TECHNICAL ARCHITECTURE DESIGN DOCUMENT

Version 1.0 ● OCTOBER, 2016

**TECHNICAL ARCHITECTURE DESIGN DOCUMENT**

**[XXXX]**

|  |  |
| --- | --- |
| VERSION: 0.1 | REVISION DATE: 31/10/2016 |

Approval of the Technical Architecture indicates an understanding of the purpose and content described in this deliverable. By signing this deliverable, each individual agrees with the content contained in this deliverable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Approver Name** | **Title** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Section 1 DOCUMENT SCOPE 3](#_Toc371321504)

[Section 2 OVERALL TECHNICAL ARCHITECTURE 3](#_Toc371321505)

[2.1 Overall System Architecture Diagram 3](#_Toc371321506)

[2.2 System Architecture Model 3](#_Toc371321507)

[2.2.1 Overall Architectural Considerations 4](#_Toc371321508)

[2.3 System Architecture Component Definitions 15](#_Toc371321509)

[2.3.1 System Architecture Component– Database Server 15](#_Toc371321510)

[*2.3.2* System Architecture Component – Application Server 16](#_Toc371321511)

[2.3.3 System Architecture Component – SOA Server 16](#_Toc371321512)

[2.3.4 System Architecture Component – LDAP Server 16](#_Toc371321513)

[2.3.5 System Architecture Component – Web Server 16](#_Toc371321514)

[Section 3 System Construction Environment 17](#_Toc371321515)

[3.1 Development Environment 17](#_Toc371321516)

[*3.1.1* Developer Workstation Configuration 17](#_Toc371321517)

[*3.2* Code Quality Environment 17](#_Toc371321518)

[*3.2.1* QA Workstation Configuration 17](#_Toc371321519)

# Section 1 DOCUMENT SCOPE

This document describes the Technical Architecture of the xxxx System that satisfies business requirements as documented in the sub-system specific SRS, and implements the functionality and satisfies technical, operational requirements described in the Functional Specification..

The goal of this Technical Architecture is to define the technologies, products, and techniques necessary to develop and support the system, and to ensure that the system components are compatible and comply with the enterprise-wide standards.

This document will also:

* Identify and explain the risks inherent in this Technical Architecture;
* Define baseline sizing, archiving and performance requirements;
* Identify the hardware and software specifications for the Development, Testing, QA and Production environments;
* Define procedures for both data and code migration among the environments.

# Section 2 OVERALL TECHNICAL ARCHITECTURE

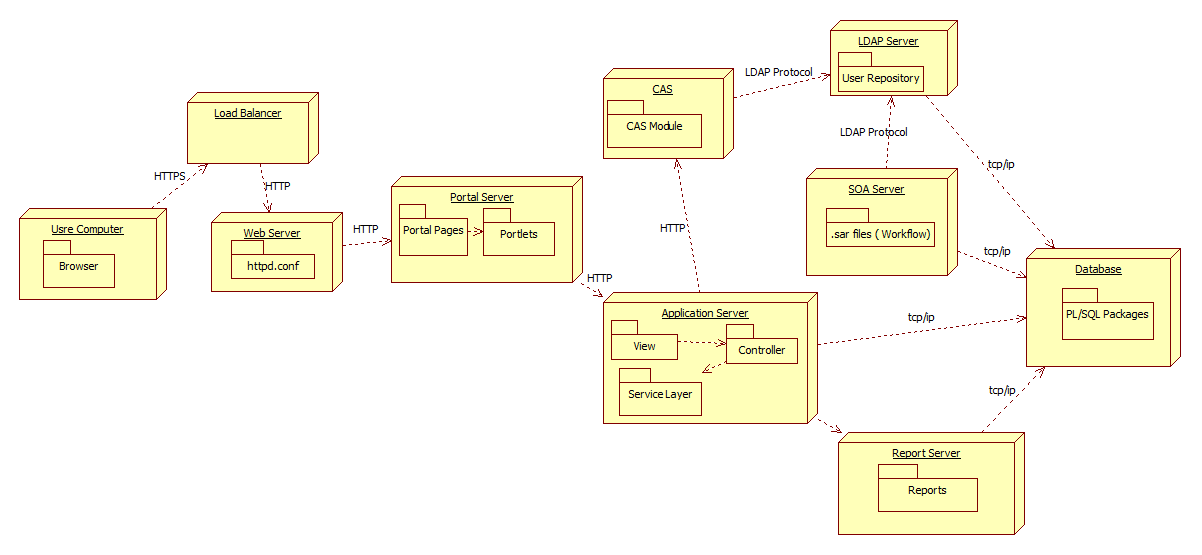
## 2.1 Overall System Architecture Diagram

The **System Architecture Diagram** provides the “big picture” view of the system’s architecture, and puts it in context with the rest of the Performing Organization’s systems portfolio, illustrating how the system’s hardware and software platforms fit into the environment.

## System Architecture Model

The **System Architecture Model** represents the various software components that comprise the system, and shows their interrelationships.

|  |  |  |
| --- | --- | --- |
| Sl.No | Software Components | Description |
| 1 | Load Balancer | Balance External traffic and share load against pair of web servers. |
| 2 | Web Server | Web server , intercepts traffic coming form load balancer and distribute against application servers. |
| 3 | Portal Server | Portal server provides runtime for JSR compliant portlets. |
| 4 | Application Server | Application server provides runtime for web applications. |
| 5 | SOA Server | SOA Server provides runtime for Workflow and other integration adaters. |
| 6 | LDAP and CAS | LDAP server stores user credentials in secured manner.CAS application is SSO application provides Granting ticket to all applications and provides necessary SSO Support |
| 7 | Database Server | Database server host data and business logics inside PL/SQL |
| 8 | Reports and BI Server | << Not Yet Decided >> |



Above diagram depicts component wise deployment of iFMS and their relationships. Major components of

User request will be fronted by a pair of active-passive load balancers. Connection to load balancer will be secured and will use VeriSign certified digital certificate. SSL connection will be terminated at load balancer level. From load balancer request will be send to a pair of active-active web servers. Web server will redirect request to portal server. Portal server will host the public page of iFMS odisha and can be accessed by any user. Portal will be the entry point or gateway for iFMS business applications.

