Dependable Privacy Management for Web Services extended to Cloud Big Data and IOT

Dr.D.Shravani
Rayalaseema University, Kurnool, A.P, India

Abstract

This research paper deals with Dependable privacy management for Web Services Security Architecture Design extended to Cloud, Big Data and IOT.

Keywords: *Security Engineering, Security Architectures, Web Services, Cloud Computing, Big Data, IOT*

I. INTRODUCTION

In this Chapter, a methodology for Dependable (Privacy Management) of Web Services Security Design is provided, with various cases design like basic Web Services Privacy, Web 2.0 Services Privacy, Financial application and Secure Stock Market design. This Security Design and implementation is done using concepts of earlier chapters like Agile Modeling and Web 2.0 Security Design.

II. DEPENDABLE PRIVACY MANAGEMENT FOR WEB SERVICES

Engineering Privacy integrates insights from diverse islands of research on electronic privacy to offer a holistic view of privacy engineering and a systematic structure for the disciplines topics. Privacy requirements are grounded on both historic and contemporary perspectives on privacy. A three-layered model of user privacy concerns relates them to system operations (data transfer, storage and processing) and they had definite effects on user behavior. Guidelines to build privacy-friendly systems needs to be developed. The two approaches are: "privacy-by-policy" and "privacy-by-architecture". The former focuses on the implementation of the notice and choice principles of fair information practices, while the later approach minimizes the collection of identifiable personal data and emphasized anonymization and client and client-side data storage and processing. Both approaches have technical overlaps as well as boundaries, providing economic feasibility. Engineers and computer scientists of privacy research domain should work on designing privacy-friendly systems [Sarah Spiekermann].

Table 5.1 provides three layer privacy responsibility framework and engineering issues.

Table 5.2 provides three-layer privacy responsibility framework and associated user privacy concerns

Table 5.1. Three Layer Privacy Responsibility Framework and Engineering Issues

Privacy Spheres	Where data is stored	Engineer's responsibility	Engineering Issues
User Sphere	Users desktop Personal, computers, Laptop, Mobile phones, RFID chips	Give users control over access to themselves (in terms of access to data and attention)	What data is transferred from the client to a data recipient Is the user explicitly involved in the transfer Is the user aware of remote/or local application storing data on his system Is data storage is transient or persistence
Joint Sphere	Web Service providers servers and databases	Give users some control over access to themselves (in terms of access to data and attention) Minimize users future privacy risks	Is the user fully aware of how his data is used and how he controls this
Recipient Sphere	Any data recipients, servers and databases of network providers, service providers or other parties with whom data recipient shares data	Minimize users future privacy risks	What data is being shared by the data recipient with other parties Can the user expect or anticipate a transfer of his data by the recipient Is personal data adequately secured Is data storage transient or persistence Can the processing of personal data be foreseen by the user Are there secondary uses of data that may not be foreseen by the user Is there a way to minimize processing (e.g. by delegating some pre-processing to User Sphere)

Table 5.2 Three-Layer Privacy Responsibility Gramework And Associated User Privacy Concerns

Sphere of Influence	User Privacy Concerns
User Sphere	Unauthorized collection Unauthorized execution

	Exposure Unwanted inflow of data
Joint Sphere	Exposure Reduced Judgment Improper access Unauthorized secondary use
Recipient sphere	Internal unauthorized use External unauthorized use Improper access Errors Reduced Judgment Combining data

Privacy is about protecting information about individuals. Furthermore, an individual can specify to a Web service provider the information that can be released about him or her. Privacy has been discussed a great deal in the past, especially when it relates to protecting medical information about patients. Social scientists as well as technologists have been working on privacy issues.

Privacy has received enormous attention during recent years. This is mainly because of the advent of the Web, the semantic Web, counterterrorism, and national security. For example, in order to extract information about various individuals and perhaps prevent and/or detect potential terrorist attacks, data mining tools are being examined. We have heard much about national security versus privacy in the media. This is mainly due to the fact that people are now realizing that to handle terrorism, the government may need to collect data about individuals and mine it to extract information. Data may be in relational databases or it may be text, video, and images. This is causing a major concern with various civil liberties unions. Therefore, technologists, policymakers, social scientists, and lawyers are working together to provide solutions to handle privacy violations.

III.IMPLEMENTATIONS AND VALIDATIONS

A. Case 1. Privacy Management for Web Services

Privacy has become a major concern because the news contains numerous stories of personal information misuse. One of the misuses of personal information is identity theft, but that's by no means the largest misuse. Many users also feel that gathering personal information for marketing purposes without permissions and full disclosure of the requestor will use the information is also a major misuse of personal information. People don't want to suffer through a barrage of unwanted sales calls as witnesses by the proliferation of "No Call" lists both locally and nationally. In fact, many people are taking positive steps to take back their personal information or at least block further attempts to acquire new information. The proliferation of spy ware blockers shows that users are becoming aware of covert attempts by some Web sites (including pop-up advertisements) to steal data from their systems. Personal information covers a range of topics today. Most developers recognize that name, address, telephone number, and other personally identifying information is private. However, users don't want developers to know a lot of information that some developers see as belonging to the public domain.

