POLITECNICO DI MILANO

Security Aspects in Evolution of Enterprise Business Applications Framework

A graduate thesis submitted to the

Department of Elettronica e Informazione

As a partial fulfillment for the award of degree

Master of Science in Engineering of Computing Systems

Advisor:

Professor Stefano Zanero

Co-Advisor:

David Frontini

Author:

Hashim Ali, 737149

Academic Year 2010/11

Table of Contents

List of Figures

List of Tables

Acknowledgments

The work presented hereby is my first little endeavor in the field of computer security. Firstly, I would like to thank Prof. Stefano Zanero for his course on Computer Security which inspired me to dive in this domain. Later, he provided an opportunity of practical work along with his supervision, support and courage which helped me to finish this with ease.

Then my heartily thanks to all team of Cargo Clay S.r.l; specially Mr. David Frontini for his support during the whole project time.

Lastly, I am grateful to my brother Qasim Ali, my sister Sadia Akhtar, my sweetest Camila Cerda, and unforgettable my parents emotional strength, support and prayers to me all the time.

Hashim Ali

Abstract

Service-oriented architectures (SOA) is an emerging approach that addresses the requirements of loosely coupled, standards-based, and protocol independent distributed computing. Typically business operations running in an SOA comprise a number of invocations of these different components, often in an event-driven or asynchronous fashion that reflects the underlying business process needs. To build an SOA a highly distributable communications and integration backbone is required. This functionality is provided by the Enterprise Service Bus (ESB) that is an integration platform that utilizes Web services standards to support a wide variety of communications patterns over multiple transport protocols and deliver value-added capabilities for SOA applications (1).

The work presented hereby is study/analysis of existing Mind4b; web application provides supply chain management platform, architecture and implemented security features. Such review requires knowledge of understanding of web information system architecture, communication protocols, different integration patterns and typical security mechanisms. First is introduction of fundamental concepts of web systems leading towards aims of the project under consideration which will help to formalize the structure of the project. The next part is being focused on different web information systems integration; which may be either point-to-point messaging or message broking framework with their corresponding security issues; which mainly contain user authentication/authorization, and management of access control list, which further will help in transforming existing web application architecture to message service bus framework. The introduction of enterprise service bus architecture and supported security mechanisms and associated protocols to obtain web information systems scalability, effectiveness and increased performance. In the last there is description of improved web architecture and security features and future work considerations.

1. Introduction

1.1 Overview

Nowadays, one of the major sources of information is internet which consists of either static web pages or dynamic web applications. These web pages (which are hypertext documents) are accessed through internet using World Wide Web (WWW). The pioneer of WWW is Tim Berners-Lee who was working at CERN in order to share documents among scientists. Hypertext Markup Language is used to create web pages; HTTP is used to transfer these web pages to remote client and web server in order to serve those pages to web browsers. Later and still there is remarkable growth of web pages (information) and internet users. Although, web pages are documents which reside on the server side and didn't take long as web browsers are thin-client.

Web applications, which are enriched form of web pages, are capable of performing business logic on web pages. The development of web applications distinguished from normal desktop applications because of no installation requirement and ready to serve on user demand and delivered over standard web browsers. Initially, these web applications performed all work on the server side and thus produced lower response as compared to desktop applications. Later, high speed computers with new web technologies overcome these issues.

Moving web applications from server to client side created new problems, especially from security point of view. As, in desktop applications users are independent and have no concern with other users, so any breakage or harm to application from user is limited to that specific user. While web applications are linked with many users over the internet so web application development needs more consideration for effective and secure working besides its performance, reliability.

1.2. Motivation

Currently, there are millions of websites available over internet for users and these websites may belong to Individuals, Commercial, Organizational, Educational, Entertainment, News, Blogs, and Hybrid. Ensuring secure access to websites is one of key issues and requires efficient security protocols for the successful operation and expansion of their deployments.

The ultimate goal of such information systems is to build enough confidence level of the user in the network and service provider, it is must that subscribers are assured that their communication across the network is quite secure and is not going to be a victim of any vulnerability. The key objectives of incorporating security to information systems are Confidentiality, Integrity, and Availability (known as CIA paradigm). Typically, incorporating security to any system has to deal with Authentication, Authorization, and Auditing (known as AAA paradigm).

Security is one of the essential rights of everyone life. Web security has also become an integrated part of our daily life. People are conducting their personal and job-related business using these sites. Many consumers purchase goods and services online using their credit card information. A large number of people also quit conventional banking because of online banking. Moreover, selling and purchase stock through broker websites is just one click away. Similarly, industries are interconnected through web too, and major part of it deals with electronic transactions. Due to sensitive nature of these sites, security is always considered at priority level. Besides deploying security protocols like SSL, corporate sector also hire security experts to audit and vulnerability assessments. Regardless of many security mechanisms, security is still one of the major concerns for institutions who offer secure websites for their potential users (2).

The work presented herein deals with security review of improved framework of corporate web solution Mind4b (SCM Web Solution) as explained in (3). The Mind4b is owned by Cargo Clay S.r.l, Italy which is a logistical company dealing with many customers all over Europe and North America.

1.3. Aim of Project

This project is sort of reverse engineering solution, below are the aims that have been defined while focusing on enriching web architecture and enhancement of secure access and efficient utilization of network resources. The aims have been divided into two phases to facilitate the whole cycle:

1.3.1. Phase – I

- To understand typical security implementation in web information systems, this includes user authentication/authorization protocols.
- To study and understanding currently implemented Mind4b web framework.
- To investigate Mind4b Business Model, User Request Management, and User Role Management based on Access Control List.
- To analyze issues regarding reconstruction of Mind4b web architecture.
- To analyze issues regarding customization of Access Control List.
- To analyze issues regarding customization of database schema.

1.3.2. Phase – II

- Introducing Enterprise Service Bus framework in order to integrate different business partners regardless of communication protocol.
- To design strategy of separating user authentication/authorization model, from rest of business model, which would be flexible against reported threats and solves issues more effectively.
- To reduce storage memory overhead by removing redundant components.
- To analyze new designed model against reported threats and tests the performance.

1.4. Thesis Structure

The thesis presented herein has been organized as follows: Chapter 2 discusses the background study, Chapter 3 provides an overview of current Mind4b architecture, evolution process to achieve flexibility, scalability and better performance and later challenges ahead, Chapter 4 describes Mule - Enterprise Service Bus Solution, and its supported security features for enterprise web applications and finally, Chapter 5 presents the implementation of proposed solution and future work.

2. State of the Art

2.1. Overview

The research regarding web architecture of enterprise business web applications which is reliable, scalable and high performance containing efficient security mechanisms needs understanding of technology which is being explored. This chapter explains security aspects in integration of business partners; with different communication protocols, to business organization running enterprise application based on Service Oriented Architecture (SOA), and Enterprise Service Bus (ESB) architecture.

2.2. Security Aspects in Integration and Enterprise Service Bus (ESB)

Enterprise Web Applications dealing with Supply Chain Management systems, Customer Relationship Management, Business Intelligence systems, and many others having diverse nature of communication protocols create problems while integration. Due to different communication protocols initiate inefficiencies; wherein identical data is stored in multiple locations due to slightly different data structures, or straightforward business processes are unable to be automated. In such scenario, there is a need of integration framework which is capable of integrating such applications within one organization together in order to automate business process without changing existing implemented architecture and data structures. Enterprise Application Integration is an integration framework which is composed of different technologies, communication architectures, services which acts as a middleware to enable integration of systems and application across enterprise business.

In the coming sections, we will see different possibilities of integration techniques and corresponding security issues which will lead us to move new architecture framework named as Enterprise Service Bus.

2.2.1. Point-to-Point Messaging Integration

One common way of integration is by using Point-to-Point Messaging which ensures that only one receiver will consume the message. The exchange of messaging is accomplished by maintaining queue: senders (Business Partners) produce messages to queue and receiver (Business Organization) will consume respective messages as shown in **Error! Reference source not found.**.

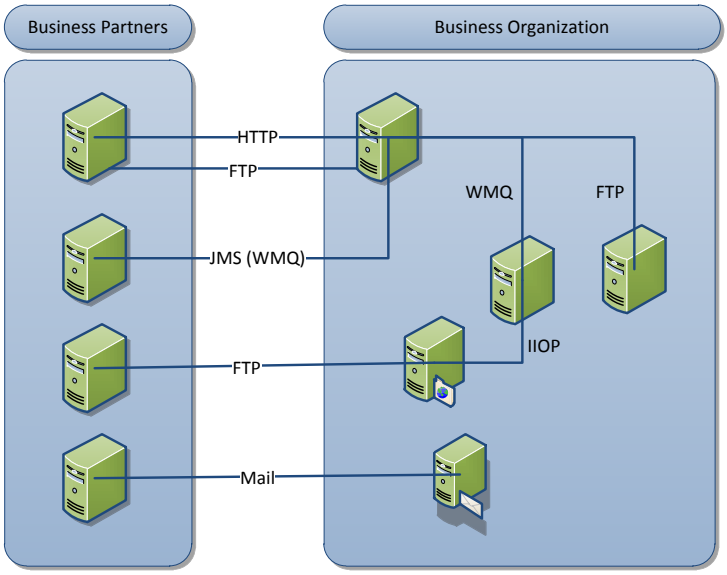

Figure 2.2-1: Integration Using Point-to-Point Messaging

Error! Reference source not found. further elaborates different scenarios about point-to-point messaging (4).

- More than one sender can produce and send messages to a queue. Senders can share a connection or use different connections, but they can all access the same queue.
- More than one receiver can consume messages from a queue, but each message can be consumed by only one receiver.
- Receivers can share a connection or use different connections, but they can all access the same queue.

- Senders and receivers have no timing dependencies: the receiver can consume a message whether or not it was running when the sender produced and sent the message.
- Messages are placed in a queue in the order they are produced, but the order in which they are consumed depends on factors such as message expiration date, message priority, whether a selector is used in consuming messages, and the relative message processing rate of the consumers.
- Senders and receivers can be added and deleted dynamically at runtime, thus allowing the messaging system to expand or contract as needed.

Advantages of point-to-point messaging are:

- Messages destined for a queue are always retained, even if there are no receivers.
- Java clients can use a queue browser object to inspect the contents of a queue. They can then consume messages based on the information gained from this inspection. That is, although the consumption model is normally FIFO (first in, first out), receivers can consume messages that are not at the head of the queue by using message selectors. Administrative clients can also use the queue browser to monitor the contents of a queue.
- The fact that multiple receivers can consume messages from the same queue allows you to use load-balancing to scale message consumption if the order in which messages are received is not important.

Security Threats

Point-to-Point messaging network welcomes different external threats; which involve worms, virus-attacks, confidentiality, authentication/authorization, and internal threats; which involve interoperability, private business on public network, adding and removing users. Moreover, such strategy can also allow employee to download or share important business data which violates business intellectual security procedures.

