DESIGNING DEPENDABLE AGILE LAYERED WEB SERVICES SECURITY ARCHITECTURE SOLUTIONS

M.UPENDRA KUMAR*

Research Scholar CSE JNTU Hyderabad A.P. India
uppi_shravani@rediffmail.com
http://sites.google.com/site/upendramgitcse

Dr.D.SRAVAN KUMAR

Principal and Professor CSE KITE WCPES Hyderabad A.P. India
dasojusravan@yahoo.co.in

Dr. B.PADMAJA RANI

Professor CSE JNTU CEH Hyderabad A.P. India
padmaja_jntuh@yahoo.co.in

K.VENKATESWARA RAO

Associate Professor CSE JNTU CEH Hyderabad A.P. India
kvenkateawarrao_jntuh@rediffmail.com

Abstract

Service Orientation Engineering (SOE) (using Web Services) and Agile modeling software development presents promising solutions for contemporary software development projects to deal effectively with challenges in increasingly turbulent business environments typified by unpredictable markets, changing customer requirements, pressures of even shorter time to deliver, and rapidly advancing information technologies. Web Services Security Architectures have three layers, as provided by NIST standard: Web Service Layer, Web Services Framework Layer (.NET or J2EE), and Web Server Layer. In services oriented web services architecture, business processes are executed as a composition of services, which can suffer from vulnerabilities pertaining to secure data access and protecting code of Web Services. The goal of the Web services security architecture is to summary out the details of message-level security from the mainstream business logic, with a focus on Web Service contract design and versioning for SOA. Service oriented web services architectures impose additional analysis complexity as they provide much flexibility and frequent changes with in orchestrated processes and services. In this paper, we discuss about developing dependable solutions for Web Services Security Architectures using Agile Layered security architectures in terms of Privacy requirements. All this research is motivated by Secure Service Oriented Analysis and Design research domain. We initially validate this by a BPEL Editor using GWT for RBAC and Privacy. Finally a real world case study is implemented using J2EE, for validating our approach. Secure Stock Exchange System using Web Services is to automate the stock exchange works, and can help user make the decisions when it comes to investment.

***Keywords*:** Security Architectures; Agile Modeling; Web Services; BPEL RBAC; Service Oriented Analysis and Design, Privacy for Business Processes

1. Introduction to Agile Modeled Layered Security Architectures

Software Engineering covers the definition of processes, techniques and models suitable for its environment to guarantee quality of results. An important design artifact in any software development project is the Software Architecture. Software Architecture's important part is the set of architectural design rules. A primary goal of the architecture is to capture the architecture design decisions. An important part of these design decisions consists of architectural design rules. In an MDA (Model-Driven Architecture) context, the design of the system

architecture is captured in the models of the system. MDA is known to be layered approach for modeling the architectural design rules and uses design patterns to improve the quality of software system. And to include the security to the software system, security patterns are introduced that offer security at the architectural level. More over, agile software development methods are used to build secure systems. There are different methods defined in agile development as extreme programming (XP), scrum, feature driven development (FDD), test driven development (TDD), etc. Agile processing is includes the phases as agile analysis, agile design and agile testing. These phases are defined in layers of MDA to provide security at the modeling level which ensures that "security at the system architecture stage will improve the requirements for that system".

2. Introduction to Web Services Security Architectures

Introduction Service Orientation Engineering (SOE) (or Web Services) and Agile modeling software development presents promising solutions for contemporary software development projects to deal effectively with challenges in increasingly turbulent business environments typified by unpredictable markets, changing customer requirements, pressures of even shorter time to deliver, and rapidly advancing information technologies. Model-based agile security engineering is a promising approach that can help to fill the gap between vulnerabilities on the one hand and concrete protection mechanism on the other. Using security patterns to develop secure systems is a major research area.

