice versa. In order to specify the same, we introduce the variable status.**
- **cmd** For each node n, the value of cmd[n] denotes the command which could either be a request from or a response to another node. Its value is set to “?”, if the node in consideration has not yet chosen a command.
- **current** This variable will be a real number that denotes the current time.
- **reg** For each node n, the value of reg[n] denotes the registration command which belongs to the set Register.

The assertions are:

1. **Client** (the set of all clients) is a finite set that does not contain the Server.
2. “?” is a command that is not known.
3. **Current** is a real number.
4. **Δ** is a positive real number.

The initial condition **SInitial** states that

- The Server’s status is either “Up” or “Down”.
- If the Server’s status is “Up”, then its cmd is an element of Command.

The condition **CInitial** asserts that

- The Client’s status is either “Up” or “Down”.
- If the Client’s status is “Up”, then it has not chosen a command yet and is represented as “?”, i.e. there has not been any request from the server.

The specification **Fail(n)** denotes that if the node fails in the next state, its status will be “Down”, and **Tup(n)** is a tuple which is denoted by cmd[n] and status[n]. **ServSpec** and **CSpec** are specifications of the server and the client respectively that are essentially used for the server to dispatch the command and the client to receive it and process the required information.

The identity management problem is an unwanted scenario in the case of federated web services that can be reduced by having a higher authority like Identity provider whose logic of actions are specified in Figure 2.
The assertions are:

1. **System** is the set of all federations along with the global organization.
2. The specification **FedInitial** denotes the initial state of the federation. This specification states that, if a new service has been added then it has to be registered again, and hence states that \( \text{reg[n]} \) belongs to the set **Register**.
3. **GSInitial** is a specification which denotes the initial state of the GlobalServer. Since there is no request initially, its value is not known.
4. **FedSpec** and **GSSpec** denote the transfer of registration information from the federation Fed to the global entity GlobalServer. The information which is passed on during this process is a function of the requirement and is contained in the set **Register**.
   - The schema **LocalSpec** is concerned with the composition of individual services which are grouped to perform a more complicated task, by mutual trust.
   - The **FederatedSpec** schema is concerned with the federation of the services which have different capabilities and security levels into one single unit. It deals with the authorization provided by the Identity Server for the communication between two clients. This facilitates intra-federation communication.
   - The **GlobalSpec** schema is used to demonstrate the concept of a global organization which authorizes and facilitates inter-federation communication. This specification defines a global organization which intends to arbitrate between conflicting federations. In part, this specification deals with the registration process of any federation **Fed** with the global organization **GlobalServer**.
   - The specification **FedInitial** denotes the initial state of the federation. This specification states that, if a new service has been added then it has to be registered again, and hence states that \( \text{reg[n]} \) belongs to the set **Register**.
• **GSInitial** is a specification which denotes the initial state of the **GlobalServer**. Since there is no request initially, its value is not known.

• **FedSpec** and **GSSpec** denote the transfer of registration information from the federation Fed to the global entity **GlobalServer**. The information which is passed on during this process is a function of the requirement and is contained in the set **Register**.

The specifications written in TLA are focused on the application level. The system specification initiates the service interactions as per the **Module LocalSpec** followed by the **Module FederationSpec**. The specification is verified under two clients and a server for its correctness, accuracy and implemented to realize security and interoperability.

**Identity Management in the Federation Scenario**

The federation under consideration is a composition of individual services which have different capabilities and security parameters. For example, the Passport Service and Income Tax Evaluation both fall under Central authorities, while the Police service and Village Admin service is controlled by State Authorities as shown in Figure 4. The federation helps us in interconnecting services under different authorities having different capabilities. The individual services are responsible for certain local actions which they carry out.

![Figure 4 Federation Scenario](image)

**Passport Application using Web Services – Case Study**

When a customer wants to get a passport, imagine that a number of services running in the passport domain like intranet connectivity. The main services required to issue a passport to any citizen are Passport Service, Income Tax Service, Police Service and Village Administration Service. The identity of the above four services are authenticated and provided by an Identity provider which may be considered as a national service provider. Each and every main service may be considered as a federation of different associated services in its domain. The identities of the associated services are provided by that corresponding main service as a static service id or it may be a variable parameter during different sessions. For Example, the Passport main service with a high degree of security and capability is in a federation of other associated services like Entry Verification service, Document Verification service, Photo Verification and Signature Comparator.

The Police Service is composed of associated services like Dependent Verification service, Address Verification service, Criminal Records division and Profession verifier. This is in association with Village Administration service composed of Water and House Tax Evaluation service, and Income Tax service made up of Tax Return Checker, Pan Checker and Pending Tax Calculator, make up the system. The Identity Provider is used to authorize any communication between the services and arbitrate any conflicts. All the four main services have a common identity but with different no. of associated services and are in different capability domains.

In any federated web service frame work, the requesting service must be asserted by an identity either locally in its domain or the assertion may be done and certified by the identity provider that is the top level service. The inter federation needs an upper or global provider to assert the incoming requestors validity. The validity checking and the assertions based verification suffers from security problems. The registration or entry of a new service into an existing federation still faces an additional problem due to the nature of service user and also on the multiple identities for a single service. The number of associated services
within a main service and the number of such main services in a federation increases the security problem to its maximum.

The identity management problem in a federated web services needs a security model that is viewed in three levels of federal setup.

1. The identity management problem between associated services and the main service of that domain
2. The identity problem between different main services through identity provider
3. The registration and inter federation management problem and the need for global service security.