The Table 5.3 displays the code for adding a compact policy to a web page. The <meta> tag at the beginning of the code is the essential addition to our application. The http-equiv attribute tells the server what kind of response header to add. Some servers don't honor this attribute, so this solution might not work completely in all cases. The content attribute tells the client where to locate the privacy policy for our Web Site – it works much the same as the <link> tag. Finally the CP attribute defines the compact policy for our server. Most tools, such as the IBM P3P Policy Editor, which tells us what these codes means and generate a text file containing them for us.

The test code consists of two functions attached to buttons on the example form. The first creates a cookie and attaches it to the document. The second retrieves the cookie stored in the document and displays the results on screen. Neither function is that exciting, but this is enough code to create an error with Internet Explorer 6 if the compact policy isn't accepted. We must have a compact policy in place and Internet Explorer 6 must accept it if we want users to use the high privacy settings. However, even if Internet Explorer 6 decides that it won't accept the compact policy, having a privacy policy in place and set up using the information provided let's user reply on the medium high privacy setting. Although the medium high setting isn't quite as comfortable as the high setting, it's much better than the low setting our web site would require if it didn't have a privacy policy.

Table 5.3 Addding a Compact Policy to a Web Page

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.0 Transitional//EN">
<html>
<head>
<meta http-equiv='P3P' content=
'policyref=http://www.mipgs.ac.in/sravani.htm",
CP="NOI DSP COR NID CURa OUR NOR NAV
INT TST"'
 name=vs_targetSchema
content="http://schemas.microsoft.com/intellisense/i
e5">
<title>Privacy Demonstration</title>
```

```
<script>
function SetCookie()
{
   var  UserCookie; // Stores the user name.

   // Create the username cookie.
   UserCookie = "UserName=" +
escape(InputVal.value);

   // Add the cookie to the document.
   document.cookie = UserCookie;

   // Tell the user the cookie was saved.
   alert("The cookies were saved.");
}
function ReadCookie()
{
   var  ACookie; // Holds the document cookie.
   var  Parsed; // Holds the split cookies.
   var  Name;    // The user name.