Portal war will be deployed in Webcenter Portal environment. WSRP 2.0 portlets will be deployed in Portlet container of Oracle Web Center portal.When user will select any business application request will be redirected to application servers. Individual applications will be deployed in Weblogic application server cluster as .war files. If user is not authenticated then request will be redirected to CAS server and CAS login page will be displayed. Once user will provide userid and password , CAS will validate user authentication against ldap server and will generated granting ticket and redirect user to request application.All workflows and business rules will be deployed in Oracle SOA server as SOA Composite deployment packages i.e .sar files.Application will intern communicate with Database and SOA servers.

### 2.2.1 Overall Architectural Considerations

*The* ***Overall Architectural Considerations*** *section defines how following technical requirements have been addressed by the architecture. Representative items in this section include:*

* *Application Access strategy.*
* *User Authentication Strategy.*
* *Presentation tier strategy*
* *Business layer strategy.*
* *Data access strategy*
* *Transaction Strategy.*
* *Error Handling Strategy.*
* *Workflow Integration Strategy.*
* *Content Management Integration Strategy.*
* *Security Strategy*
* *Inter application communication strategy.*
* *External Interface design and integration strategy.*
* *Code Quality assurance strategy*
* *Deployment Strategy*
* *Data Migration Strategy*
* *Disaster recovery*

#### Application Access strategy

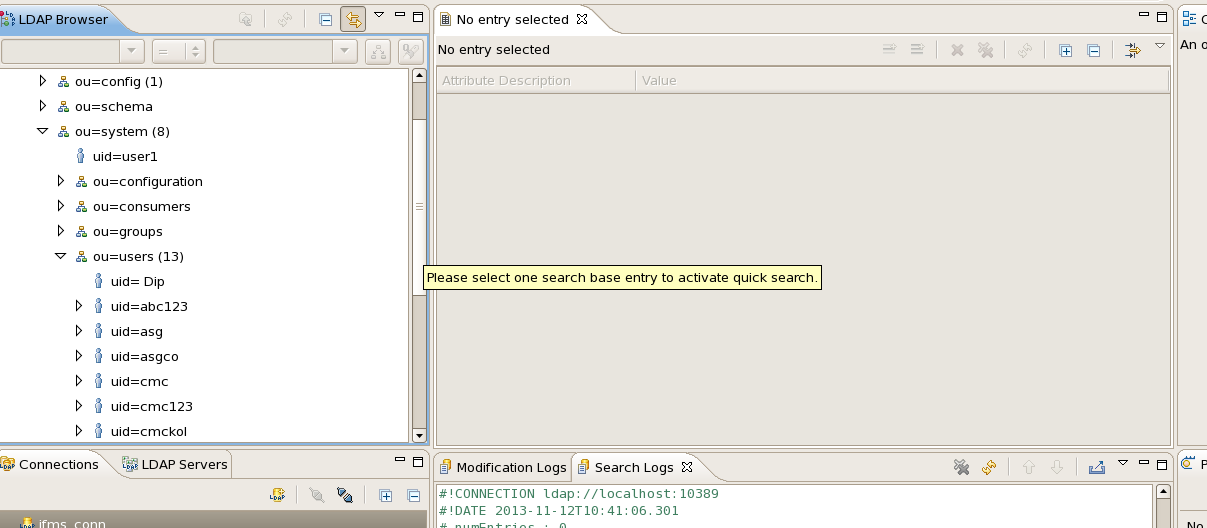
#### iFMS application will be accessed using internet and intranet. IFMS will support following browser versions – IE 7/8, Google Chrome 29/30, Firefox. Major stakeholders of the systems include DDO/CO/Departmental users /treasury users, Citizens of xxx.

#### User Authentication Strategy

User details of iFMS registered users will be stored in Apache DS LDAP server. As user roles will be changed frequently, details of user role will not be stored inside LDAP Existing iOTMS applications stores user roles and credentials in Oracle database, to promote reuse and smooth execution of iOTMS applications, it has been decided – iFMS user roles will be reused using existing RDBMS and user credentials will be migrated to LDAP server. Existing iOTM users will continue to use RDBMS based user credentials and authorization details. But as a next phase migration, user credentials and roles will be moved into LDAP discarding RDBMS based approach. Following user class will be inherited in iFMS users

* Top
* Person
* OrganizationalPerson

Application will adhere to password policy defined by the Technical team of client and described in attached password policy document.



Welcome page of iFMS portal application will be hosted inside Oracle WebCenter portal application server. This will be a public page and will contain link to different business applications. All business applications will be protected using SSO . CAS will be as SSO solution for iFMS application. As deployment will be made in Cluster environment, iFMS will use JPATicketRegistry to store tickets in Oracle RAC database.