The trusted members or the associated services in any service domain have to register with the main service and get a service id for each and every session. The duration of the session may be static or dynamic, based on the statistical request arrival rate. The associated services once get a service id, and then the composition between individual services will be monitored by the main service alone. For the next session the associated services again have to re-register with the main service with the same or different identity.

The objective of this work focuses on the specification and implementation of federated services and their attributes and also establishing and reflecting trust between associated services in different domains. The work also concentrates the methodology by which the service requestors to securely find the correct Web services of a given business process and for business service providers to securely identify and expose the right Web Service to only authorized requestors. The cost of the proposed identity management is not only reducing duplication of each individual’s identity that is almost always already managed by a trusted organization.

**Security tokens and WS - Policy**

The service ids are used as Security Tokens for each and every communication between services and they can also be made as variable ones. The authentication, assertion attributes and service privileges are checked for each trusted communications between services. By using trust and federation mechanisms, along with trust policies, these identities allow federations to share and map identities automatically. There are many tokens like Mac Id, password at the beginning of each session and after that the authenticated, associated services can interact with single sign on technique.

This feature improves not only the performance in service completion but also incorporate security among federations. The purpose of this technique is to establish security tokens required to access a resource within the Web of federated domain/realms. Similarly, federated sign-out is used to clean up any cached state and security tokens that may exist within the federation [3]. The Microsoft Visual Web Developer 2005 and BEA weblogic workshop 8.1 are used to achieve interoperability among the federated web services. The summary of the services in each federation, number of services with their associated security and identity type is shown in Table 1.

**Table 1 Summary of Federated Web Services**

<table>
<thead>
<tr>
<th>Identity Provider for Federation</th>
<th>Services in each federation</th>
<th>Security level</th>
<th>Identity type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passport</td>
<td>4</td>
<td>high</td>
<td>dynamic</td>
</tr>
<tr>
<td>Police</td>
<td>4</td>
<td>high</td>
<td>dynamic</td>
</tr>
<tr>
<td>Income Tax</td>
<td>3</td>
<td>medium</td>
<td>static</td>
</tr>
<tr>
<td>Village Admin</td>
<td>2</td>
<td>low</td>
<td>static</td>
</tr>
</tbody>
</table>

In this scenario, if a client wants to acquire a passport, then his/her application will have to be processed among a number of web services which are of different capabilities and security levels. For example, the passport office needs the services of police, village administration and income tax departments which may be considered as individual federations themselves. The identity management problem arises when the passport department communicates with any other federation. To authenticate the passport office as a legitimate calling service, there should be a server which acts as a global identity provider. In order to enhance the performance, variable sessions time between federations is incorporated. The issue in this context is to empirically determine the exact duration of individual sessions over seasonal and non-seasonal peak traffic timing. As long as the session period doesn’t expire, the performance of such a scenario will be maximum but at the same time, at the end of each session, if the global id provider doesn’t respond, then the overall throughput of the federation becomes zero.
Table 2 Session, Cost, Security – Summary

<table>
<thead>
<tr>
<th>Identity Provider for Federation</th>
<th>Duration of Session</th>
<th>Capability Level</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passport</td>
<td>Very Low</td>
<td>High</td>
<td>Very High</td>
</tr>
<tr>
<td>Police</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Income Tax</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>VillageAdmin</td>
<td>Very High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Service Cost may be calculated as a product of the number of requests and the capability level with their corresponding security at that point of time. The Passport Service and the other services are implemented and tested using the BEA WEBLOGIC server and ASP.NET server respectively. The end-user can request any of the federated web services whose identities are provided by the central Identity provider. The sample code for the Identity provider which creates the Service Identity (service_id) for that session using the MAC address (macAddress) of host and the randomly generated password is shown in Figure 5.

BEA WEBLOGIC

```java
public class IDM implements com.bea.jws.WebService {
    static final long serialVersionUID = 1L;
    String command = "ipconfig -all";
    Process pid = Runtime.getRuntime().exec(command);
    BufferedReader in = new BufferedReader(new InputStreamReader(pid.getInputStream()));
    while (true) {
        String line = in.readLine();
        if (line == null) break;
        Pattern p = Pattern.compile(".*Physical Address.*: .*");
        Matcher m = p.matcher(line);
        if (m.matches()) {
            macAddress = m.group(1);
            break;
        }
    }
    serviceControl.Registration(macid, pwd, x);
}
```

Microsoft VWD

```vbnet
Public Function Registration(ByVal a As String, ByVal b As String, ByVal ID As String) As String
    Dim strConn As String
    Dim str1 As String = "UPDATE Table1 set macid=" + a + " WHERE ID =" + ID
    Dim MySqlCommand As String = str1
    Dim MyConn As New OleDbConnection(strConn)
    Dim Cmd As New OleDbCommand(MySQL, MyConn)
    MyConn.Open()
    Cmd.ExecuteScalar()
```

Figure 5 Sample Code

Conclusion

An effort is made to address the identity management problem of federated web services. The proposed web service model is formally specified using the Temporal Logic of Actions. A case study using four different services is used to test the performance of the model. The security model is verified under Java and VB.NET clients to achieve interoperability among the web services. The ideal session time during peak network traffic, is kept low in order to get latest security updates even though it increases the load on the identity server. Only those services which can pay a higher cost are given more priority thereby achieving cost benefits. An intelligent system may be implemented to strike a balance between the cost and security of the request being serviced by considering the capability of the federation and the peak hour network traffic.

References