   // Get the cookie.
   ACookie = unescape(document.cookie);
   // Split the cookie elements.
   Parsed = ACookie.split("=");
   // Get the user name.
   Name = Parsed[1];
   // Display the name.
   alert("Your name is: " + Name);
}
</script>
</head>
<body>
 <h1 align="center">Privacy Demonstration</h1>
 <label id="Input"><span style="TEXT-
DECORATION: underline">U</span>ser Name:<
/label>
 <input id="InputVal" type=text
value="Dr.D.Sravan Kumar" name="InputVal"
accesskey="U" title="Type your name."
autocomplete=off/><p/>
 <input id="SaveCookie" type=button
accesskey="S" value="Save Cookie"
onclick="SetCookie()" name="SaveCookie"
title="Click or press Alt+S to save the cookie."/>
   <input id="ReadCookie" type=button
accesskey="R" value="Read Cookie"
onclick="ReadCookie()" name="ReadCookie"
title="Click or press Alt+R to read the cookie."/>
</body> </html>
```

Figure 5.1 shows the Privacy Demonstration with user name displayed, with options for save cookie and read cookie. Figure 5.2 shows the save cookie option and Figure 5.3 shows the read cookie option.

FIGURE 5.1. PRIVACY DEMONSTRATION

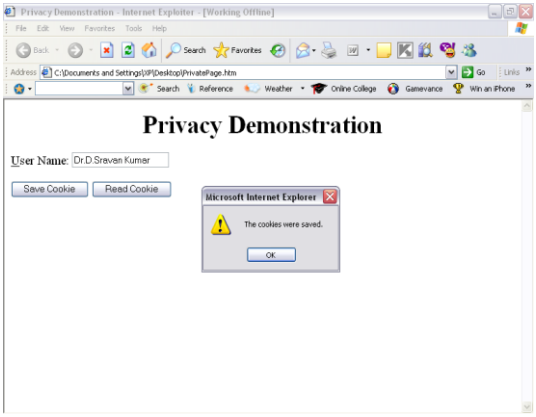

Figure 5.2. Save Cookie Option

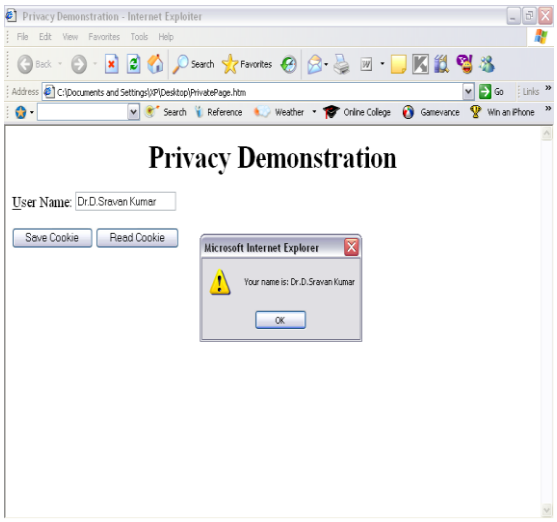

Figure 5.3 Read Cookie option

B. Case 2. Validation of the Privacy of Web 2.0 Services Application

MDA with executable UML offers an approach that embodies all the key ingredients of the process for developing dependable systems, by offering: A uniform strategy for preserving investment in existing models built using unsupported tools, by automatically migrating them to profiled UML models for subsequent maintenance and development using state of the art UML tools; A clean separation of application behavior from the platform specific implementation using technologies such as Integrated Modular Avionics (IMA), allowing the full potential of IMA to be realized in a consistent and dependable way; A semantically well defined formalism that can be used a basis for modular certification of safety related systems; The ability to generate not only the components of the target system, but components of development tool chain, providing scope for model translation and offering "executable specifications" that can be tested early and mapped reliably onto the target, leading to greater levels of dependency.

MDA is a new approach for most organizations, and therefore carries additional training and learning curve costs and also currently the availability of production quality code generators is currently limited. MDA requires developers to work at a more abstract level than code although experience shows that most do not have any difficulty making the adjustment, there will be some who find this change of emphasis difficult to achieve. Building upon the initial success of MDA deployment so far, work is now proceeding on the enhancement of Ada code mapping rules to cover the entire xUML formalism. Work is also underway to develop a generic "adapter/router" component to provide a standard component to provide a