Web Services Security Architectures Web Services has emerged as a dominant paradigm for constructing and composing distributed business collaborations over the web. [1] Security is one of the major concerns when developing mission critical business applications and this concern motivated the Web Services Security specifications. This paper surveys current Security Mechanisms for Web Services and Security in a Web Services World Proposed Architecture and Roadmap, which includes secure communication protocol, authentication, Signature, Encryption, Authorization, and Transport Security etc. It provides Strong ways to protect information for Browser/Server applications. Some interesting web services securing mechanisms are XKMS, SAML, XACML and WS-Security. These mechanisms can be used to provide a wide variety of web services security models and encryption technologies. The goal of the Web services security architecture is to summary out the details of message-level security from the mainstream business logic [2]. Web Services Privacy for Business Processes is important because Web Services are increasingly being adopted for business and government applications as a viable means to access web-based applications. [3]

Secure Service Oriented Analysis and Design. The first step is to analyze the application and determine the services that describe the applications. [4] The logic encapsulated by each service, the reuse of the logic encapsulated by the service, and the interfaces to the service has to be identified. From a security policy of view, in defining the services we have to consider the security policies. What is the security level of the service? What are the policies enforced on the service? Who can have access to the service? When we decompose the service into smaller services to see how we can ensure that security is not violated. The next step is for the relationship between the services, including the composition of services, to be identified. In a top-down strategy, one has to identify all the services and the relationships before conducting the detailed design and development of the services. For large application design, this may not be feasible. In the case of bottom-up design, one has to identify services and start developing them. In agile design, both strategies are integrated. From a security policy point of view, there may be policies that define the relationship between the services. Furthermore, such an approach sets the stage for orchestration-based service-oriented architectures. Orchestration essentially implements workflow logic that enables different applications to interoperate with each other. Also, we have stated orchestrations themselves may be implemented as services. Therefore, the orchestration service may be invoked or different applications also implemented as services to interoperate with each other. Business services also promote reuse. From a security point of view we have yet to determine who can involve the business logic and orchestration services. A lot of work has gone into security for workflow systems including the BFA model. Therefore, we needed to examine the principles in this work for business logic and orchestration services. When a service is reused, what happens if there are confecting policies on reuse? Also, we have to make sure that there is no security violation through reuse.

Secure Service modeling The main question is, how do you define a service? [4] At the highest level, an entire application such as order management can be one service. However, this is not desirable. At the other extreme, a business process can be broken into several steps, and each step can be a service. Te challenge is to group steps

that carry out some specific task into a service. However, when security is given consideration, then not only do we have to group steps that carry out some specific task into a service, we also have to group steps that can be meaningfully executed. If security is based on multilevel security, then we may want to assign a security level for each service. In this way, the service can be executed by someone cleared at an appropriate level. Therefore, the challenge is to group steps in a way that is meaningful not only from a task point of view but also from a security point of view. Next, we must examine the service candidates and determine the relationships between them. One service may call other services. Two services may be composed to create a composite service. This would mean identifying the boundaries and the interface, and make the composition and separations as clear as possible. Dependencies may result in complex service designs. Te service operations could be simple operations such as performing calculations or complex operations such as invoking multiple services. Here again, security may impact the relationships between the services. If two services have some relationships between them, then both services should be accessible to a group of users or users cleared at a particular level. For example, if service A and service B are tightly integrated, it may not make sense for a service C to have access to A and not to B. If A is about making a hotel reservation and B is about making a rental car reservation, then an airline reservation service C should be able to invoke both services A and B. Once the candidate services and the service operations are indentified, the next step is to refine the candidates and state the design of the services and the service operations. Therefore, from a security point of view, we have to refine the services and service operations that are not only meaningful but also secure. Mapping of the candidate service to the actual service has to be carried out according to the policies.

Secure Services Web Services and service-oriented architectures are at the heart of the next-generation web.[4] They are expected to make use of Semantic Web technologies to generate machine understandable web pages. Major initiatives like global information grid and network centric enterprise services are based o n web services and Service oriented architectures. A unified approach with a security model is required for securing Services Oriented Analysis and Design Life cycle.

Dependable Systems Dependability includes trust, privacy and integrity.[4] Multilevel security is a part of dependability. Trust management and negotiation techniques should take the advantage of semantic web technologies. Standards such as P3P and appropriate technologies that enforce various privacy policies needs to be researched.