Security Mechanisms & Protocols

To have authenticated communication among business partners, there is need to deploy security mechanisms which can be either based on symmetric keys or asymmetric key cryptography, or maybe combination of both. Then authorization protocol based on strong cryptography which ensures communication being established is verified.

Commonly implemented protocols are:

- Secure Socket Layer (SSL)
- IPSec Technologies
- Public Key Infrastructure (PKI)

Virtual Private Network (VPN) Security

VPN, the network which is private over public, is common solution adopted by business organization to have secure communication among peers as shown in Figure 2.2-2. VPN hardware/software is compatible with IPSec technologies in which authentication mechanism is not user-based rather client's IP address or certificate associated to it (e.g. X.509) by establishing user identity and ensuring network integrity. VPN also needs firewalling to have secure network, as it secures only communication tunnel. In VPN enabled network, communication model is not centralized due to nucleated security zone which lacks common way of auditing.

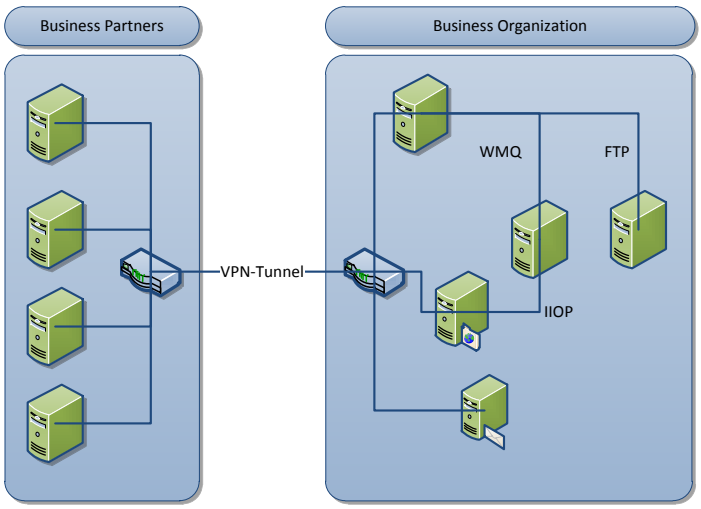

Figure 2.2-2: VPN Security in Point-to-Point Messaging

File Transfer Protocol (FTP) Security Issues

The FTP is most widely used network protocol by enterprises for sharing files over TCP-based network (Internet). FTP supports web information systems established over client-server architecture. FTP doesn't have secure communication due to its protocol structure and has many holes from security point of view. Furthermore, communication over FTP doesn't have file checksum (hash code calculation) as well. In May, 1999 authors of RFC 2577 (5) elaborates FTP security flaws, which are:

- Bounce attacks
- Spoof attacks
- Brute force attacks
- Packet capture (sniffing)
- Username protection
- Port stealing

Moreover, there is no encryption strategy for FTP communication, so everything which is being transferred over network is plain text which can be sniffed easily over network. Later, there are some security mechanisms implemented such TLS and SSL which helps to achieve secure FTP communication. There are different methods to provide secured FTP which includes FTPS, FTP over SSH (manages FTP session by creating tunnel over SSH connection). Secure-FTP also provides file checksums besides encryption.

Hypertext Transfer Protocol (HTTP) Security Issues

The HTTP is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP offers request-response protocol in client-server architecture paradigm (6).

HTTP standard defines two methods of authentication for the web. These are Basic and Digest Authentication as explained below:

Basic Authentication: When a protected resource is accessed by client software, the server software response with the HTTP 401 authorized status. The client requests user

name and password from user. Then it uses Base64 algorithm to encode the combined username and password. The Base64 encoding means no encryption at all.

Digest Authentication: In addition to basic authentication, the server response with WWW-Authenticate header that request digest authentication. The authentication digest is computed based on username, realm, password, request method and requested resource. It is computed using the MD5 hash algorithm, challenge response, and some other technique. The digest authentication is designed to work over unencrypted communication links.

The biggest problem of provided authentication mechanisms of the HTTP protocol is that they provide very little support for session management, not to mention only username/password is supported and both techniques are vulnerable to several attacking techniques (7). Since there is no way that the server might determine that user has been logged out or not, HTTP authentication is not widely used.

Certificates Based Authentication: Certificates provide excellence mean for web authentication. Certificates authentication is very promising. It is developed with the help of public key encryption and is supported in the SSL protocol. More information on encryption and the SSL protocol is discussed in the next sections.

Denial-of-Service Attacks over HTTP

Denial-of-Service attacks are based mainly on the weakness of the TCP/IP protocol suite. They can be sending SYN flood packets (8); implementing a lot of attack agents (9), or using a lot of “reflectors” in combination with address spoofing (10). All of these are to consume all available resource for the system to function. Denial-of-service attacks are very hard to prevent, mostly because of the design of the TCP/IP protocol suite.

HTTP A/Synchronous Message Delivery

Messages delivered over HTTP required high and continuous availability at both systems. In synchronous message delivery, if commutation breaks between requester and responder during message delivery, message might get lost or duplicated which is shown in Figure 2.2-3. There was a proposal to improve this issue by introducing HTTP-

Reliable which works at protocol level but no industry support this protocol nowadays. The working flow of HTTP-Reliable is shown in Figure 2.2-4.

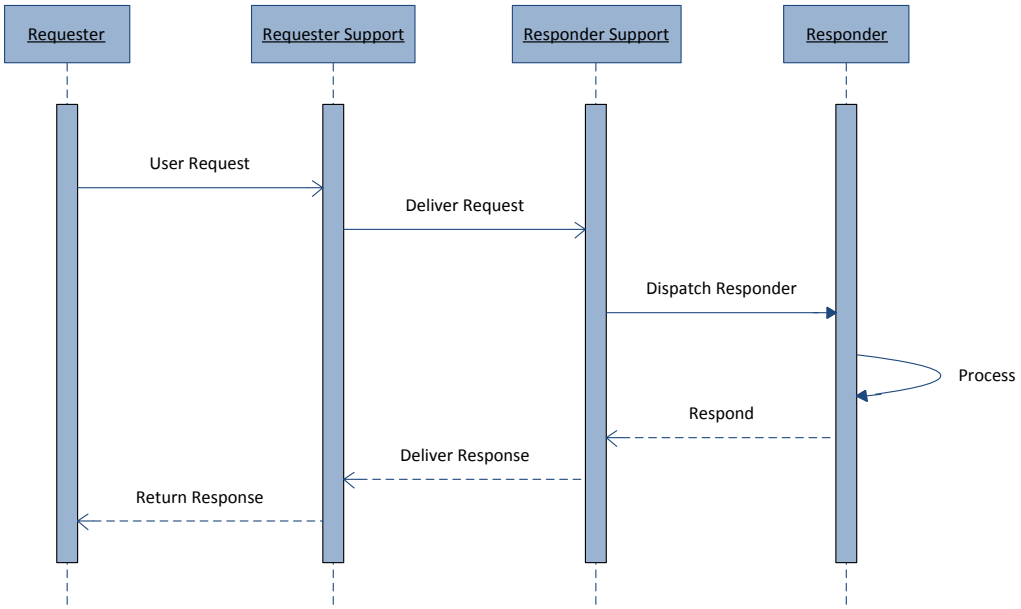

Figure 2.2-3: HTTP – Problem, Synchronous Message Delivery

For message delivery assurance, messages should be delivered once (not zero times) and only once (no duplicates) and the better way to guarantee message delivery is design your systems to communicate asynchronous messaging between systems. There are different ways to implement asynchronous messaging, for example JMS, Websphere MQ etc.

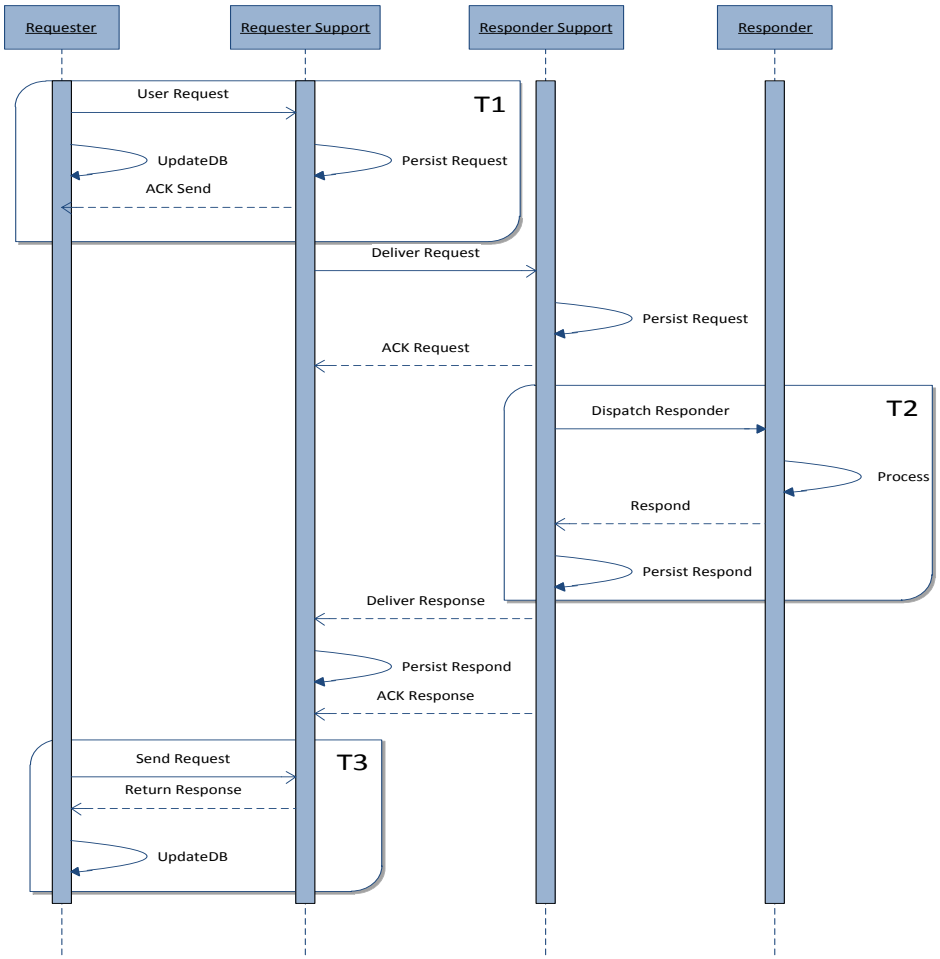

Figure 2.2-4: Protocol Level Solution for Synchronous Message Delivery

Point-to-point vs. end-to-end security in messaging

The key points are:

- Point-to-point messaging secures communications between business partners
  - Origin of the data message is business partner(identified as organization)
  - Message level and transport layer security can used
- End-to-end messaging security
  - Origin of the data message is business partner service requestor and
    - With message level security this is hard to archive unless originating systems use same format all the ways.
- Transport layer security can only archived using federated security i.e. both business-partners access management software have trust each other and able to federate identities

2.2.2. Integration using Message Brokering

Message Broker is a centralized manager for validation, transforming and routing messages to specific destinations. Different communication channels are supported through adapters where security solutions are dependent in adapter's implementation. This means security is still protocol specific in integration using Message Brokering as shown in Figure 2.2-5. Message Broker typically is a solution of transforming and routing messages, not a security gateway.