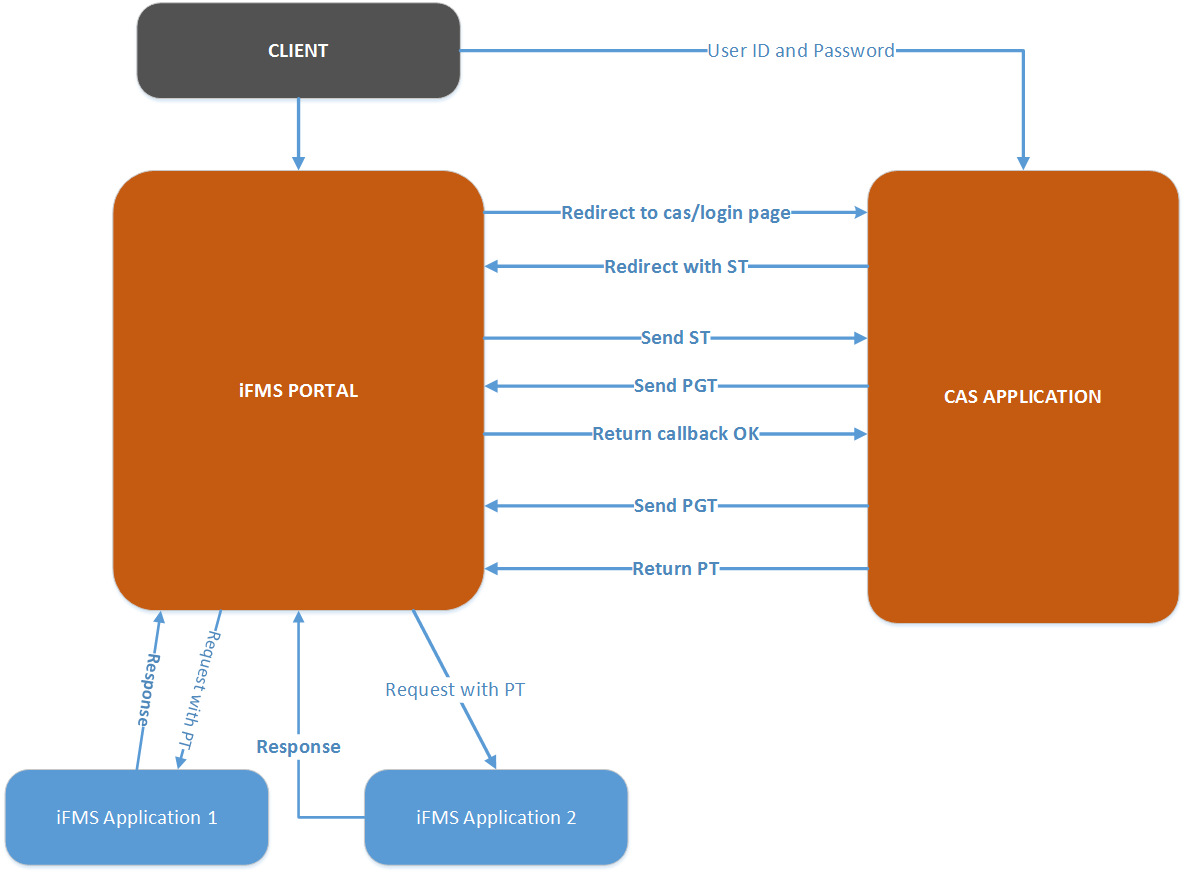


Figure 1 iFMS CAS integation

Integration with CAS Server :-

Followings are the steps to integrate iFMS application with CAS Server.

CAS client uses following filters to connumicate with CAS server.

* CAS Authentication Filter
* CAS Validation Filter

1. Put CASClient.jar inside WEB-INF/lib directory of the application.
2. Put CAS\_SERVER\_LOGIN = <<casserverUrl:Port/cas/login>> in casserver.properties
3. Put APP\_SERVER=<<appserver.host:Port>> in casserver.properties
4. CAS\_SERVER = = <<casserverUrl:Port/cas>> in casserver.properties
5. Add followings in web.xml

<filter>

<filter-name>CAS Authentication Filter</filter-name>

<filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>

<init-param>

<param-name>casServerLoginUrl</param-name>

<param-value>${CAS\_SERVER\_LOGIN}</param-value>

</init-param>

<init-param>

<param-name>serverName</param-name>

<param-value>${APP\_SERVER}</param-value>

</init-param>

</filter>

<filter>

<filter-name>CAS Validation Filter</filter-name>

<filter-class>org.jasig.cas.client.validation.Saml11TicketValidationFilter</filter-class>

<init-param>

<param-name>casServerUrlPrefix</param-name>

<param-value>${CAS\_SERVER}</</param-value>

</init-param>

<init-param>

<param-name>serverName</param-name>

<param-value>${APP\_SERVER}param-value>

</init-param>

</filter>

.

#### Application Architecture

All iFMS applications will be developed using layered architecture. All applications mostly contains at least following tiers

1. Presentation Tier
2. Business Ti
3. Data Access Tire

Additionally, applications require to access SOA or content service or intra application common services, will contain a gateway tire to access these functionalities.

**Presentation Tier**

By choice Spring MVC will be used to develop all the iFMS applications. Spring context will be loaded using DispatcherServlet provided in web.xml

All developer needs to register , ContextLoaderListner as well as Spring DispatcherServlet in web.xml .

Spring configuration xml will use UrlBasedViewResolver to resolve request redirection.

Following needs to be entered inside <<Springconfig-Servlet.xml>>

<bean id="viewResolver"

class="org.springframework.web.servlet.view.UrlBasedViewResolver">

<property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>

<property name="prefix" value="/WEB-INF/jsp/"/>

<property name="suffix" value=".jsp"/>

</bean>

All, jsp must be reside inside WEB-INF folder to restrict direct access of jsp. All controller must be annotated using @Autowire tag.

**Business / Service and Data Access Tier / Transaction Management Strategy**

To promote reuse and existing code , most of the business logic of iFMS resides inside PL/SQL blocks,

iOTM application was developed using sightly non decoupled architecture where , all business

functionalities are residing inside PL/SQL blocks. To reuse those functionalities iFMS has no other choice

to use those PL/SQL api’s . So, business tier is extremely thin layer and only used to call these DAO and

to manage transactions.

Previously written PL/SQLs returns error messages instead of throwing exceptions, changing those

PL/SQL has a huge impact on existing iOTMS which is not under the scope of iFMS application.

Hence, transactions inside iFMS is used using programmatic transaction management instead of

declarative transaction management strategy. As iBATIS is selected data access framework for iFMS ,

SQLMApClients are injected inside Service layers .

Each service starts with a **benginTransaction** Statement

As example

public void SampleTxn( User user,

String taskId) {

sqlClient.startTransaction();

From business service , appropriate DAO method will be called to communicate with database.

DAO’s are injected into Service class using spring injection mechanism.

sampleDAO. SampleTxn (sqlClient,cbean,user).

DAO must not catch any exception so that exceptions can be propagated to Service layers.

Once , response for DAO is available in service class, returned error message and error code of PL/SQL

is checked inside service and based on that transaction is either committed or rollbacked. If transaction is

rollbacked then IFMSException will be raised and populated with appropriate exception message and

passed to the controller,

Controller will catch those exception and put appropriate message in httprequest .



#### Workflow Strategy

Workflow need of the iFMS application, will be taken care by Oracle SOA Workflow engine. Oracle SOA Workflow engine runs on top of Weblogic application servers. Workflows are long running component and it stores its state inside oracle SOA repository created inside Oracle RAC database. JDeveloper SOA extension provides design time view of the workflows. Following points will be taken care during the development of Oracle Workflow.