standard way to interface re-engineered xUML components with pre-existing components. These techniques are now being applied to another avionics system in the same organization, in response to the customers need for a faster and cheaper upgrade capability. While we consider systematically all actions within a use case and analyzed how they could be subverted, it produces all (or most) of the threats to a given application. While all this could be done in textual version of the use case, the use of UML activity diagrams produces a clear and more intuitive way to analyze these attacks. From the threats we derive necessary policies to stop or mitigate them. The figures 5.4, 5.5, 5.6 provide the validated class diagram, sequence diagram and detailed sequence diagrams respectively of the previous implemented privacy web services application.

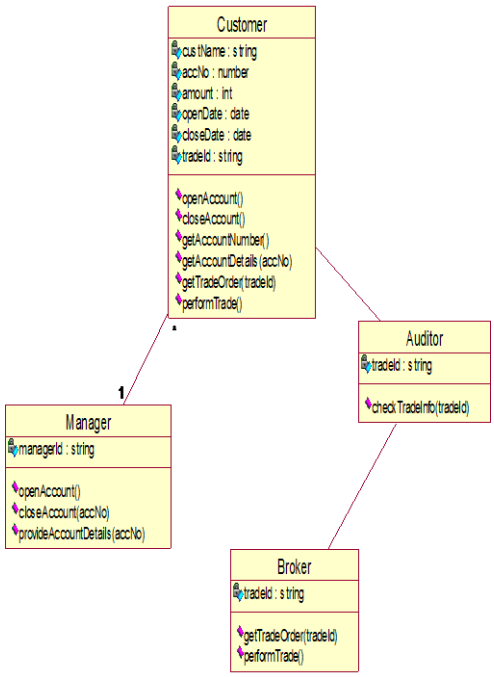

Figure 5.4. Class Diagram of the Web 2.0 Services Privacy

application

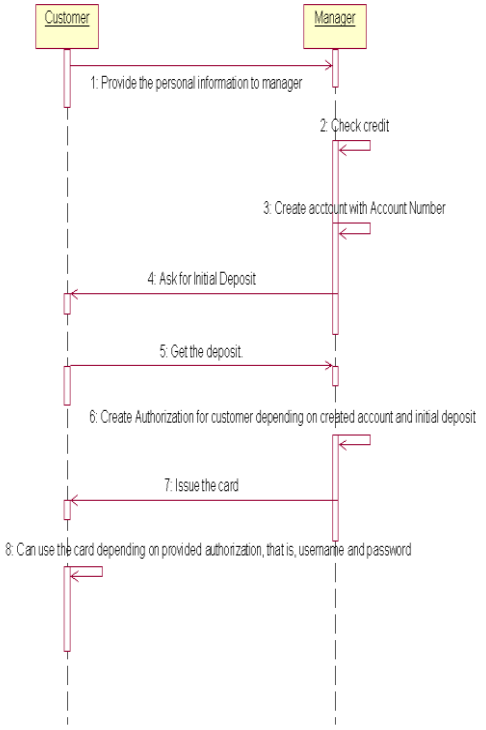

Figure 5.5. Sequence Diagram of the Web 2.0 Services Privacy Application

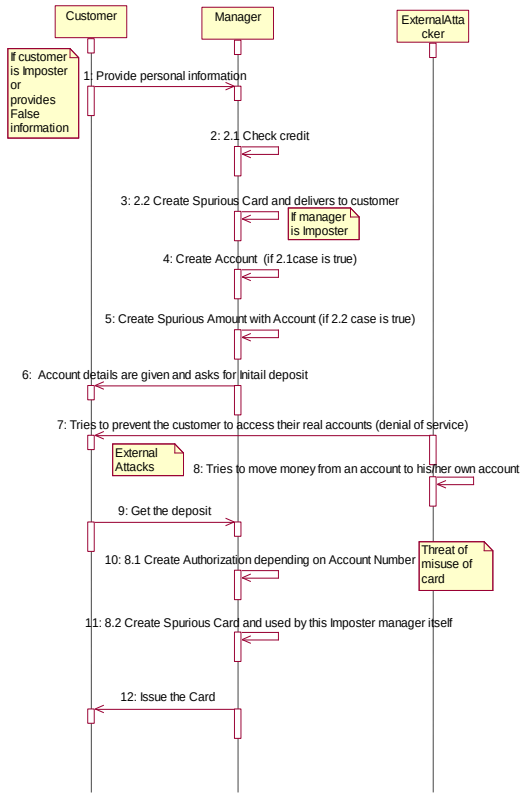

Figure 5.6. Detailed Sequence Diagram of the of the Web 2.0 Services Privacy Application

C. Case 3. Validation of the Privacy Web Services Application with an extension to Web Services Cloud

Privacy is a well recognized sticking point in the Web Services network. In this case study implementation, we explore privacy protection brings about many new security challenges. Web Services extended Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. Here data (i.e., message/file) is transferred between the sender and the receiver in an extensively secure manner. Hence the communication between the sender and receiver is guaranteed by the Third Party Auditor (TPA).The software is designed in such a way that the user can easily interact with the screen because they are GUI and screen has several buttons with captions indicating the functionality like Sender details, message typed, searching for a file, keys and signatures generated, shows the encrypted data, verification of data, system name details, Receiver details. Business layer of this application are to be developed in such a way they must be easily maintainable and extensible. Software developed will able to do ant type of data transfer between sender and receiver in an authenticated, privacy of data contents.

The Figures 5.7, 5.8, 5.9 provides the class diagram, sequence diagram and execution screen shot respectively of the privacy web services application implemented.

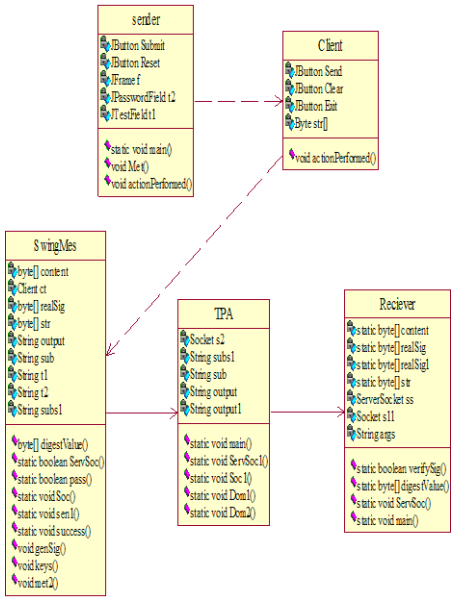

Figure 5.7. Class Diagram of the Privacy Web Services application with an extension to Web Services Cloud

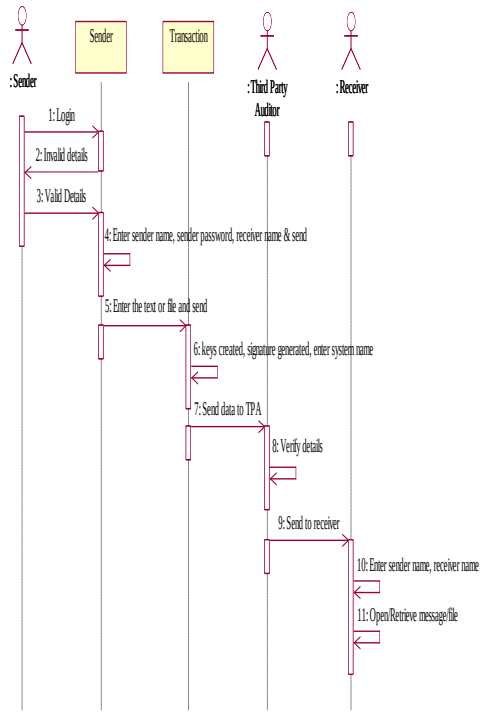

Figure 5.8. Sequence Diagram of Privacy Web Services Application with an Extension to Web Services Cloud

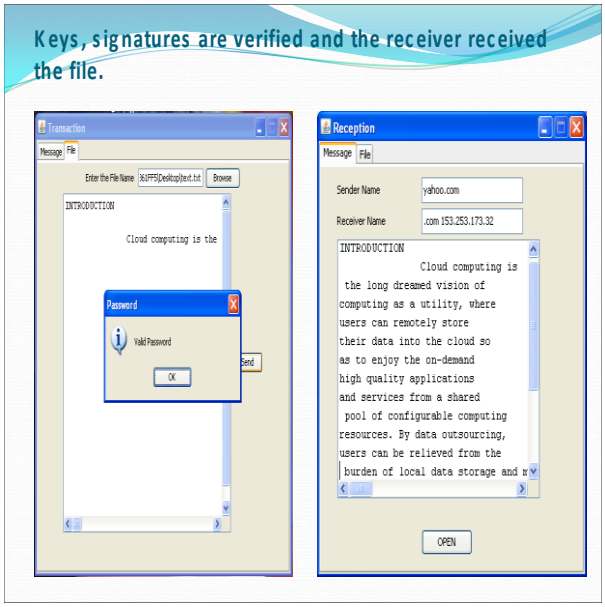

Figure 5.9. Execution Screen Shot of Privacy Web Services Application with an Extension to Web Services Cloud