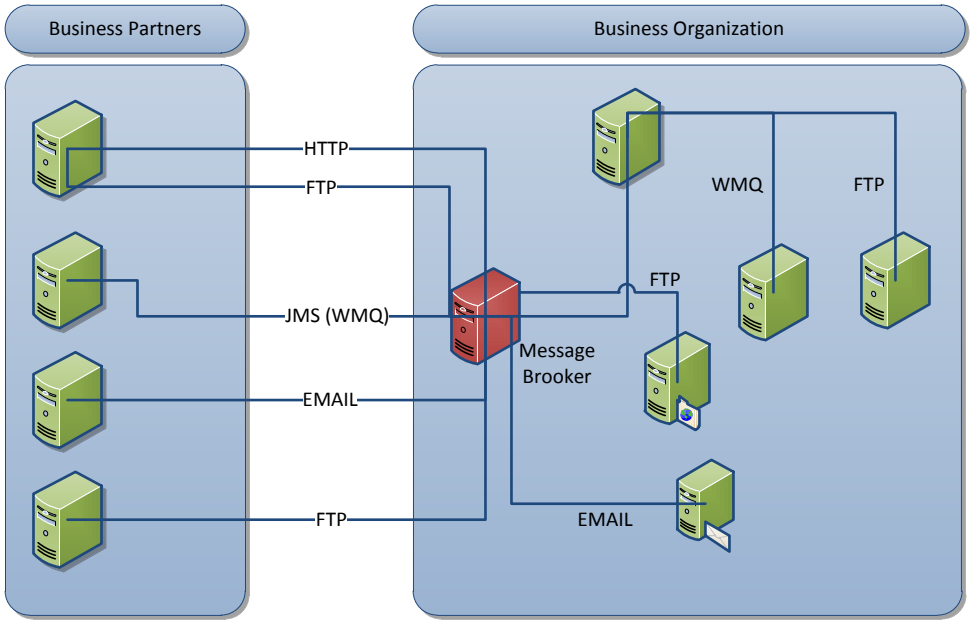

Figure 2.2-5: Common Integration Pattern Using Message Brokering

The main difference between Message Broker and ESB are:

- Message Broker is intermediately program that relies on certain underlying protocol(s) and uses it's features
- ESB is software architecture construction
- ESB uses normalized messages and doesn't use protocol specific features

2.2.3. Introduction to ESB and Security Constraints

"As more systems designed with SOA technology are being developed, the backbone of the system is predominately the ESB. An ESB is infrastructure that allows different applications to communicate via standards-based, port-level communication. It provides the services to communicate via different protocols, such as UDP and HTTP along with the infrastructure to provide messaging capabilities such as publish and subscribe,

request and response, and asynchronous messaging. Several corporations have implemented ESB applications, such as Mule™ and BEA AquaLogic™. ESB applications are amongst the tools and software that organizations are collecting to provide the infrastructure for their SOA systems, i.e. the SOAIF.

In order to deploy services on an ESB, the services are established as endpoints. Each service is connected to the ESB using a particular type of endpoint a variety of which are available depending on the ESB chosen. For example, the Mule ESB provides several different options for types of endpoints including inbound routers, outbound routers, components, catch-all strategies and exception strategies. Inbound routers receive requests from other services, which may reside locally on the ESB or may reside somewhere across the Internet. The inbound router then determines if the request should be routed into the system or rejected based on filtering criteria. Outbound routers are used to send events once the processing has been completed by a service.

ESB endpoints are built by creating a configuration file for the ESB to use as a deployment descriptor. The ESB configuration file is an Extensible Markup Language (XML) document that describes the types of endpoints, connectors, properties, transformers, and more being built for this particular SOA application. The configuration file describes the connections between the components of the SOA application and is the enabling concept behind one of the major advantages of SOA, namely loose coupling. The services in a SOA are loosely coupled if they do not rely on the underlying implementation of other services. They are also loosely coupled with respect to the provider and consumer roles if the provider does not know all the consumers of the service. The ESB enables loosely coupled services by providing the infrastructure and implementation details allowing the services to abstract away the low-level details of the connections between services. A service provider can simply respond to a request coming across the ESB without knowing who made the request or where the response should be routed to. The ESB configuration file maps the necessary connections

between the services and during execution the ESB routes the messages appropriately according to the configuration file (11).”

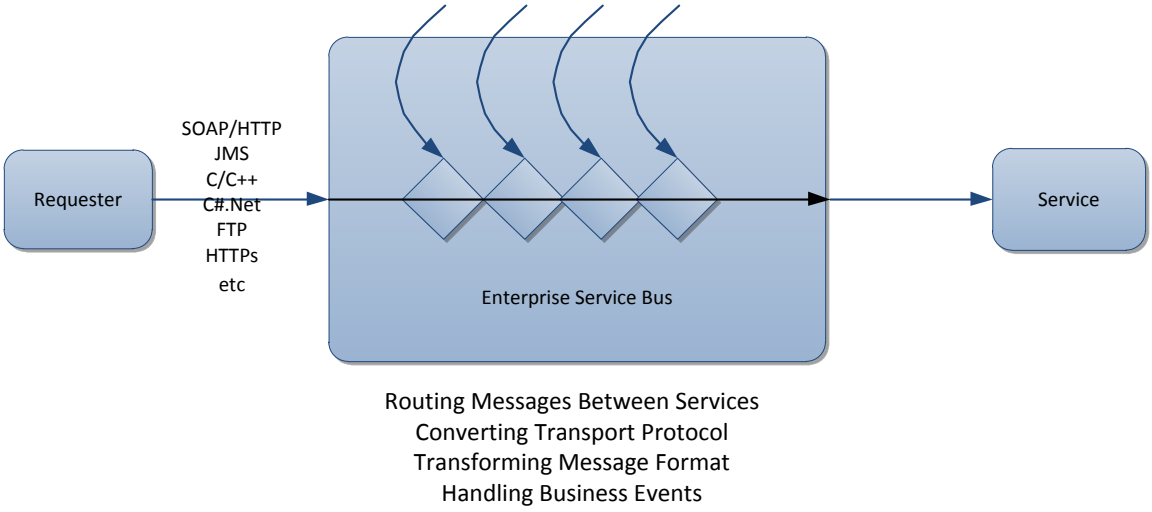

Figure 2.2-6: Enterprise Service Bus

Availability threats

- To make services and ESB available
  - Capacity planning
  - Find the weakest link of the chain
    - Design pattern in integration should be Event Driven rather than State-full
    - Event driven architecture scales to infinity
- Network rate limiting
- Monitor service and ESB processes
- Monitor service logs
- B2B Monitoring is also needed

3. Analysis of Current Mind4b System

3.1. Mind4b Overview

Mind4b, a Collaborative-Supply Chain Management (C-SCM) platform, was initiated by Cargo Clay for effective organization of logistical supply chain. Mind4b simplifies the challenges in front of stakeholders maintaining bigger Supply Chain. Mind4b target customers are manufacturers, distributors, importers, logistic operators that need to have a full governance of the Supply Chain (from factories distribution centers to retailers/end users) in order to have stronger customer loyalty, higher margins and a closer relationships with final users. It provides online SAAS (Software as a Service) support together with a clear and organized overview of the logistic chain which is start of improving business performance and time management. This is accomplished through process optimization and flexible access to the program.
The main features of Mind4b are:

- Collaboration-Based Management
- Integration with all IT Platforms
- End-to-End Visibility and Control over the Entire Logistics Chain
- Ease of Operation and Custom Configuration of Access

3.1.1. Supply Chain Management (SCM)

By definition, SCM is tightly coupled network of organizations by coordinating flow of information, material and financial flows. This network is involved in different processes and activities that produce value in terms of products and services in the hands of ultimate customer.

Due to rapid changes in global market economy, organizations find themselves in complex situation where the key component is to exchange information among stakeholders in more effective, quickest and reliable way. These intentions lead the companies to have broad picture of SCM at any time and put corrective measures into

place as needed. This requires in-depth understanding of clients for maximizing the profit.

All this requires outsourcing many functions, which diminishes a company's level of direct control in these areas. Third parties often employ their own systems and standards which can be completely different from their partners. These differences have a tremendous impact on a company's ability to offer value and compete efficiently. It's clearly necessary, therefore, to transform Supply Chain Management into a new process which takes into consideration all of these influences: Collaborative Supply Chain Management, or C-SCM.

3.1.2. Support as C-SCM

Mind4b was developed by considering above scenarios and from companies demands to overcome new challenges in the global marketplace. Through its incorporation of C-SCM, Mind4b allows for the creation of a genuine Supply Chain community. This community is united by a common technology platform which is completely capable of supporting every production phase of its members.

Mind4b is a dynamic solution, able to quickly adapt itself to evolving situations. It gives each player maximum visibility throughout the entire Supply Chain. It permits the creation and monitoring of shared objectives. Finally, Mind4b assures total control and preemptive corrective measures at the first sign of deviation from the planned objectives.

3.1.3. As Cloud Structure

In the Cloud structure, obtained through innovative Software as a Service (SaaS) approach, Mind4b greatly simplifies the whole management process, making it possible for different players to simultaneously access the same information as shown in Figure 3.1-1, thus ensuring a rapid, efficient and cost-effective process.

Whether we're talking of a customer, a supplier, a freight-forwarder or a member of your own staff, any Mind4b user can contribute to managing and improving your Supply

Chain with a simple click of the mouse. As an additional bonus, every player involved will receive real-time updates on the whole process.

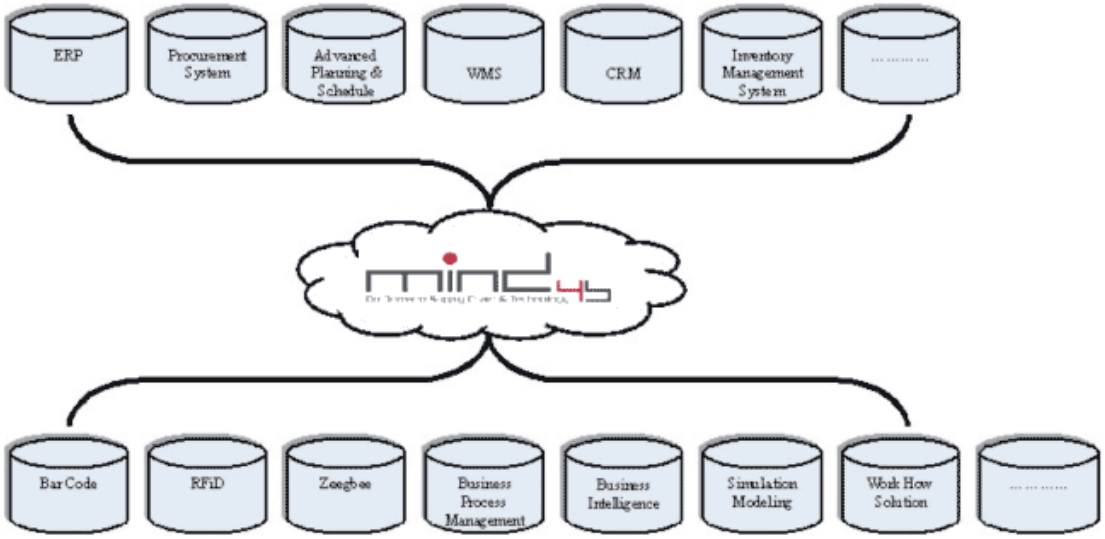

Figure 3.1-1: Mind4b, Supply Chain Cloud Services Flow

3.1.4. Web-based C-SCM

By incorporating Mind4b as Software-as-a-Service (SaaS), a company can completely eliminate the need for start-up investment in hardware or software architecture. This greatly reduces implementation time and avoids any risk of legacy system incompatibility.