* As all roles of iFMS are not streamlined from the client end , Wokflows will be user based workflows not role based.
* Escalation, forward, reassign, pushback methods will move workflow from one hierarchy of user to another hierarchy.
* Workflows will validate user from Enterprise LDAP i.e ApacheDS.
* Workflow hierarchy details will be stored in Oracle database.
* Primary key of the Business data will be passed into the workflow to associate workflows with application.

Integration of Workflow with iFMS application.

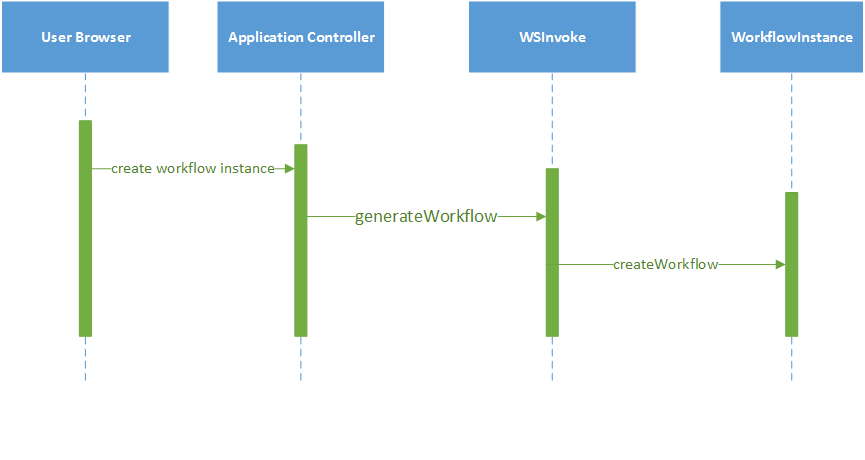
Oracle workflow provides two types of integration hooks for integration with custom applications.

* EJB based integration.
* Webservice based integration.

All iFMS applications are using Spring MVC and Spring Service layers , hence introduction of EJB within spring application will make workflow calls unnecessarily complicated and hence iFMS applications will be integrated with Workflow using Web Services.