Secure SOAD approach using Secure UML for Services: Secure UML for services essentially developed secure UML for service-oriented analysis and modeling.[4] Several approaches to applying UML and other object-oriented analysis and design approaches to secure applications have been proposed. We need to extend these approaches to secure SOAD. We also need to examine the security impact on service-oriented discovery and analysis modeling, service-oriented business integration modeling, service-oriented logical design modeling, service-oriented conceptual architecture modeling, and service-oriented logical architecture modeling. [5]

Designing Solutions for Dependability: Layered Services Design Consider using a layered approach to designing service applications and avoid tight coupling across layers.[10] Separate the business rules and data access functions into distinct components where appropriate. Use abstraction to provide an interface into the business layer. This abstraction can be implemented by using public object interface definitions, abstract base classes, or messaging.

3. Business Process Execution Language RBAC and Privacy

BPEL RBAC Security for Workflow and Business Processing, focuses on an important component that makes it possible to build and manage complex applications.[11] 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.[12]

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 create 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 will create 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 will develop graphical user interface, connecting the several web services using BPEL and making the connection with the database for accessing the web services. Third we will show 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. Refer to the Figure 1, Figure 2, Figure 3 below, which consists of sequence diagram, class diagram and execution screen shot respectively of this BPEL editor application.

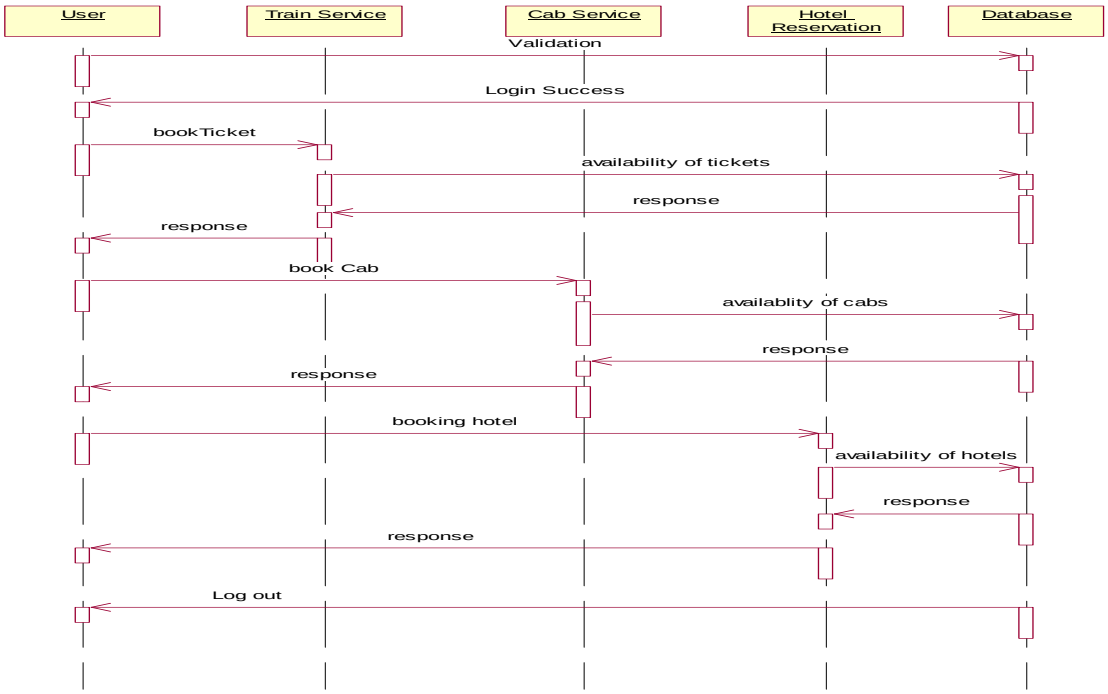

Fig. 1. Sequence diagram of the BPEL Editor application

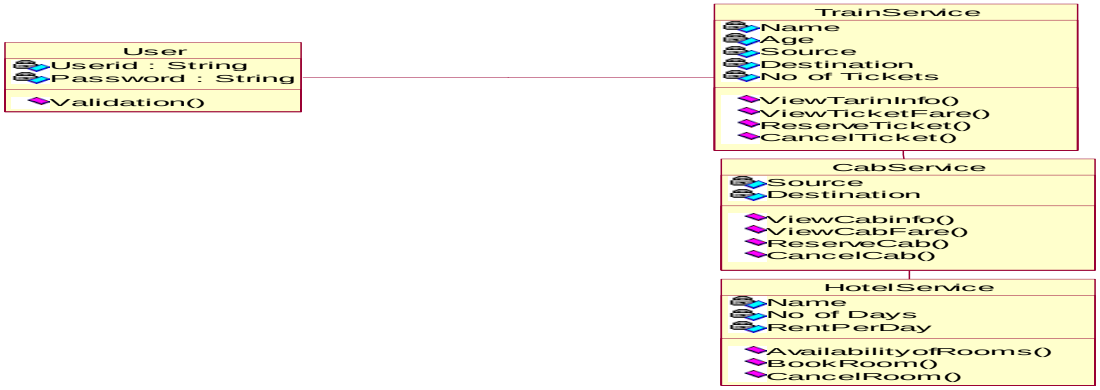

Fig. 2. Class diagram of the BPEL Editor application

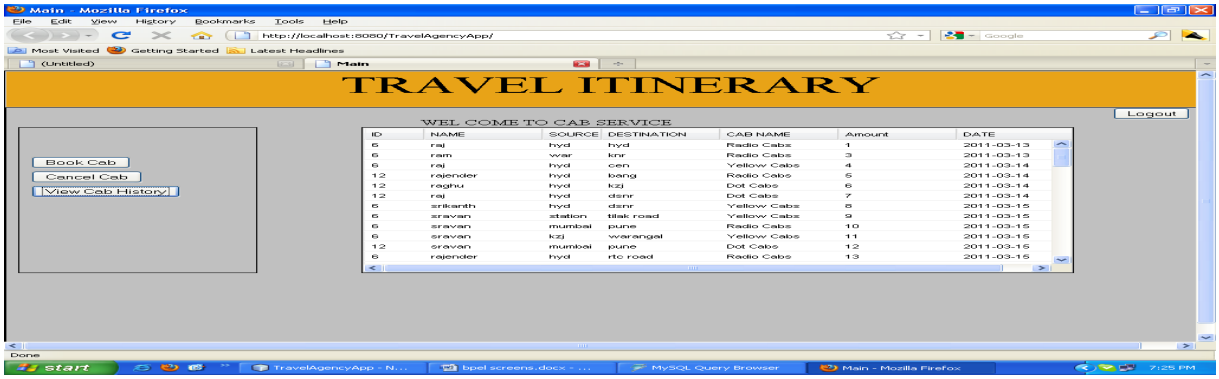

Fig. 3. Execution screen shot of the BPEL Editor application

4. Implementations and Validations

Secure Stock Market using Web Services

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, Order. And keep track of the submitted orders and their status using the orders view.

Refer to tbe Figures 4 through 7, which depicts the implementations of this case study application: architecture diagram, class diagram, sequence diagram and screen shot of this application respectively.