A PC with an Internet connection is all that is needed in order to access the platform. Each user is granted customized access, giving the platform total security and allowing suitable levels of access restriction based on users' job functions and seniority. Mind4b is therefore capable of responding to the needs of small and medium sized companies which lack an IT department but want to avoid complex and demanding systems. It is equally adept for large multinationals that require a platform capable of serving multiple locations across time zones, with diverse linguistic or managerial needs.

3.1.5. Real-time Problem Solving

Better visibility across the entire chain automatically results in higher production capacity and improved control over potentially risky situations. Mind4b anticipates any

potential risk related to managing the Supply Chain, activating "push" type alert systems and sending specific notifications to a predefined group of users once critical situations arise. For example, if the lead time of a shipment exceeds a company's acceptable threshold, the system will send an e-mail or text message to a group of affected users, warning them of a potential delay and giving them time to rectify the situation.

3.2. Mind4b Web Framework

Mind4b-Web Application is implemented using PHP, and JAVA-specific web technologies such as Servlets and JavaServer pages (JSP), and AJAX technology. The web framework can be divided into:

- ***Mind4b-Business Model****, composed of active Use Cases, Web Architecture, and Hardware Architecture*
- ***Mind4b-Security Model****, composed of user Authentication/Authorization, Request Management, Roles Management*

3.3. Mind4b – Business Model

Mind4b has a user-friendly and intuitive interface, to ensure maximum collaboration is achieved among all players. The Business Model mainly composed of three major components consists of:

- Use Cases
- Web Architecture
- Hardware Architecture

3.3.1. Use Cases

Generally, web information systems providing logistical services, mostly, consist of two types of use cases; Business Use Cases & System Use Cases. Mind4b manages Production, Shipping, and Sales (as Business Use Cases) and System Admin, Raw

Material, Transports, Stock, Purchase Order (as System Use Cases). Mind4b acts as central point of interaction between users as shown in Figure 3.3-1.

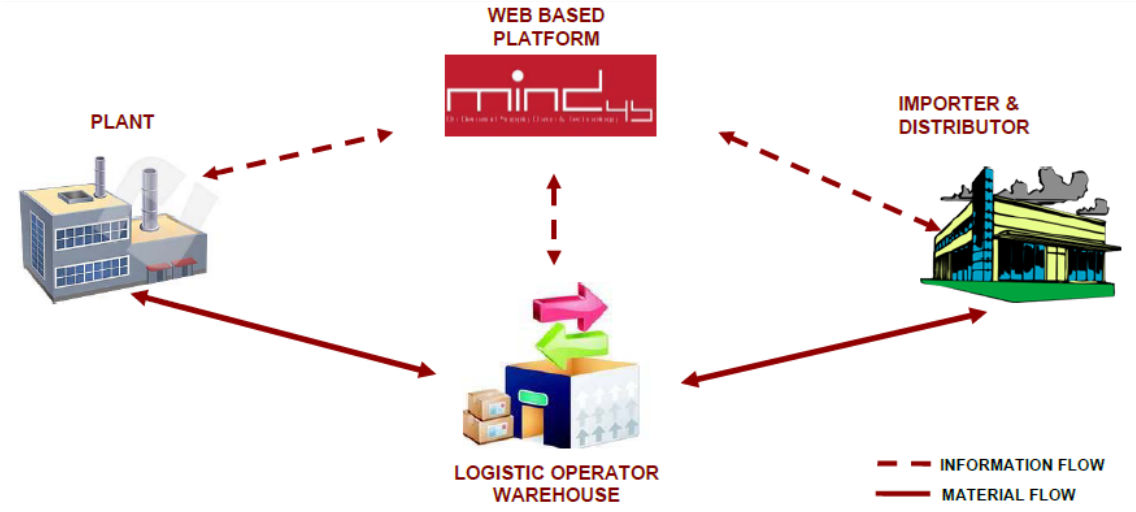

Figure 3.3-1: Mind4b, A Collaborative Platform

The business use cases defined above has the interaction between system use cases and Mind4b web system to accomplish specific goals. Mind4b web framework manages these use cases as its features (Roles). Each feature further composed of different resources, for example; Sales feature contain resources as New Sales Order, View Sales Order, Modify Sales Order etc.

3.3.2. Web Architecture

Currently, Mind4b follows 2-tier web architecture based on Client-Server Model paradigm. The Web Server based in Apache resides Web Application containing PHP files is behind external firewall and Database Server is under internal firewall running PostgreSQL as shown in Figure 3.3-2.

The platform can also interface with virtually any communication device (email, mobile or other alert systems) to keep every operator constantly updated on the status of the process through “push” technology, this means you won't have to wait for a third party to log into the system before a potential issue is identified.

This allows our customers to anticipate virtually every potential problem or delay, and translates into further time and cost savings.

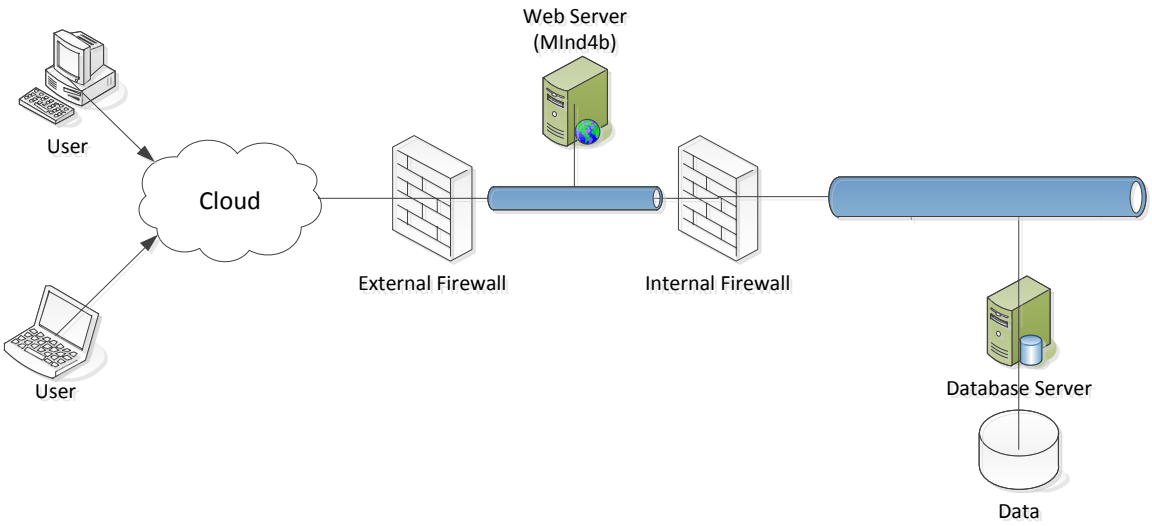

Figure 3.3-2: Current Web Framework of Mind4b

3.3.3. Hardware Architecture

The Hardware Architecture is simply composed of two server machines shown in Figure 3.3-3, Web Server and Database Server, as it is obvious from the web architecture of Mind4b.

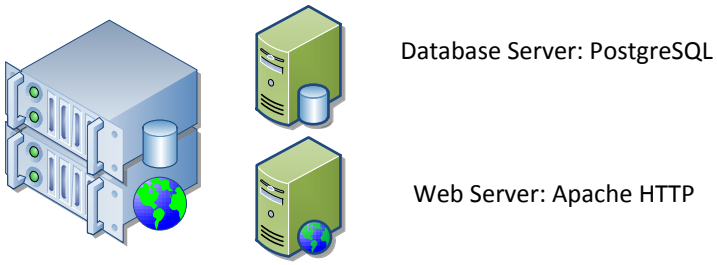

Figure 3.3-3: Current Hardware Architecture of Mind4b

3.4. Mind4b – Security Model

To ensure secure access to web application developed for E-business needs implementation of security features as well. The Security Model is composed of User Authentication/Authorization, Request Management and Roles Management. The general security flow model of web application is shown in Figure 3.4-1.

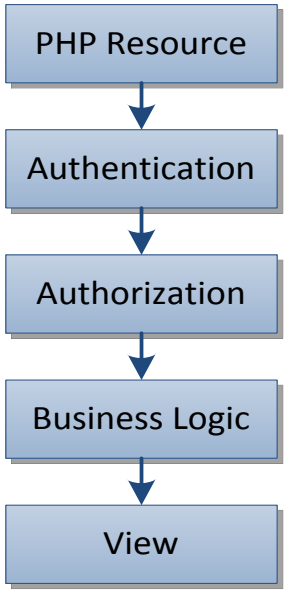

Figure 3.4-1: Security Model

3.4.1. Authentication & Authorization

In Mind4b, all PHP requests made by client are considered as resources available to the system. When client's browser connects to Mind4b, it stores state of application on the server side and respond the request with login page for authentication. After successful user authentication, Mind4b sends cookie identifying that session has been created between server and client. This session cookie contains useful information about the user. Authenticated user is then bound to session and later identified by session cookie residing on client side. Security Model assumes that only one user and one server can have knowledge the value of session cookie. Mind4b Authorization phase is managed by Role Based Access Control Lists which contain (user, resource) tuple. Only those resources for which authenticated user are allowed will be accessible.

In the next section, we will see in detail current implemented security management.

3.4.2. User Request Management

In the current version there are four types of resources available for user interface.

- User Login Management
- Requests with Context Menu
- AJAX type requests displayed in IFRAME
- AJAX type requests used to populate information in form

Any request made by a browser Mind4B is encoded by a PHP file that implements the business logic wherever possible yielding classes provided the frame work m4b. In the jargon Mind4B each request is identified by resource availability.