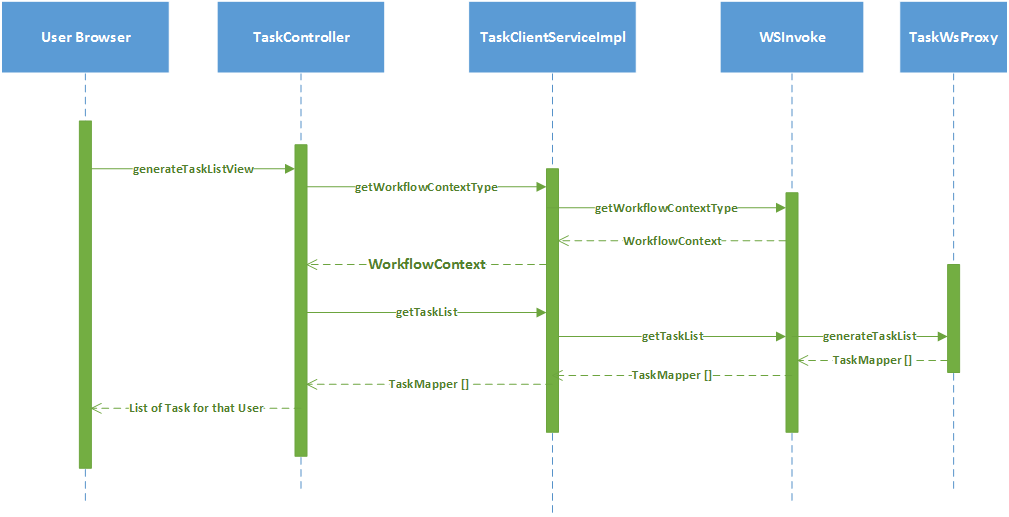
Followings are major Workflow Use cases

1. Create Workflow



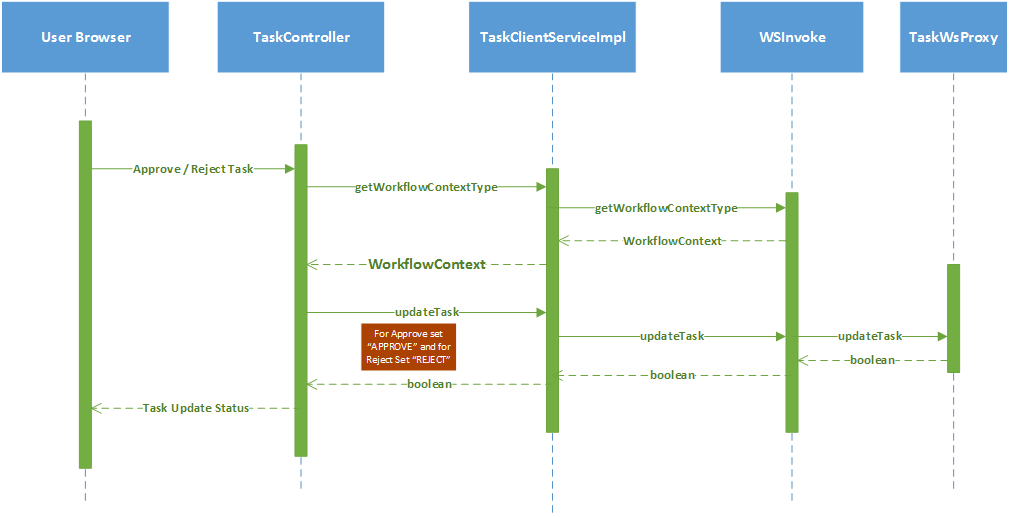
Create Workflow

1. View TaskList



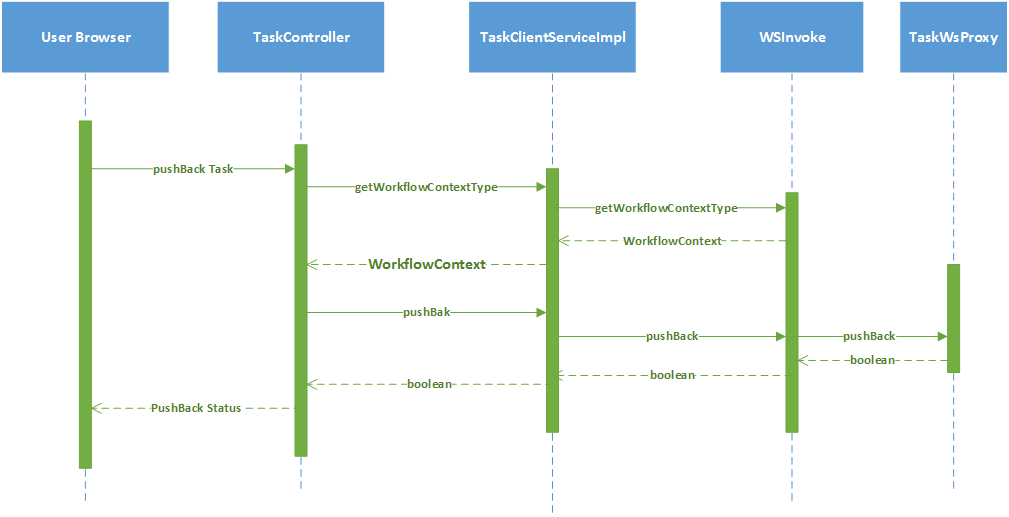
View TaskList

1. Approve/Reject Task



Approve / Reject Task

1. Pushback Task



Pushback Task

**Workflow Exception Handling**

All workflow web services throws WorkflowException Fault , this is caught at WSInvoke layer and Propagated to Controller layer as Service Exception.

#### Inter application (within iFMS ) communication strategy

Almost all iFMS applications are interconnected and hence they need strong data level integration. Integration among iFMS applications are taken care by following strategies.

1. WebService Level Integration.
2. Database level Integration

***WebService Level Integration***

Till this version of architecture design document, CMC has identified a part common service list and repositories. These common services are exposed as web services and hence consumed by all other applications. Following are the list common services and functionalities.

|  |  |  |  |
| --- | --- | --- | --- |
| SL.No | Category | Service Name | Service Description |
| 1 | Workflow | generateWorkflow | This service creates workflow instances |
| 2 | Workflow | getTaskCount | This service provides number of Task for a particular user |
| 3 | Workflow | getTaskList | This service provides list of task for an user. |
| 4 | Workflow | getParticularTask | This service provides details of a particular task based on task ID. |
| 5 | Workflow | updateTask | This service approves / reject tasks. |
| 6 | Workflow | pushbackTask | This service push task one level below |
| 7 | Workflow | getWorkFlowontext | This service provides workflow context for an user. |

***Database Level Integration***

A separate schema is created for each sub module of iFMS and common tables are stored inside a common schema. Applications access the tables residing in different schema using synonyms . Data posting to other modules uses PL/SQL codes which are accessed from application using JDBC connections.

***External Interface design and integration strategy.***

As of now all external applications uses files based integration strategy , as these applications are not completely compatible as SOA Service consumer . Even , if they are not completely SOA enabled, but iFMS consumes all inbound external request using FTP Adapters. External inbound applications sharers files in a ftp server as comma separated files structure , and inbound ftp adapter receives these data and apply appropriate conversion strategy and call outbound DB Adapter to store those data in data base.

This section will be upgraded as an when more external interfaces will be identified.



#### Security Strategy

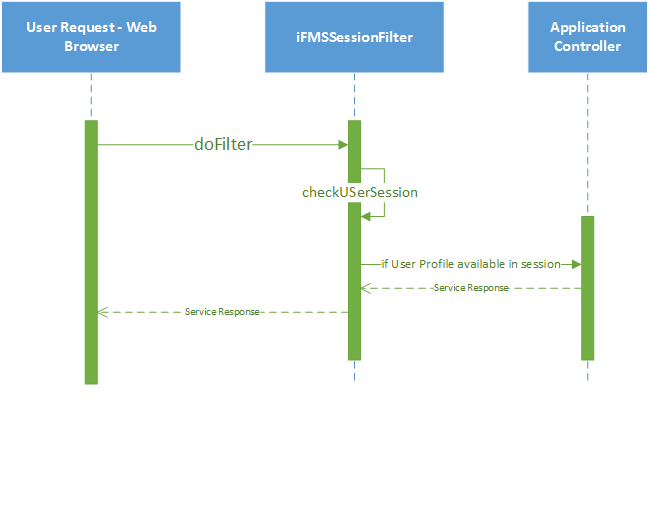
iFMS application will be access be internet and intranet both . Hence, applications need to implement industry standard security best practices to prevent against all possible attacks.

Followings are incorporated in the applications.

* Incepting filter to check user session.
* Proper data validations during data capture.
* Use of bind variables to restrict SQL injections.
* SSL layer security.
* Role based access of functionalities
* Secured password policies

***Intercepting Filter***

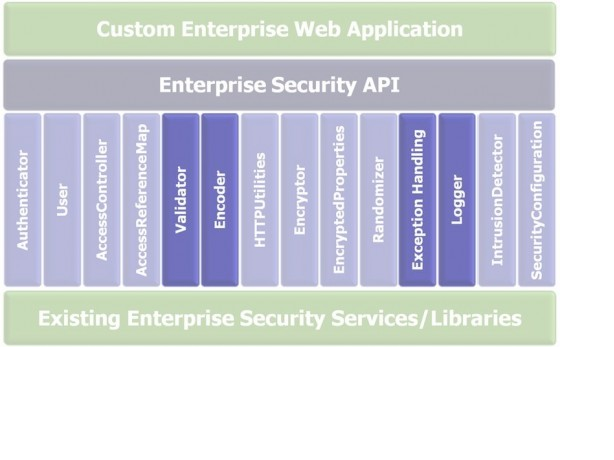
Each url of the application are protected using Session Filter configured in web.xml , this filter will check valid user session and user profile object . If no user profile is found in the session application will redirect to invalid session page.



***Proper data validations during data capture.***

To ensure XSS prevention all user input data must be validated properly before being saved into database

Also before displayed in pages inside HTML tags. All data must be validated in front end level using java scripts also these data must be validated at server level. ESAPI security API will be used to ensure data validation, injection prevention.



***ESAPI data validation.***

Please put esapi.jar available in SVN inside applications Web-INF/lib folder.

Following are inbuilt functionalities available inside ESAPI to validate user input.