D. Case 4. Secure Web Services Design for Stock Market with Business Processes Role Based Access Control Privacy

Secure Stock Exchange Web Services design in J2EE is all about Secure Stock Exchange System using Web Services containing the following: :Stock Markets & Investments, Stock Options, Related Information. A stock exchange is simply a market that is designed for the sale and purchase of securities of corporations and municipalities. This means that a stock exchange sells and buys stocks, shares, and other such securities. In addition, the stock exchange sometimes buys and sells certificates representing commodities of trade. Secure Stock Exchange system is simply a system that is designed for the sale and purchase of securities of corporations and municipalities. A stock exchange sells and buys stocks, shares, and other such securities. In addition, the stock exchange sometimes buys and sells certificates representing commodities of trade. At first, stock exchanges were completely open. Anyone who wished to buy or sell could do so at a stock exchange. However, to make stock exchange more effective, membership became limited to those in clubs and other associations. Today, professionals who have a seat at the exchange are the people who trade at the exchange if a broker approaches a post and sees that the price of the stock is what they are authorized to pay, the broker can complete the transaction themselves. As soon as a transaction occurs, the broker makes a memorandum and reports it to the brokerage office by telephone instantly. The buying and selling of stocks at the exchange is done on an area which is called the floor. All over the floor are positions which are called posts. Each post has the names of the stocks traded at that specific post. If a broker wants to buy shares of a specific company they will go to the section of the post that has that stock. If the broker sees at the price of the stock is not quite what the broker is authorized to pay, a professional called the specialist may receive an order. The specialist will often act as a go-between between the seller and buyer. What the specialist does is to enter the information from the broker into a book. If the stock reaches the required price, the specialist will sell or buy the stock according to the orders given to them by the broker. The transaction is then reported to the investor. If a broker approaches a post and sees that the price of the stock is what they are authorized to pay, the broker can complete the transaction themselves. As soon as a transaction occurs, the broker makes a memorandum and reports it to the brokerage office by telephone instantly. At the post, an exchange employee jots down on a special card the details of the transaction including the stock symbol, the number of shares, and the price of the stocks. The employee then puts the card into an optical reader. The reader puts this information into a computer and transmits the information of the buy or sell of the stock to the market. This means that information about the transaction is added to the stock market and the transaction is counted on the many stock market tickers and information display devices that investors rely on all over the world.