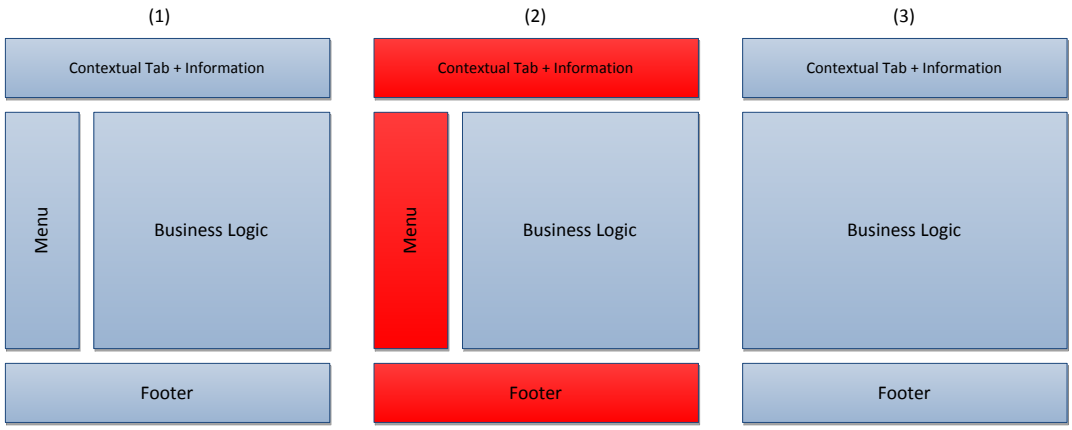

Figure 3.4-2: Screen Types Coded in Mind4b

Figure 3.4-2 shows different types of graphics where:

- Requests with context menu - full graphics Mind4B
- AJAX type requests displayed in DIV - with full graphics, although included in a complete graphical model
- Requests with context menu - full graphics Mind4B
- AJAX type requests displayed in IFRAME - with full graphical

Login Management

In order to protect information from unauthorized access, Mind4b use Login/Password authentication method. This authentication scheme is most commonly used in most of the web information systems based on HTTP sessions. After verifying the correctness of use credentials, authentication token as M4B_Token is created and send to client browser. PHP pages involved in the management are shown in Figure 3.4-3.

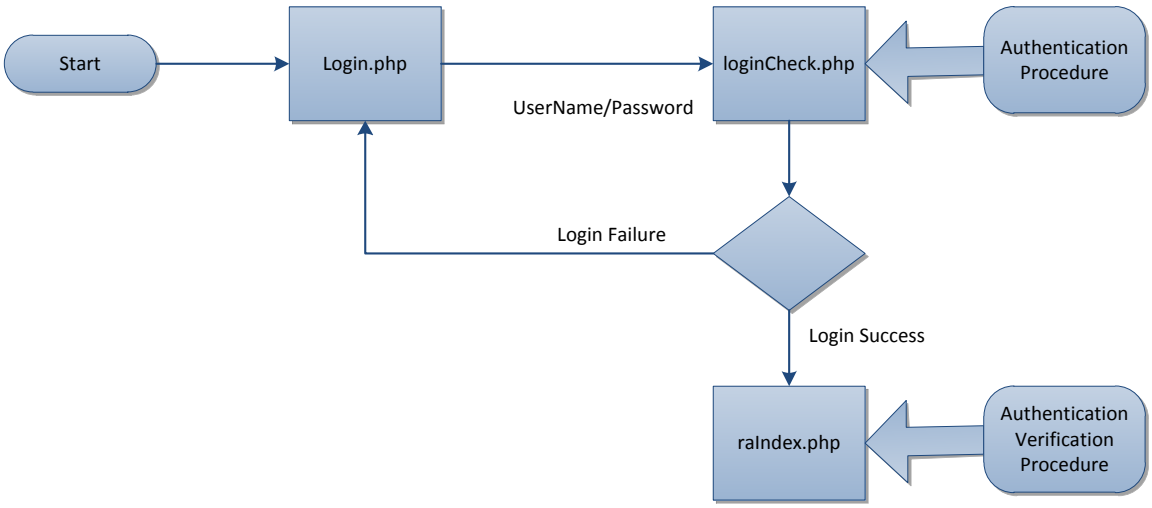

Figure 3.4-3: User Request Management

- Login.php: It manages the login page which shows login form.
- Logincheck.php: It verifies user credentials for authentication and creates authentication token M4B_Token.
- Raindex.php: It manages all the pages, to which registered token (user) is authorized to view/modify. All authorized pages to specific token are managed by ACL (Access Control List).

3.4.3. User Roles Management

Mind4b manages association of available features and resources to user with Access Control List. Every feature in this Mind4B is provided to a client through the exhibition of a series of resources. Each resource is characterized by a URI (Location of PHP file).

Each role in Mind4B has permission to use a set of features and / or resources. Through the user interface you can configure the association Role / Resources; the association is

done by a set of preconfigured resources. The management is through a common set of classes that belong to the package ACLs.

Package ACL

The main class package ACL, **M4B_ACL**, which exposes two types of ACL implemented inside of the application:

- **ACL by Resource:** Manages access to a single resource. And 'the atomic element on the management of the authorization. Through the UI can bind the single resource to user roles.
- **ACL by Group Name:** In order to make management more flexible authorization - do list all possible resources in Mind4B - has introduced the concept of Group. A role is associated with an ACL group and all the resources belonging to the group in question have visibility on the role.

The management of the ACL occurs in relation to current context execution of this resource. The context is determined by an entity referred to as tokens (the token is built during the authentication phase). The class provides only the functionality M4B_ACL verification given the context initial authorization by ACL or ACL By Resource Group and responsibility of individual resources to verify the ACL. ACLs are not applied automatically, to be used by the various resources explicitly. In that regard has been implemented a global utility function in the file inc.php/authentication.inc.php:

```
function authorization( $auth , $none_AC = false, $ACL_by_group=false, $ACL_group=null)
```

Analysis using ACL

The use of ACL by single resources is very simple; just call the function authorization by passing parameters on the type you want to use ACLs:

- $auth: M4B_Auth instance, is not a mandatory field in the absence retrieves the instance connected to the current context
- $none_ACL: If set to true disables the ACL for the resource in question

- ACL_by_group: if true indicates a willingness to use the Group By ACL Management
- $ACL_group: indicates the group to which you want to check the authorization

By default the function performs verification by Resource ACL. If the authorization request is successful the function returns an instance of type M4B_ACL.

Structure ACL DBMS

Figure 3.4-4 reflects the structure of the DBMS offers two types of ACLs.

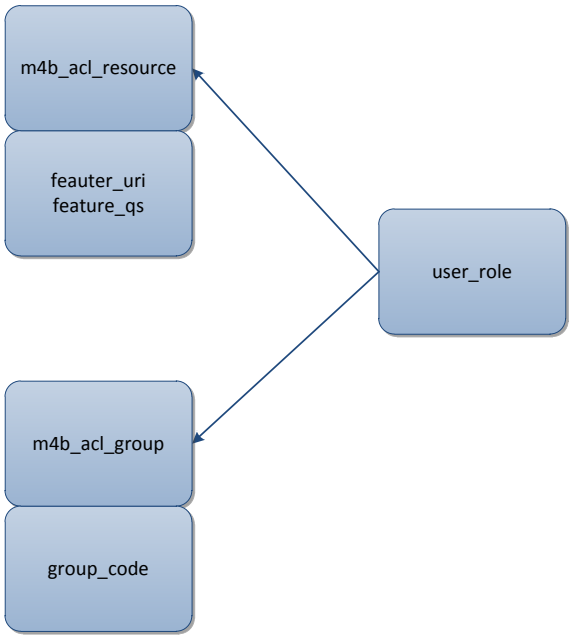

Figure 3.4-4: DBMS Structure of ACL's Implementation

The authorization management of ACL by Resource is through the table m4b_acl_resource while ACL by Group management is through the table m4b_acl_group.

Management ACLs by Group is simple, it is necessary census of all the groups that want to manage and provide the association with the various roles: the association is performed through the UI. The insertion of new element represents the group needs to be done manually.

Managing ACLs by Resource uses two pieces of information to characterize the URIs associated with a resource.

- URL = URI+ ? + Query String

URL where the item is the resource in question, the breakdown is inserted into the tuple:

- URI -> feature_uri
- Query String -> feature_qs

The tricky part because it still managed automatic inclusion of all resources associated with different roles in these Mind4B. Having surveyed the resources you can use the UI for the association.

Menu Manager UI

The management of the UI menu is linked to ACL as it uses the authorization associated with the current role to filter menu items. Also the management of the menu is used to implement the management of the association through the UI Roles / Resources.

The management menu Mind4B occurs in two different places in the code: the library and in folder M4B\inc every single instance. The reason for the division into two different areas of the library: Mind4B all instances share the library and the library M4B have entered all the common classes, the specialization of the menu is placed in individual instances.

To simplify the exposition we show the two lines of code needed for the construction and contextualization of the menu Mind4B:

```
$acl = authorization( $auth, isset ($none_ACL), $ACL_by_group, $ACL_group );
$ctx = new UIContext ();
$root_m = $ctx->create_context_menu($acl, true); // create menu
```

- You create the object M4B_ACL by applying an appropriate strategy of ACL

- It creates the object UIContex, local object instance
- You create the context menu using the object $acl

UIContext class instance is local and contains the customization menu M4B_UIContext extends class (class belonging to the frame work M4B) and must implement a single function:

make_context_menu: method that builds custom menu

The method does is create the tree on the menu using the classes provided by the frame work. As a general rule:

- For each item added **M4B_MenuItem** menu, insert a record in the table m4b_acl_feature with URI and QS obtained the $uri passed to the constructor.
- For each item added **M4B_MenuItem** menu, insert a record in the table m4b_acl_group enhancing the group_code with the element $ACLgroup passed to the constructor.

In a future version you can run the construction method **make_context_menu** with its management of the database through an automated tool.

3.5. Mind4b – Deployment

As explained in section 3.1, Mind4b is flexible solution to stream processes in SCM environment. So, obviously there is possibility that different customers/stakeholders have their own custom requirements in web application and database model. There are three instances of Mind4b currently deployed based on Service Oriented Architecture (SOA) sharing common database model as shown in Figure 3.5-1, which provides a way for consumers of services, as web application, to be aware of available SOA-based services.

Key Facts: In described situation above, system administrator plays a key role to facilitate new consumer/company, ensuring availability of services, Roles/Resources

association/modification to each consumer/company, and managing database schema (as it is shared among all instances).