* isValidCreditCard
* isValidDataFromBrowser
* isValidDirectoryPath
* isValidFileName
* isValidFileContent
* isValidSafeHTML

From the application controller use **org.owasp.esapi.reference.DefaultValidator** object to validate user input

Please visit

<http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/reference/DefaultValidator.html> for detail API

**ESAPI XSS prevention**

Typically output in most web frameworks leads to XSS and CSRF vulnerabilities. ESAPI encoder allows direct encoding depending on context.

**Use ESAPI.encoder().encodeForHTML**() inside jsp while display saved data .

As example

<p>Hello, <%=name%></p>

<p>Hello,

<%=ESAPI.encoder().encodeForHTML(name)%>

</p>

***Use of bind variables to restrict SQL injections***

Web based application are prone to injection Attacks. To stop all kind of SQLInjection attacks , all SQL statements of iFMS will use bind variables . Use of bind variables will nullify the SQL injection attack.

iFMS is also prone to LDAP injection and thus application will use EASPI encoder to prevent LDAP injections during searching of LDAP data.

ESAPI validator contains encodeForLDAP method to validate user input while communicating with LDAP.

***SSL layer security***

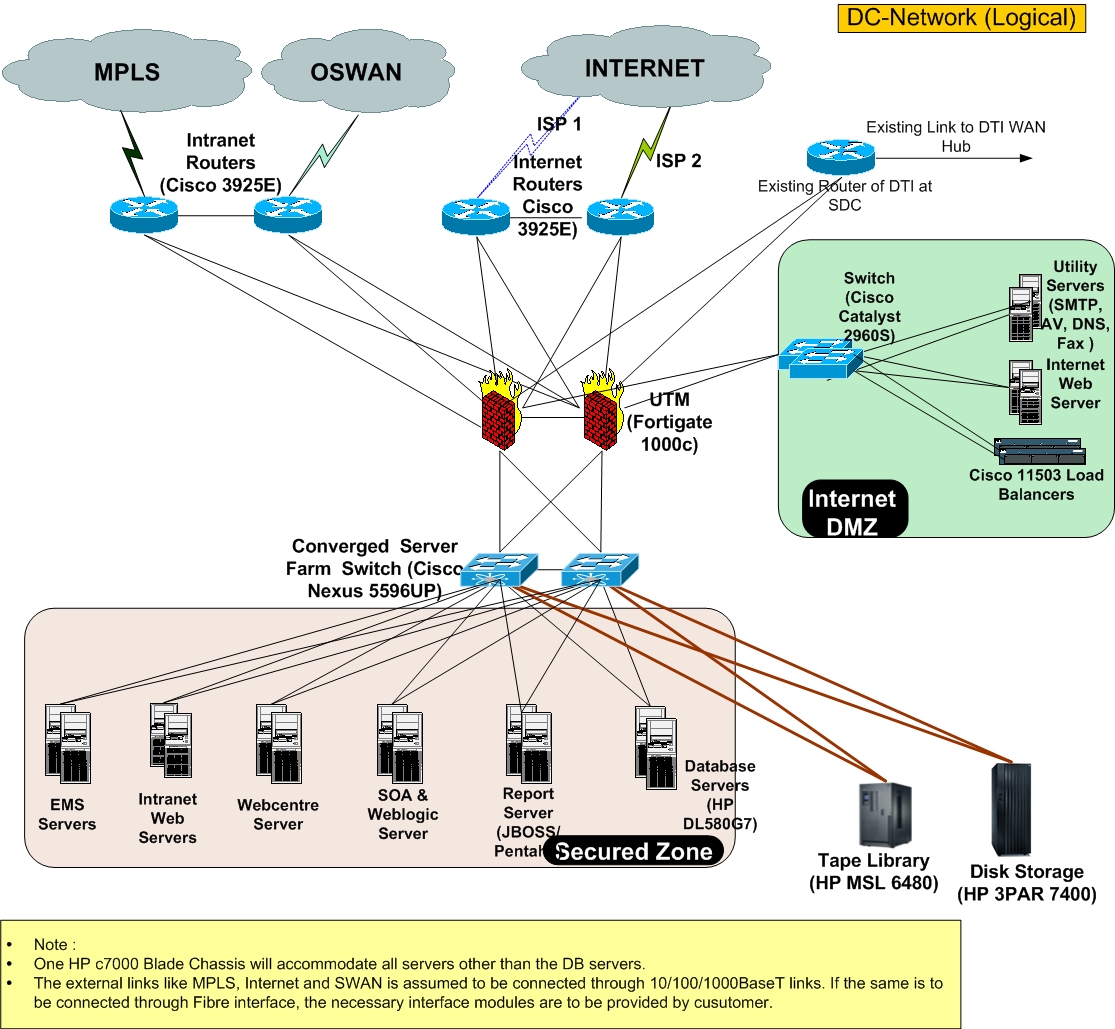
iFMS portal will use secured https transport layer connection . SSL certificate will be purchased form only recognized certificate authority like verisgn etc. Minimum 128 bit SSL is recommended. As all the servers after load balancer resides inside trusted secured zone, SSL will be terminated at load balancer level only. Rest all the connections downstream are non-secured.

***DataMigration Strategy***

Existing database of iOTMS will be migrated into new hardware. As database OS and hardware are diffenet in target and source database, RMAN backup will not work properly. Using Oracle export and import facility data of existing database will be migrated from source to destination database.

Apart from existing database, there exists some disparate applications runs inside Oracle database, these data will be loaded into a staging table and using PL/SQL migration code data will be converted into iFMS required format will be put into iFMS database.