Today, markets are instantly linked by the Internet, allowing for faster exchange.

The following are the modules implemented in this Secure Stock Exchange System using Web services: Securities: The securities view provides a list of all available securities. From here you can open charts and news headers specific to each security, and drag a security to populate other views. Watch List: The watchlist view allows you to keep track of the price trend of the securities. The watchlist wizard allows you to define the name of the watchlist and the columns to display. To add the securities to a watchlist drag a security from the securities view and drop it to the watchlist. The security will be added at the end of list. Charts: Chart views provide a graphical representation of the historical prices for a given security. The view's pull-down menu to add indicators and drawing objects to the chart that helps you to perform the technical analysis of the price trends. Indicators can be added over existing indicators, on a new tab in the same row with other indicators or alone in a row. Patterns: From the Watchlist or Securities views you can search each security for a pattern in the price history. The patterns search view provides a list of all performed search results**.** Accounts: The accounts view provides a list of trading accounts and keep track of your owned assets. All accounts are transaction-based. Portfolio: The portfolio view provides a list of all open positions for the available trading accounts. Trading: The trading feature allows you to submit orders to a broker using: Account, Security, Provider, and Order. And keep track of the submitted orders and their status using the orders view. The Figure 5.10 which provides the architecture of this application.

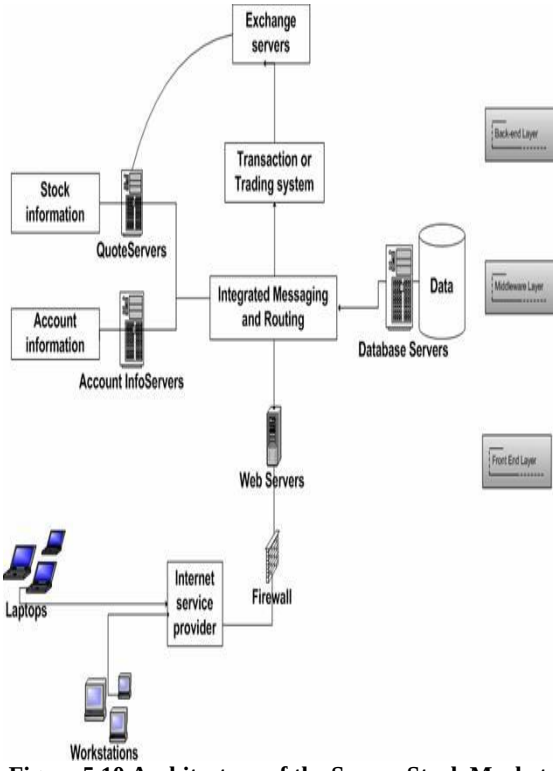

Figure 5.10 Architecture of the Secure Stock Market Application the Figures 5.11 to 5.15 Provide the Sequence Diagrams and Execution Screen shots of this Application.

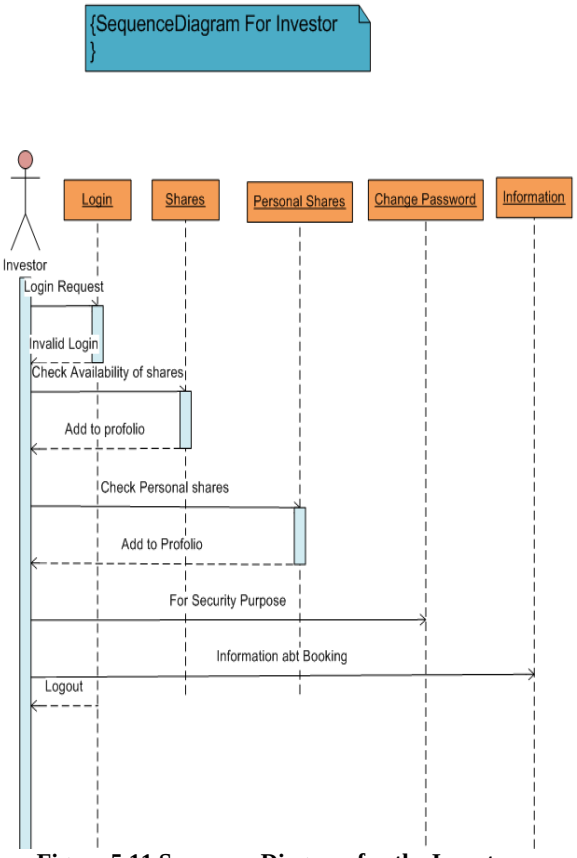

Figure 5.11 Sequence Diagram for the Investor

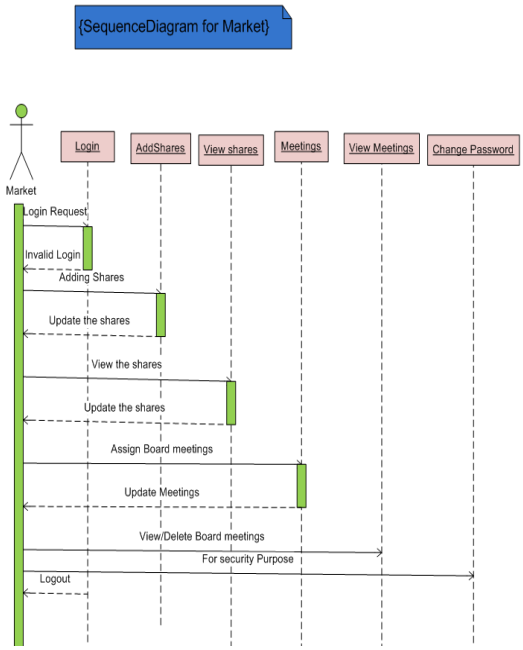

Figure 5.12 Sequence Diagram for the Market

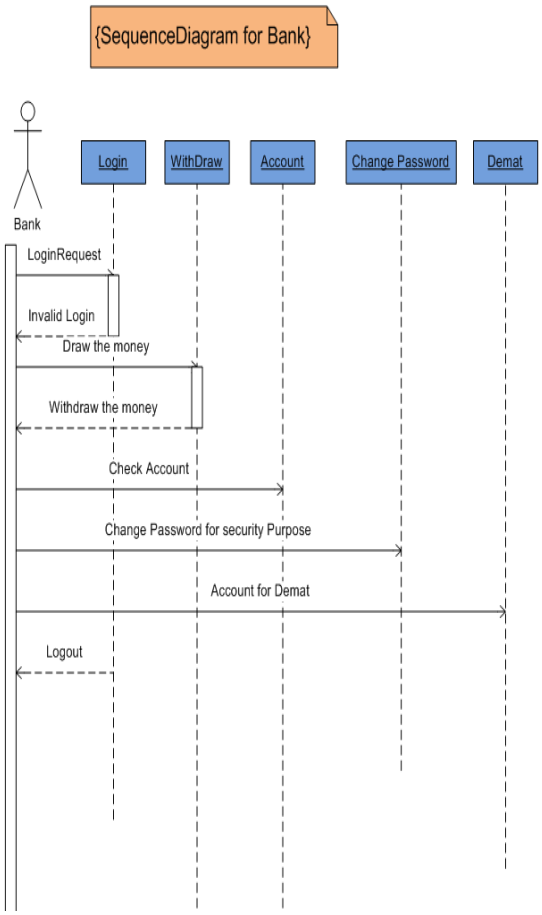

Figure 5.13 Sequence Diagram for the Bank

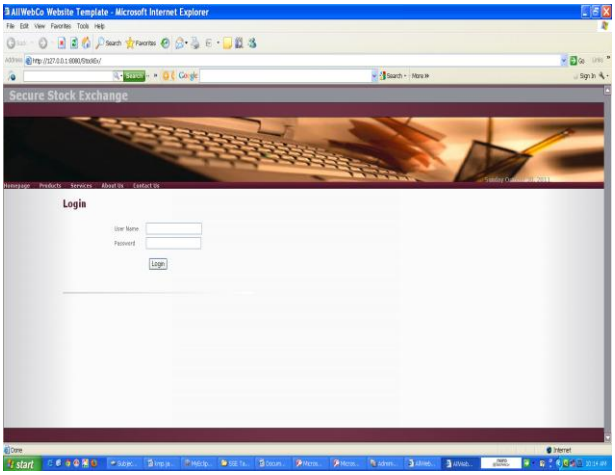

Figure 5.14 Login Form of the Bank

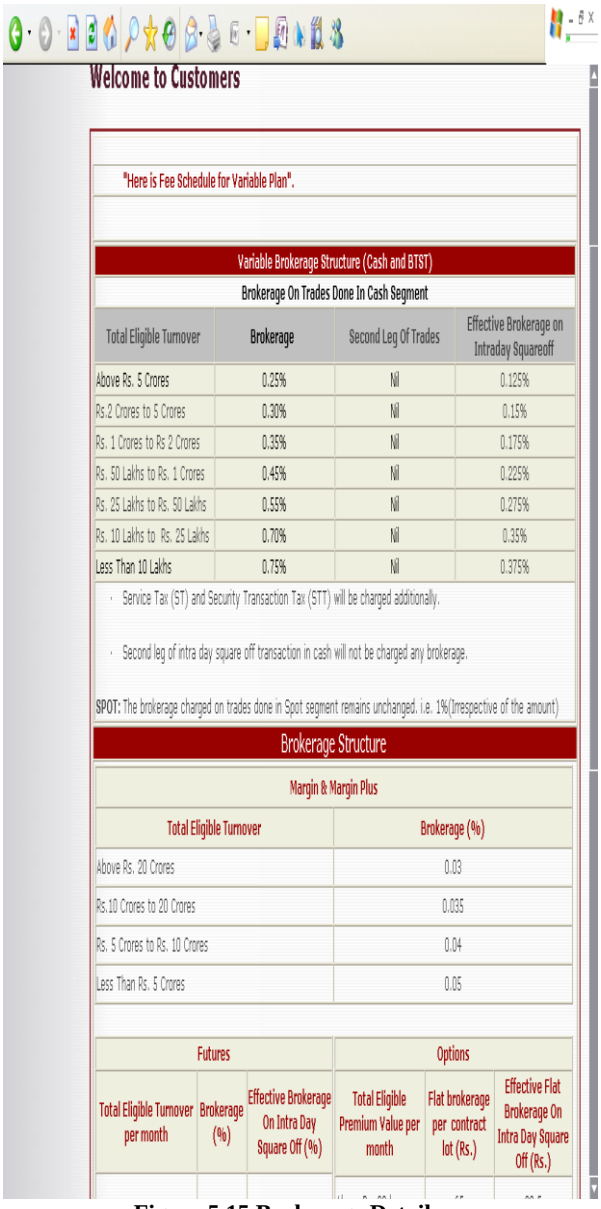

Figure 5.15 Brokerage Details

BPEL RBAC Security for Workflow and Business Processing, focuses on an important

component that makes it possible to build and manage complex applications. In a Web-based environment, business processes or workflows can be built by combining Web services through the use of a process specification language. Such languages basically allow one to specify which tasks have to be executed and the order in which those tasks to be executed. One such language is WS-BPEL, which provides syntax for specifying business processes based on Web Services. A problem of particular relevance for security is the development of access control techniques supporting the specification and enforcements stating which users can execute which tasks within a workflow, while also enforcing constrains such as separation of duty on the execution of those tasks.