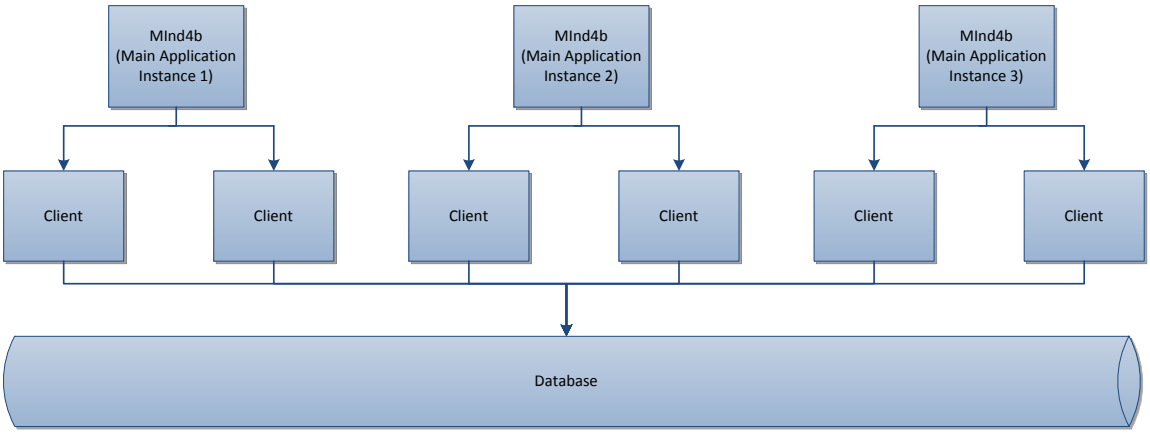

Figure 3.5-1: Currently Deployed Mind4b Instances Overview

3.6. Evolution Phase Challenges

Up to now, we got familiarity with currently implemented and deployed architecture which is functional and meeting sufficient requirements. Currently, Mind4b solution supports HTTP and FTP communication protocols with security model as described in above sections. The evolution phase deals enhancing web architecture which would be centralized point of contact between business partner and business organization to support more communication interfaces, furthermore, centralized security model which would be strong enough to handle predicted security relates threats and challenges. Initial issues while expanding business domain are:

- Reconstruction of Web Architecture having capability of scalability, reliability and increased performance than before
- Separation of Security Model from rest of business logic
- Customization of User ACL Roles Management which may or may not be specific for company
- Customization and integration of Database Model which may or may not be specific for company

Database Integration Possibilities:

- Allocating independent space with separate database space
- Allocating independent space with shared database space

Allocating independent space with separate database space:

In case of distributing space of owner companies, Independent application interface with fully dedicated database schema fulfilling their own requirements will be required:

- Upper layer of the architecture will be managed by Mind4b application interface which will serve as for authentication purposes
- In this scenario, every owner company has possibility to ask Mind4b to accommodate application interface/database schema according to their needs.

Apart from company, Development team has to put more effort in order to maintain different instances of application and database schema as well.

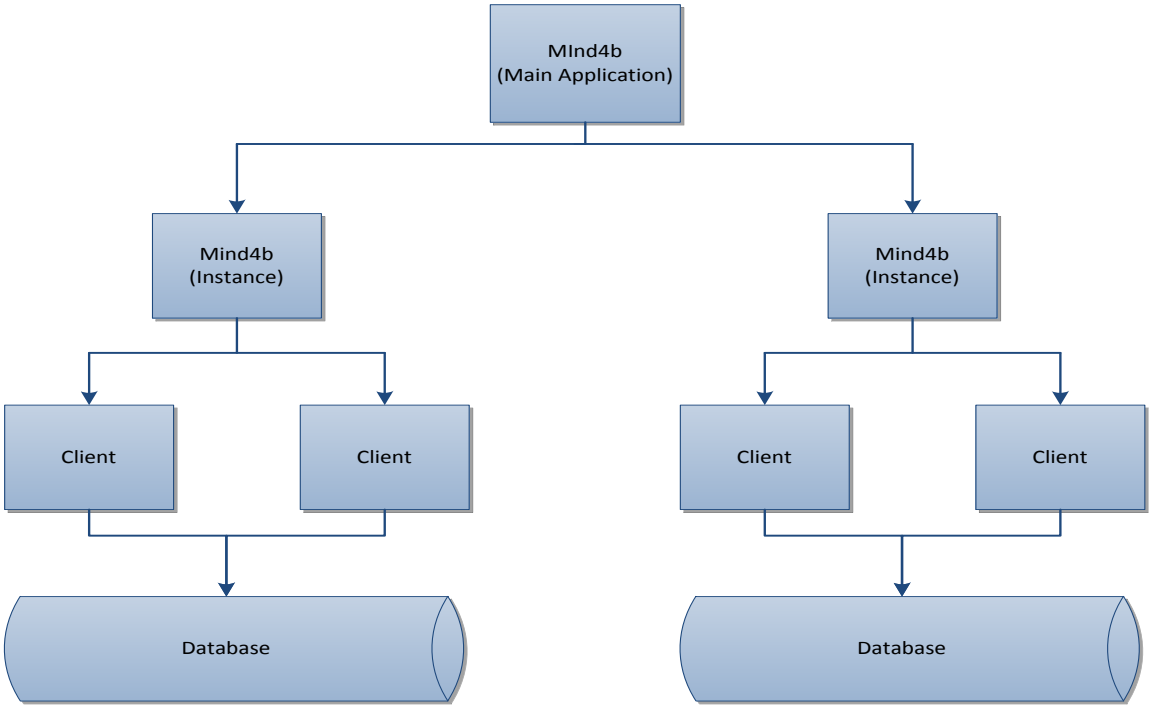

Figure 3.6-1: Scenario 1, Mind4b Instances with Separate Database

Allocating independent space with shared database space:

On the other hand, another possibility to have single instance of Mind4b application with common shared database schema:

- This approach requires to have complete understanding of companies requirements, with some flexibility measures
- So, if in future any of the company wants to have new Role or Resource, then application should have capability of integration

From maintenance point side, single instance of application and common database seems to be more effective. But strong Database integrity will be required which ensure that only concerned and authenticated user is accessing data.

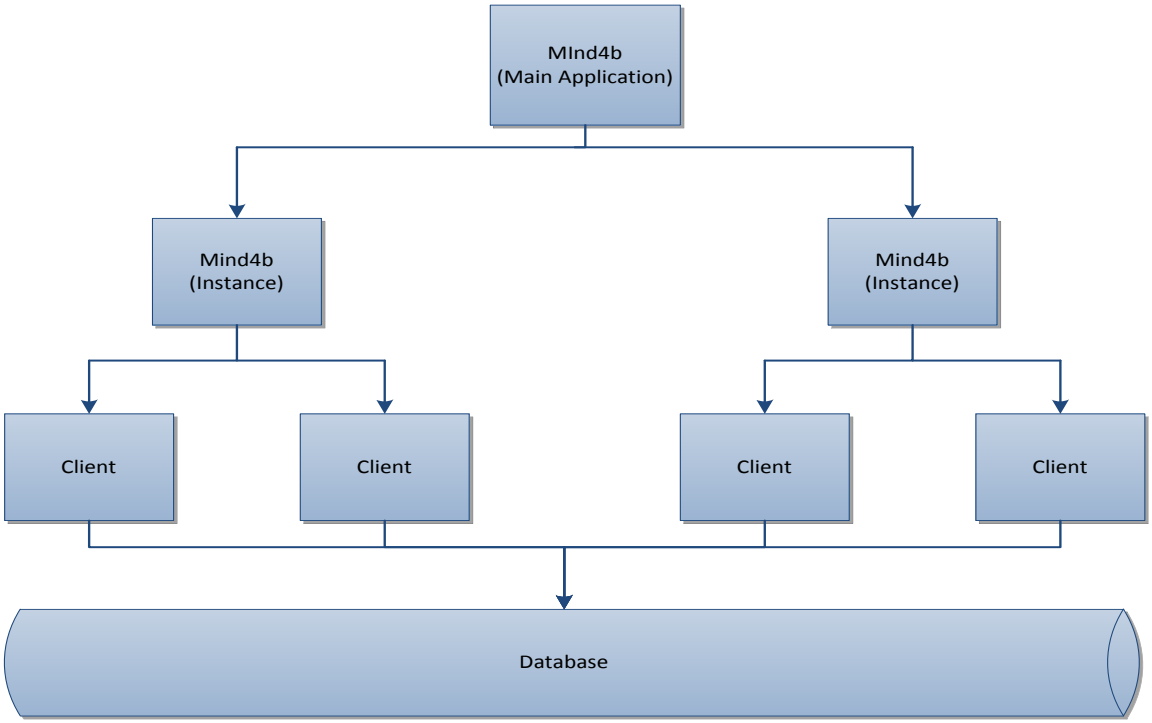

Figure 3.6-2: Scenario 2, Mind4b Instances with Shared Database

4. Security Features in Mule Enterprise Service Bus (ESB)

4.1. Introduction

"As Service-Oriented Architecture (SOA) technology matures and as more organizations include SOA in their system development efforts, they tend to employ Enterprise Service Bus (ESB) applications to facilitate SOA solutions. An ESB is part of what is collectively known as the Service-Oriented Architecture Implementation Framework, or SOAIF, which includes the infrastructure that an organization needs to implement a SOA system. An ESB provides the messaging core for the SOA demanded reliable connectivity between distributed systems and business services.

One of the major advantages of ESB applications is that they easily enable multi-language development in an organization. As such, the services deployed on and ESB do not have to be written in the same language, but can be written in the language that developers are most comfortable with (11)."

4.2. Mule ESB and Security Considerations

"Mule ESB provides a messaging framework that enables exchange of data among applications. The application functionality is wrapped as a service, which includes a service component (the business logic that processes the data), routers (which use endpoints to specify where to send the message), and other configuration settings. Transports carry the messages on different channels from service to service, and transformers convert the messages and data as needed along the way.

Mule is not a replacement for existing application frameworks. Instead, Mule leverages many open source projects such as Apache CXF, Spring, and ActiveMQ and fills a void in enterprise Java development where an application requires complex interactions with a variety of systems on a variety of platforms. Mule makes light work of wiring systems

together in a robust, decoupled environment with little to no code and provides the necessary support to route, transport, and transform data to and from these systems (12)."

Mule ESB can be configured with any one or more security providers' framework.

- Spring Security
- JAAS
- SS4TLS
- PGP

Configuring Security: Mule ESB allows you to authenticate requests via endpoints using transport-specific or generic authentication methods. It also allows you to control method-level authorization on your service components. The Security Manager is responsible for authenticating requests based on one or more security providers. All security is pluggable via the Mule security API, so you can easily plug in custom implementations.

The coming section elaborates Acegi Security (Spring Security project http://www.springsource.org/), Authentication/Authorization mechanisms and providers, Key features.

4.3. Acegi Security

Acegi Security provides comprehensive security services for J2EE-based enterprise software. The particular emphasis is on supporting projects built using The Spring Framework. There are two major operations in Acegi Security; Authentication & Authorization.