#### Deployment View



## 2.3 System Architecture Component Definitions

### System Architecture Component– Database Server

The **Architecture Component Definitions** section provides narrative describing and explaining each architecture component in the System Architecture Model, and identifies specific elements that comprise that component in this system.

|  |  |
| --- | --- |
| **Architecture Component** | **Component Elements** |
| Database Server | Oracle 11g RAC will be installed over Oracle Linux. |

### System Architecture Component – Application Server

|  |  |
| --- | --- |
| Architecture Component |  |
| Application Server | Oracle Weblogic Server 10.1.6 will be installed over Oracle Linux. Application servers will be in Active-Active custer mode |

### System Architecture Component – SOA Server

|  |  |
| --- | --- |
| Architecture Component |  |
| SOA Server | Oracle SOA server - 11.1.1.6 will be installed over Oracle Linux. Two node of SOA Server will be in Active-Active cluster Mode |

### System Architecture Component – LDAP Server

|  |  |
| --- | --- |
| Architecture Component |  |
| Apache Directory Server | Apache DS will be installed over Oracle Linux as primary secondary failover mode |

### System Architecture Component – Web Server

|  |  |
| --- | --- |
| Architecture Component |  |
| Apache Web Server | Apache Web Server will be installed over Oracle Linux as active-active mode and will be fronted by a hardware load balancer. |

### Software License Information

|  |  |  |
| --- | --- | --- |
| **SL.No** | **Product** | **License Information  ( in Processor based )** |
| 1 | Oracle Database Enterprise Edition | 24 |
| 2 | Real Application Cluster | 16 |
| 3 | Partitioning | 24 |
| 4 | Oracle Active Data Guard | 24 |
| 5 | Diagnostic Pack | 24 |
| 6 | Tuning Pack | 24 |
| 7 | Oracle WebCenter Portal | 9 |
| 8 | Weblogic Suite | 12 |
| 9 | SOA Suite for Oracle Middleware | 12 |
| 10 | SOA Maagement Pack Enterprise Edition | 8 |

# Section 3 System Construction Environment

*The* ***System Construction Environment*** *section details the various environments necessary to enable system construction and testing.*

## 3.1 Development Environment

### Developer Workstation Configuration

Each developer workstation must be configured with 4 G.B RAM and Weblogic 10.1.6 version.

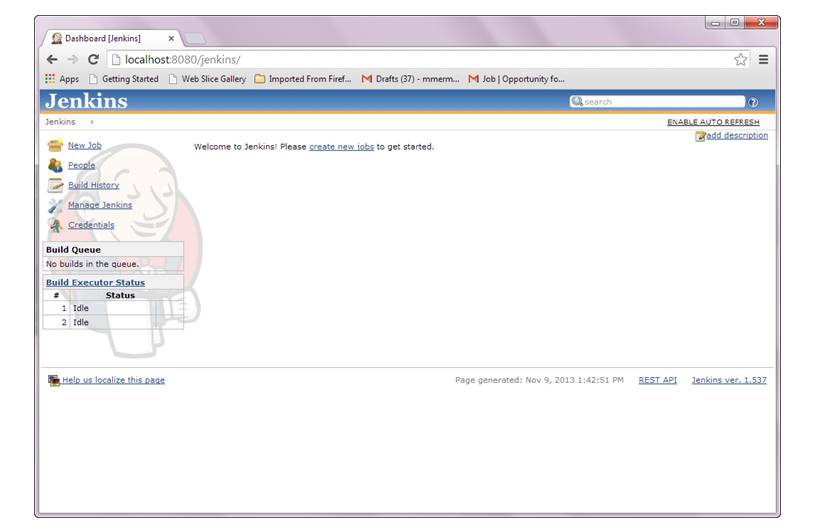
Eclipse must contain SVN, WebLogic, and Code Pro Plug-ins.

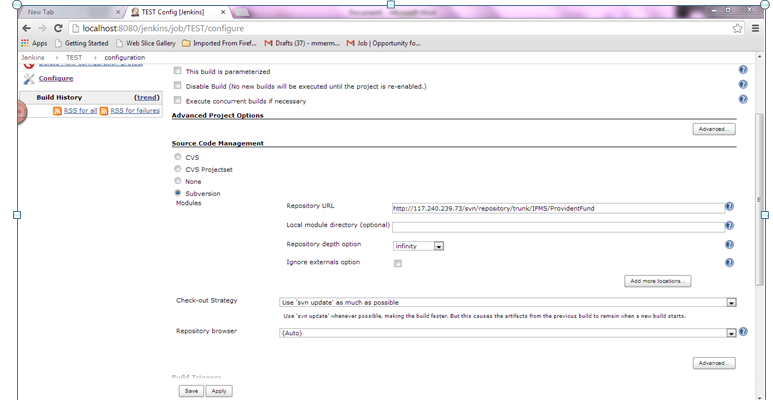
## *3.2* Code Quality Environment

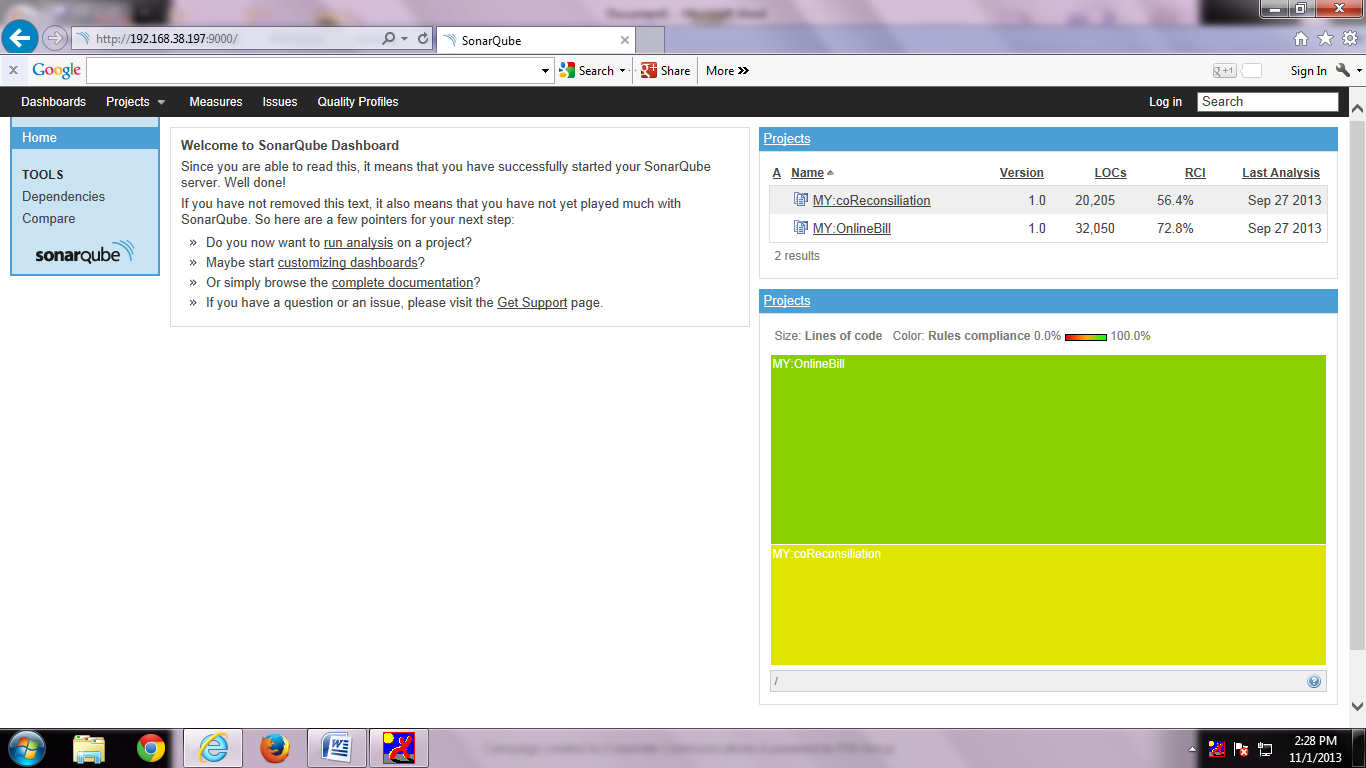
Code quality of iFMS will be ensured using industry standard best practices. Each developer will use Google CodePro as initial level of Quality assurance tool. In every week , code pro result will be shared with leads and based on given code review checklist application will be reviewed by peers and leads .