BPEL Editor using Google Web Toolkit (GWT) GWT is an open source set of tools that allows web developers to create and maintain complex JavaScript front-end applications in Java. When the application is deployed, the GWT cross-compiler translates the Java application to JavaScript, CSS and HTML. GWT does not revolve only around user interface programming; it is a general set of tools for building any sort of high-performance client-side JavaScript functionality. BPEL is an orchestration language built on the foundation of the XML and Web services which use a XML based language that supports the Web Services technology Stack. If any application wants to work on multiple Web Services, for example if there is an application which involves three business processes using three Web Services of hotel reservations, cab reservations Airline reservations, then BPEL is the solution.

Business Process Execution Language (BPEL) is a XML-based language used to define Enterprise business processes within Web services. The key objective of BPEL is standardize format of business process flow definition so companies can work together seamlessly using Web service. Processes written in BPEL can orchestrate interactions between Web Services using XML documents in a standardized manner. BPEL is used to model the Behavior of both executable and abstract processes. Executable processes model actual behavior in business transactions. Abstract processes interact without revealing their internal behavior. In the existing system, in order to access two or more interdependent web-services simultaneously, the client has to use one web-service, close the connection and then use the second service. In the proposed system we will created front end using GWT. For orchestration of several web services at a time we use BPEL. This would overcome the above drawbacks and also reduces overall cost and maintenance. First we created an application with GWT as front end. The customer should register himself in order to proceed to access service. The user needs to input all the required particular details during the registration process. The web service will perform validation checks on customer input and length constraints. Upon successful login, the customer will be registered officially to the web service and he can login using his username and password. Second we developed graphical user interface, connecting the several web services using BPEL and making the connection with the database for accessing the web services. Third we showed how the user can access service. Like retrieving information of availability of tickets and can book a ticket and if another services user needs may go for it and all the details will be stored in the database [Rafuel Accorsi].