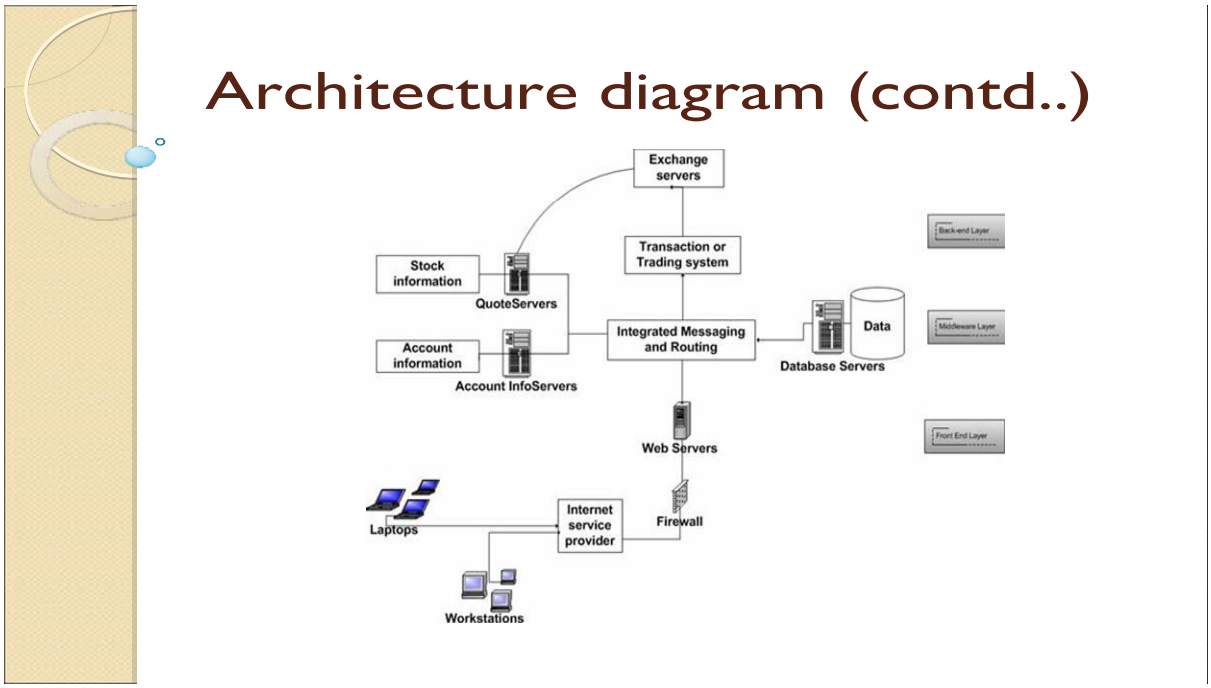

Figure 4 .Architecture Diagram for Secure Web Services Stock Market application

.