It supports Authentication with these technologies:

- HTTP BASIC authentication headers (an IEFT RFC-based standard)
- HTTP Digest authentication headers (an IEFT RFC-based standard)

- HTTP X.509 client certificate exchange (an IEFT RFC-based standard)
- LDAP (a very common approach to cross-platform authentication needs, especially in large environments)
- Form-based authentication (for simple user interface needs)
- Computer Associates Siteminder
- JA-SIG Central Authentication Service (otherwise known as CAS, which is a popular open
- Source single sign on system)
- Transparent authentication context propagation for Remote Method Invocation (RMI) and HttpInvoker (a Spring remoting protocol)
- Automatic "remember-me" authentication (so you can tick a box to avoid re-authentication for a predetermined period of time)
- Anonymous authentication (allowing every call to automatically assume a particular security identity)
- Run-as authentication (which is useful if one call should proceed with a different security identity)
- Java Authentication and Authorization Service (JAAS)
- Container integration with JBoss, Jetty, Resin and Tomcat (so you can still use Container Manager Authentication if desired)
- Your own authentication systems

It supports Authorization in three main areas:

- Authorizing web requests
- Authorizing methods can be invoked, and
- Authorizing access to individual domain object instances

4.3.1. Authentication/Authorization Mechanism

The security framework under discussion follows typical web application's user authentication mechanism which utilizes its own distinct process handling classes. The key participants are:

ExceptionTranslationFilter: It is an Acegi Security filter which is responsible for detecting any exceptions that are thrown. Such exceptions are generally thrown by *AbstractSecurityInterceptor* (Authorization Service Provider). *ExceptionTranslationFilter* either returns HTTP error code 403 (or redirect user to specific page) or launch an *AuthenticationEntryPoint* (which means user is authenticated).

AuthenticationEntryPoint: After successful user authentication, respective business logic will be executed as per web application structure. At this point, server collects user credentials (referred as authentication mechanism) by using protocol filters shown in Table 1 and "Authentication Request" object is built and then presented to *AuthenticationProvider*.

AuthenticationProvider: It is responsible for processing authentication request object and decides whether or not it is valid. The provider will either throw exception or return fully populated authentication object. After the authentication mechanism receives fully populated authentication object by responding valid request, put the authentication object into *SecurityContextHolder* and cause the original request to be retired. The *SecurityContextHolder* is responsible to forward authentication object to *AbstractSecurityInterceptor* for request authorization.

Acegi supports the following Authentication Providers:

- DAO Authentication Provider
- Java Authentication and Authorization Service (JAAS) Provider
- Siteminder Authentication Mechanism
- Run-As Authentication Replacement
- Form Authentication Mechanism

- BASIC Authentication Mechanism
- Digest Authentication
- Anonymous Authentication
- Remember-Me Authentication
- X509 Authentication
- LDAP Authentication
- CAS Authentication
- Container Adapter Authentication

Commonly Used Filters	
HttpSessionContextIntegrationFilter	Keeps the contents of the *SecurityContext* between HTTP requests
AuthenticationProcessingFilter	Form based authentications (JSP for ex)
BasicProcessingFilter	BASIC HTTP header-based authentication (WebServices)
RememberMeProcessingFilter	Cookie that enables remember-me services
AnonymousProcessingFilter	Allows anonymous access
FilterSecurityInterceptor	Protects web URIs

Table 1: Acegi Security Authentication Filters

4.3.2. Channel Security

In addition to coordinating the authentication and authorization requirements of your application, Acegi Security is also able to ensure unauthenticated web requests have certain properties. These properties may include being of a particular transport type, having a particular *HttpSession* attribute set and so on. The most common requirement is for your web requests to be received using a particular transport protocol, such as HTTPS.

An important issue in considering transport security is that of session hijacking. Your web container manages a *HttpSession* by reference to a *jsessionid* that is sent to user agents either via a cookie or URL rewriting. If the *jsessionid* is ever sent over HTTP, there is a possibility that session identifier can be intercepted and used to impersonate the

user after they complete the authentication process. This is because most web containers maintain the same session identifier for a given user, even after they switch from HTTP to HTTPS pages.

If session hijacking is considered too significant a risk for your particular application, the only option is to use HTTPS for every request. This means the *jsessionid* is never sent across an insecure channel.

4.3.3. Future Work

Spring Security will offer

- Considerably simplified configuration
- Windows NTLM authentication
- A user management API
- Persistence-backed remember-me services
- Hierarchical roles
- Spring LDAP Template support
- Considerable ACL enhancements
- Portlet support and much more.

4.4. ESB References

- http://www.mulesoft.org/documentation/display/MULE2USER/Configuring+Security
- http://www.acegisecurity.org/
- http://www.springsource.org/
- http://www.springsource.com/
- http://en.wikipedia.org/wiki/Acegi_security_framework_(Java)
- http://www.thespringexperience.com/show_session_view.jsp?presentationId=9249&showId=147
- http://www.javalobby.org/articles/acegisecurity/part1.jsp

4.5. ESB Free Online Resources

Mule Community provides free (product demos, technical webinars and whitepapers) to the attention of developers, architects and IT decision makers working on integration and SOA projects:

- Documentation:

  http://www.mulesoft.com/downloads/mule-2.2.0-getting-started.pdf

- Videos:

  http://www.mulesoft.com/demo-getting-started-mule

  http://www.mulesoft.com/demo-building-simple-service-mule

- Online training:

  http://www.mulesoft.com/mule-training

5. Proposed Security Model for Mind4b

5.1. Overview

This chapter presents the proposed security model implemented on improved Mind4b architectural model and its operational structure. This model has been designed to solve the predicting security problems under the domain of Enterprise Application Integration by using Enterprise Service Bus framework. This protocol works in an efficient manner and solves the major problems by using the existing resources to allow for lower computational and storage overheads. The proposed solution is aimed to develop, enhance and strengthen the security architecture of the Enterprise Business Applications.

5.2. Proposed Solution

After analyzing currently deployed security mechanisms in Mind4b web information system and getting understanding of workflow of ESB architecture, the proposed solution is to first separate security model from rest of the business logic. By creating a separate web service containing all security mechanisms will behave as front end layer with which user has to interact with. So, in this way this web service will provide security shield to all other services of enterprise web application.

The coming section will describe network architecture and working of new web application architecture model.

5.3. Network Infrastructure

The network infrastructure consists of two server machines with Debian Linux operating system as shown in **Error! Reference source not found.**. The specifications are as under.

The first machine named as App1 is running with:

- An Apache Web Server with PHP module to initiate Mind4b web application instance.
- The Tomcat Servlet container handles Web Services and Mule business application logic.

The second machine named as App2 is running with:

- One instance of database (PostgreSQL) for managing database related transactions.

Both machines are located within private network shielded by firewall, and are accessible via SSH and SCP for file transfer. **Error! Reference source not found.** shows architecture configuration.

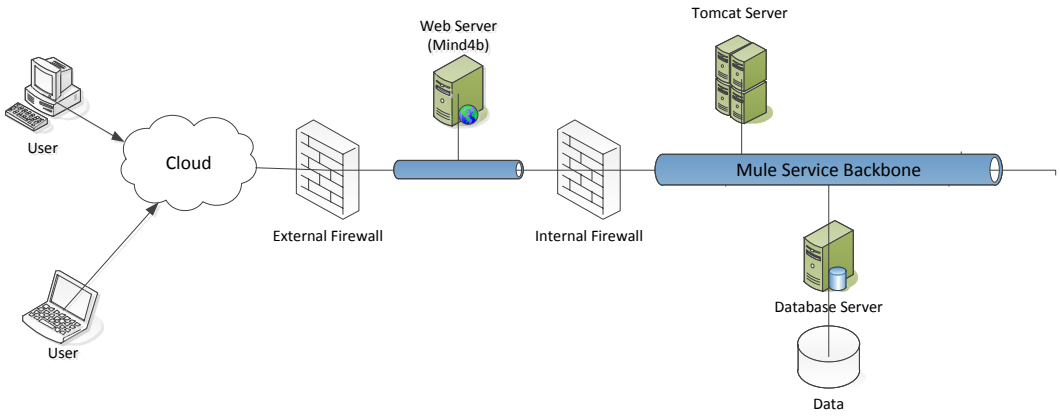

Figure 5.3-1: Integration of Mule ESB with Mind4b

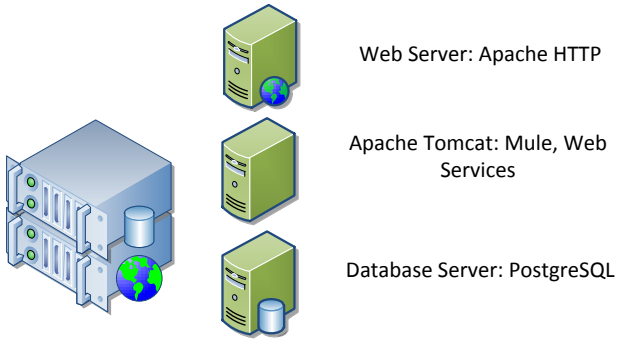

Figure 5.3-2: Improved Architectural Model

5.4. New Web Architecture

As explained ESB framework working structure in previous chapter; User Authentication/Authorization mechanism is encapsulated in dedicated web service along with other web services. All user requests will be handled by main web server contains web application interface of Mind4b implemented using PHP. Then user requests will be transformed to ESB message format and forwarded to ESB bus where different service components are implemented. On receiving messages for User Authentication/Authorization service component, the message will be forwarded to respective Authentication/Authorization web service which actually runs the business logic.

Error! Reference source not found. shows the above explained working scenario.

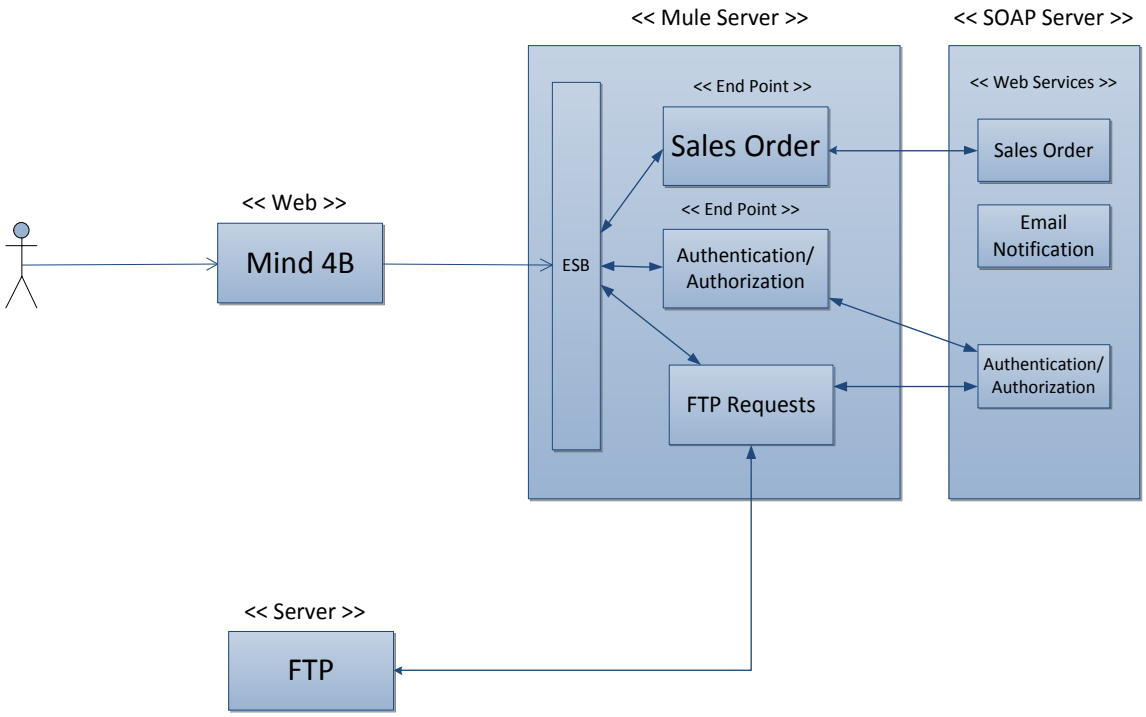

Figure 5.4-1: Improved Web Architecture

5.4.1. Authentication/Authorization Web Service structure

The business logic of authentication/authorization web service consists of following four functions:

- **Authentication:** User Authentication is linked with function login which receives userId and password and returns either m4bToken (if success) or exception (if failure).

  login (userId, password) → token | exception

- **User Access Control Lists:** After successful user authentication, next task is to generate user ACLs as explained in Section 3.4.3 by using function buildACL. In case of retaining user, authentication will be checked through already generated token (if valid) but authentication failure will return exception and redirect user to login page.

  buildACL (m4bToken) → m4bACLList | exception

- **Token Verification:** Before every user activity on the web application, there is a need to verify user m4bToken with associated user's ACLs and then generate new m4bToken or return exception. The function chenckToken is:

  checkToken (m4bToken) → new m4bToken | exception

- **Authorization:** User will be authorized by verifying already generated ACLs. Every user can only access those features for which he/she is authorized. The function checkAuthorization is:

  checkAuthorization (m4bToken, url)

Mind4b User Token Structure:

The Mind4b token contains the following user related information attributes.