IV. CONCLUSION

In this paper we discussed about privacy management for web services security design for dependability with various case studies like web 2.0 services using agile modeling.

REFERENCES

[1] Martin Naedele [2003], “Standards for XML and Web Services Security”, IEEE April 2003, pp. 6 – 14.

[2] Mark Harman, Afshin Mansouri [2010], “Search based Software Engineering Introduction to the special issue of the IEEE Transactions on Software Engineering”, November December 2010, pp. 737 – 741

[3] Massimo Barloletti, Pierpaolo Degano, Gian Luigi Ferrari, Roberto Zunino [2008] , “Semantics-Based Design for Secure Web Services”, IEEE Transactions on Software Engineering, Vol 34, No.1, January 2008, pp. 33 – 49.

[4] Matt Bishop [2003], “Computer Security Art and Science”, Pearson education, pp. 56 – 97.

[5] Meiko Jenson, Nils Gruschke, Ralph Herkenhoner [2009], “A Survey on attack of Web Services Classification and Counter measures”, Journal Computer Science Research and Development Vol 24 No 4, November 2009, pp. 185 – 197

[6] Michael S Kirkpatrick, Elisa Betrino [2010], “Enforcing Spatial Constraints for Mobile RBAC Systems”, ACM 2010 SACMAT10, June 9-11, 2010, Pittsburg, USA, 99. 1 –6.

[7] Michael Juntao Yuan, 2004, “Enterprise J2ME Developing Mobile Java Applications”, Pearson Education Inc., ISBN 81-297-0694-6, pp. 145 – 196.

[8] Michele Barletta, Alberto Calvi, Silvio Ranise, Luca Vigano, Luca Zanetti [2011], “Workflow and access control reloaded. A declarative specification framework for the automated analysis of web services” Scalable Computing Volume 12 Number 1 pp. 1-20

[9] Mokbel M.S., Jiajin. L, [2005 – 2008] “Integrated Security Architecture for Web Services and this challenging”, Journal of Theoretical and Applied Information Technology JATIT, pp 518 – 525.

[10] Mouratidis and Giorgini [2007], “Security and Software Engineering: Advances and Future Vision.” Idea Group Publishing Inc., pp. 1 – 17.

[11] Mordinyi, R.Kuhn, E Schatten [2010], “Towards an Architectural Framework for Agile Software Development” in IEEE 17th International Conference and Workshop on Engineering of Computer Based Systems (ECBS), pp. 276-280.

[12] Munindar P.Singh, Michael N.Huhns [2005], “Service Oriented Computing, Semantics, Processes, Agents”, John Wiley & Sons, Ltd, pp. 45 – 97..

[13] Murat Gunestas, Dumminda Wijesekera [2009], “Forensics over Web Services: The FWS”, IGI Global, Information

Science Reference, DOI:10.4018/978-1-60566-950-2.ch005, 83 – 98.

[14] Nakamura Y, Tatsubori M, Imamura T, Ono K, SCC [2005] ,"Model driven Security based on a Web Services Security Architecture", proceedings of the 2005 IEEE International Conferences on Services Computing, pp. 1 – 15.

[15] Nico Brehm, JorgeMarx Gomez [2009] , "Secure Service Rating in Federated Software Systems based on SOA", IGI Global, Information Science Reference, DOI:10.4018/978-1-60566-950-2.ch004, pp. 83 - 98. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.