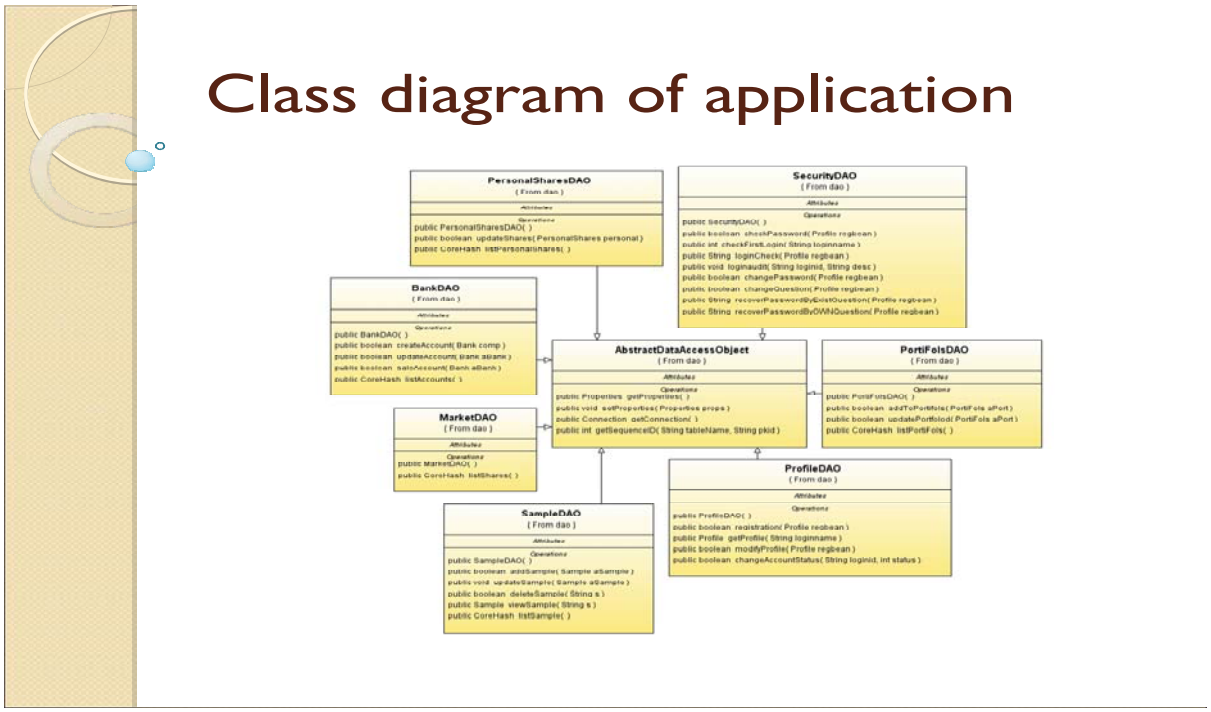

Figure 5.Class Diagram for Secure Web Services Stock Market application

.