- userID – contains User ID
- dbUserID – refers database where user information is stored

- userRole – related with user role group
- born – time on which token is created
- ttl – defines token validity time

5.5. User Authentication/Authorization Process

Whenever any user will interact with Mind4b web application, the generated request (SOAP Message) will be forwarded to Mule Server where user request will be processed by Authentication/Authorization Service Component. Then service component will transfer user credentials to Authentication/Authorization Web service for processing. The web service will connect PostgreSQL database server to check user credentials. In case of success, web service will create M4b_Token & buildACL and will send this token back to Mule server and then ultimately to user where this token will be settled as session cookie. In case of failure, user will be notified with 'Login Failure' message and will redirect to login page. User Authentication sequence flow is shown in **Error! Reference source not found.**.

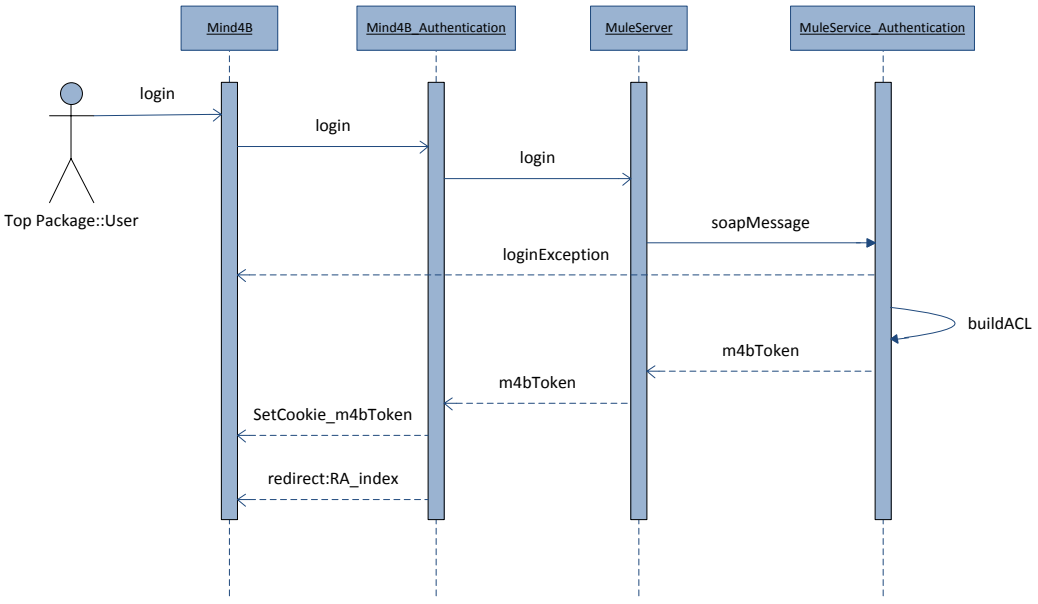

Figure 5.5-1: Authentication Sequence Diagram

After successful authentication, every next user activity will be authorized by m4bToken containing user related information. The Authentication/Authorization web service will verify m4bToken and will issue new m4bToken with updated ttl time as explained in above section. Authorization failure will redirect user to login page. **Error! Reference source not found.** is Authorization sequence diagram represents the working flow.

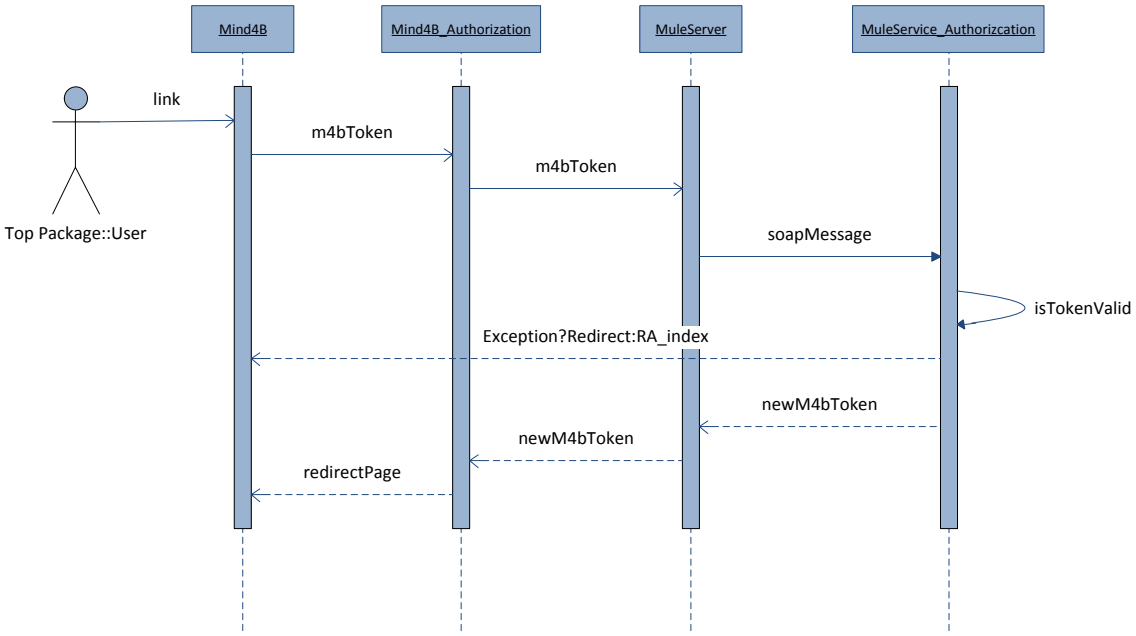

Figure 5.5-2: Authorization Sequence Diagram

6. Conclusions

6.1. Summary

The project presented above supposed to have knowledge of business web applications and their architecture, application integration patterns, and typical security implementations, mechanisms and protocols. Later, there is an overview of web information systems, enterprise application integration, identification and analysis of security requirements and threats during integration phases. Moreover, there is an overview of Enterprise Service Bus framework, corresponding features, support to web information systems, and security support.

The most important element of exertion is composed of understanding, identification, and analysis of currently deployed, Mind4b – Collaborative Supply Chain Management, enterprise web application. During the evolutionary phase of Mind4b, the requirements of efficient communication protocol among business partners, improvement in security mechanisms/protocols while user authentication/authorization to increase customer trust level and preventing web information system from being compromised and vulnerability.

6.2. Future Work

The major outcome of the thesis work is to integrate proposed Mule ESB framework as described in (3) for Mind4b which is being focused on enterprise service bus. In addition to separate all security related functions from the rest of the business logic which will strengthen investigation process in case of any unexpected activity during the whole process.

As the proposed architecture with basic security feature has been implemented and functioning perfectly with same database system. The next step would be to increase security mechanism by increasing user authentication/authentication procedures and

management of user access control lists. HTTP being stateless protocol, most commonly used interface between client and server, consist of many known problems can be converted to state-full protocol. Session management problems including cookie parameters issues are well known which may cause the first point of session-hijacking. The issues related to data integrity and confidentiality is still under discussion because of possibility of customization of database schema related to specific customer. Furthermore, Integration issues while integrating different database models, transaction management among database server, load balancing issues.

Bibliography

1. *Service Oriented Architectures: Approaches, Technologies and Research Issues.* **Papazoglou, Mike P. and Heuvel, Willem-Jan van den.** s.l. : The VLDB Journal, 2007. DOI 10.1007/s00778-007-0044-3.

2. *Analyzing Websites for User-Visible Security Design Flaws.* **Falk, Laura, Prakash, Atul and Borders, Kevin.** Pittsburgh, PA USA : s.n., 2008. Symposium on Usable Privacy and Security (SOUPS).

3. **Bugna, Alberto.** *Un Approccio a Servizi all'Integrazione, per l'Integrazione della Filiera Logistica in Full Outsourcing.* Milano, Italia : Politecnico di Milano, 2009/2010.

4. **Oracle.** Point-to-Point Messaging. *Oracle Documentation.* [Online] http://download.oracle.com/docs/cd/E19798-01/821-1798/aerbj/index.html.

5. **M. Allman, S. Ostermann.** RFC 2577: File Transfer Protocol. *In The Internet Engineering Task Force.* [Online] May 1999. http://www.ietf.org/rfc/rfc2577.txt.

6. **Fielding, Roy T., et al., et al.** RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. *In The Internet Engineering Task Force.* [Online] June 1999. http://tools.ietf.org/html/rfc2616.

7. **Franks, J.** RFC-2617: HTTP Authentication: Basic and Digest. *In The Internet Engineering Task Force.* [Online] 1999.

8. *Analysis of a Denial of Service Attack on TCP in Proceedings of the 1997 IEEE Symposium on Security and Privacy.* **Schuba, C.L.** s.l. : IEEE Computer Society Press, 1997.

9. *Distributed Denial of Service Attacks. in IEEE International.* **Lau, F.** Nashville, TN, USA : s.n., 2000.

10. *An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks.* **Paxson, V.** 3, s.l. : ACM Computer Communications Review, Vol. 31.

11. *A Multi-Language Service-Oriented Architecture Using an Enterprise Service Bus.* **Sward, Ricky E. and Whitacre, Kelly J.** Portland, Oregon, USA : ACM, 2008.

12. **MuleESB.** Configuring Security. *Mule ESB Commnity.* [Online] http://www.mulesoft.org/.

13. **Oppliger, R.** Chapter 4. *Cryptographic Techniques, in Internet and Intranet Security.* Boston : Artech House, 1998, p. 348.