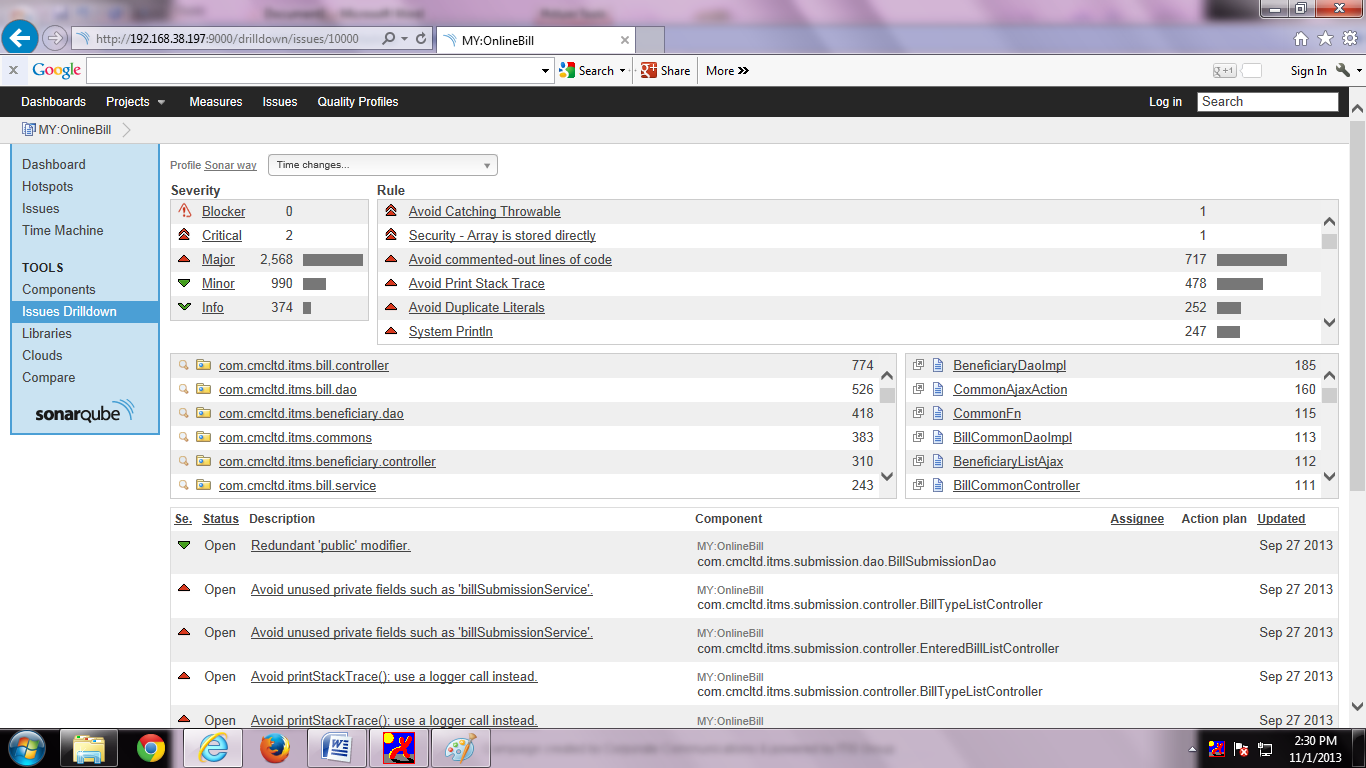
For continuous code quality assurance, SONAR is set up in development environment currently. All deployable codes will be assessed against SONAR tools and there must not be any ‘CRITIAL’ error in the code.

iFMS continuous Integration environment is setup using Jenkins. Jenkins allows to create scheduled jobs and these jobs will access code stored in SVN and then it will fire SONAR – RUNNER to gather code quality statistics. These statistics will be available using SONAR dashboard.









### *3.2.1* QA Workstation Configuration

As development IDE is used as Eclipse , google Code pro has eclipsed based plug ins. All develop workstation must be installed with eclipse with CodePro Plugins.

# Section 4 Architecture Issue and constrains

Following section describes architecture issue and constrains

* Requirements of integration and complete business functionalities are not available as Customer is providing phase by phase requirement, so complete and holistic service view and catalogs and data exchange formats cannot be defined until business requirements are given by customer.
* BI Servers are yet to be decided by the customer , hence integration with BI reports are not clear.
* Weblogic 10.1.6 does not completely support JPA 2.0 and CAS JPATicketRegistry used JPA 2.0 specification and hence there is a conflict. But, weblogic latest patch promises to take care that issue and hence deployment of CAS inside weblogic is under consideration.
* Data format of existing modules not available till date , hence migration logic and migration procedures are not written and tested as of date.
* Customer is not able to provide complete list of role and hence LDAP structure may undergo some changes in later time. Because of this all workflow are user based not role based.