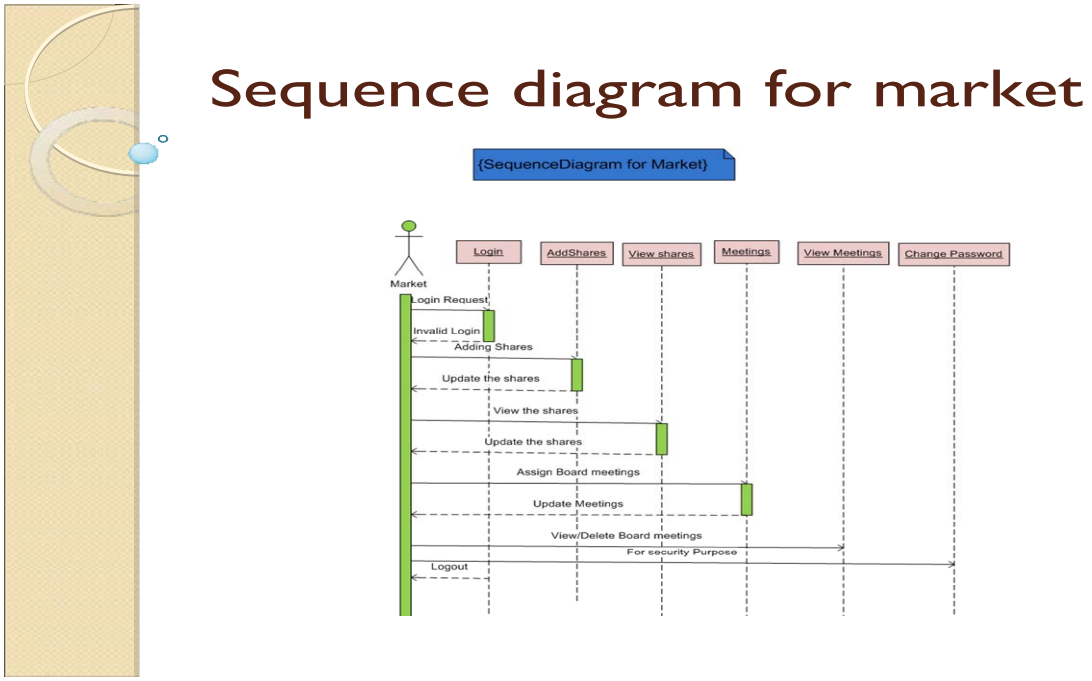

Figure 6.Sequence Diagram for Secure Web Services Stock Market application

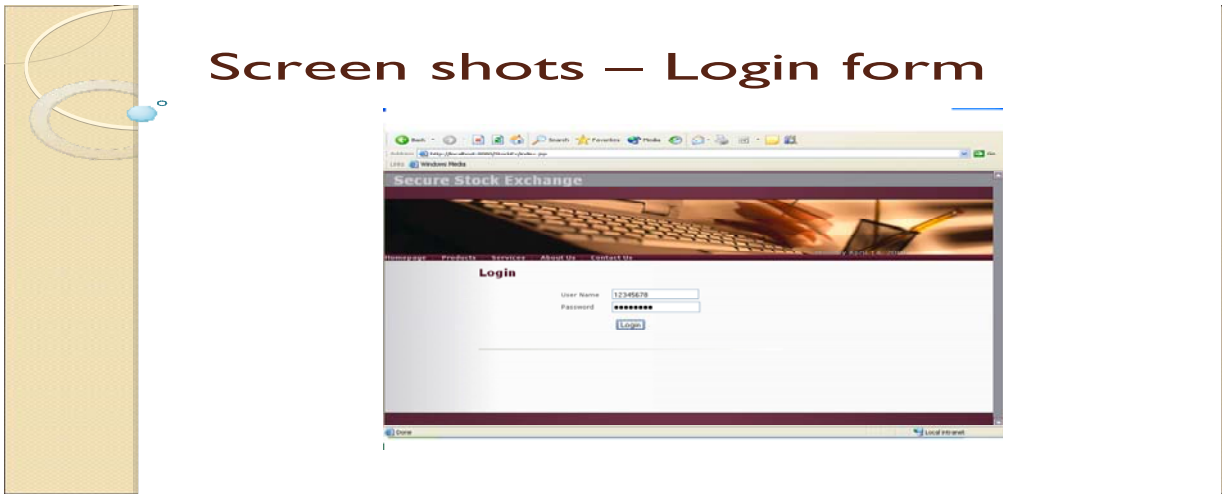

Figure 7.Screen shot for Secure Web Services Stock Market application

For details of implementation code, documentation and detailed UML diagrams, please refer to the website http://sites.google.com/site/upendramgitcse/

5. Conclusions

In this paper we discussed about designing dependable web services security architecture solutions using layered agile modeling with appropriate case studies of BPEL RBAC and Privacy and another secure application. Future work includes developing privacy and RBAC for Business Processes for Business Intelligence applications which provide insights of Web Science and Web Engineering applications and Cloud computing virtualizations. Also securing Web services contract design, its vulnerability detection, prediction of effects of these detected vulnerabilities, for the business process under consideration needs to be examined. These security patterns needs to be formalized and model checked, verified for security, tool supported, detected threats needs to be stopped or mitigated, and run-time monitoring is required.

References

[1] Marzouk.S.Mokbel, Le Jiajin: (2005 – 2008) Integrated Security Architecture for Web Services and this Challenging: In Journal of Theoretical and Applied Information Technology JATIT pp. 518 – 525

[2] Ma li, Jiang Cheng-yan: (2009) A research on Web Security Service Architecture: in Journal of Chongqing Electric Power College China

[3] Nina Godbole: (2009) Information Systems: Security Management, Metrics, Frameworks and Best Practices. Wiley India Publishers

[4] Bhavani Thuraisingham: (2011) Secure Semantic Service Oriented Systems: Auerbach Publications

[5] James S.Tiller: (2011) Adaptive Security management Architecture. Auerbach Publications

[6] Kanchan Hans: (2010) Cutting edge practices for Secure Software Engineering: in International Journal of Computer Science and Security IJCSS Voume 4 Issue 4 pp. 403 – 408

[7] 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

[8] Eduardo B.Fernandez, Nobukazu Yoshika, Hironori Washizaki, Jan Jurjens, Michael VanHilst, Guenther Pernul: (2011) Using Security Patterns to Develop Secure Systems: DOI: 10.4018/978-1-61520-837-1.ch002 pp. 16 – 31, IGI Global

[9] George Spanoudakis and Andrea Zisman: (2010) Discovering Services during Service-Based System Design Using UML. In IEEE Transactions on Software Engineering, Vol 36, No.3, May/June 2010, PP 371 – 389

[10] David Hill: (2009) Microsoft Application Architecture Guide, Patterns and Practices. Second Edition, Microsoft Press

[11] Elisa Bertino, Lorenzo D Martino, Federica Paci, Anna C Squicciarini: (2010) Security for Web Services and Service Oriented Architectures. Springer Publisher book

[12] Li Liang Xian: (2011) Research of B2B e-Business Application and development technology based on SOA: In W.W.Song et. al. (eds) Information Systems Development Springer