Oracle® Banking Platform

Security Guide

Release 2.4.1.0.0

E70795-01

February 2016

ORACLE®

Oracle Banking Platform Security Guide, Release 2.4.1.0.0

E70795-01

Contents

List of Figures

List of Tables

Preface

This document provides a comprehensive overview of security for Oracle Banking Platform. It includes conceptual information about security principles, descriptions of the product's security features, and procedural information that explains how to use those features to secure Oracle Banking Platform.

This preface contains the following topics:

Audience

The Oracle Security Guide is intended for Bank IT Staff responsible for application installation and security configuration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Organization of the Guide

This document contains:

Chapter 1, "Overview"

This chapter presents an overview of Oracle Banking Platform and explains the general principles of application security.

Chapter 2, "Secure Installation and Configuration"

This chapter provides an overview of secure installation process through recommended deployment topologies and describes the installation and configuration procedure for the infrastructure and product components of Oracle Banking Platform.

Chapter 3, "Security Features"

This chapter outlines the specific security mechanisms offered by Oracle Banking Platform.

Appendix A, "Appendix"

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure Oracle Banking Platform.

Related Documents

For more information, see the following documentation:

- Hardening Tips for Default Installation of Red Hat Enterprise Linux 5 at http://www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf
- Oracle® Fusion Middleware Installation Guide for Oracle WebLogic Server at https://docs.oracle.com/middleware/11119/wls/WLSIG/toc.htm
- Oracle® Collaboration Suite Security Guide at http://docs.oracle.com/cd/B25553_01/collab.1012/b25494/toc.htm
- Oracle® Fusion Middleware Application Security Guide - Configuring and Managing Auditing at

  http://docs.oracle.com/cd/E23943_01/core.1111/e10043/audpolicy.htm
- For installation and configuration information, see the Oracle Banking Platform Installation Guide - Silent Installation
- For the complete list of Oracle Banking licensed products and the Third Party licenses included with the license, see the Oracle Banking Licensing Guide
- For information related to setting up a bank or a branch, and other operational and administrative functions, see the Oracle Banking Administrator's Guide
- For information related to customization and extension of Oracle Banking, see the Oracle Banking Extensibility Guide
- For information on the functionality and features of the Oracle Banking product licenses, see the respective Oracle Banking Functional Overview documents

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Overview

This chapter presents an overview of Oracle Banking Platform and explains the general principles of application security.

1.1 Product Overview

Oracle Banking Platform lays the foundation of a single unified Core Banking platform having the following features:

- Amalgamation of Origination, Business Banking, Direct Banking
- Common SMS
- Common Architectural Principles
- Enterprise Ready Business Services

1.2 General Security Principles

The following principles are fundamental for using any application securely.

1.2.1 Restrict Network Access to Critical Services

Keep both the Oracle Banking Platform middle-tier and the database behind a firewall. In addition, place a firewall between the middle-tier and the database. The firewalls provide assurance that access to these systems is restricted to a known network route, which can be monitored and restricted, if necessary. As an alternative, a firewall router substitutes for multiple, independent firewalls.

If firewalls cannot be used, be certain to configure the TNS Listener Valid Node Checking feature which restricts access based upon IP address. Restricting database access by IP address often causes application client or server programs to fail for DHCP clients. To resolve this, consider using static IP addresses, a software or a hardware VPN or Windows Terminal Services or its equivalent.

1.2.2 Follow the Principle of Least Privilege

The principle of least privilege states that users should be given the least amount of privilege to perform their jobs. User privileges should be reviewed periodically to determine relevance to current job responsibilities.

1.2.3 Monitor System Activity

System security stands on three legs:

1. Good security protocols
2. Proper system configuration
3. System monitoring

System needs to be constantly monitored from Oracle Enterprise Manager.

1.2.4 Keep Up To Date on Latest Security Information

Oracle continually improves its software and documentation.

2

Secure Installation and Configuration

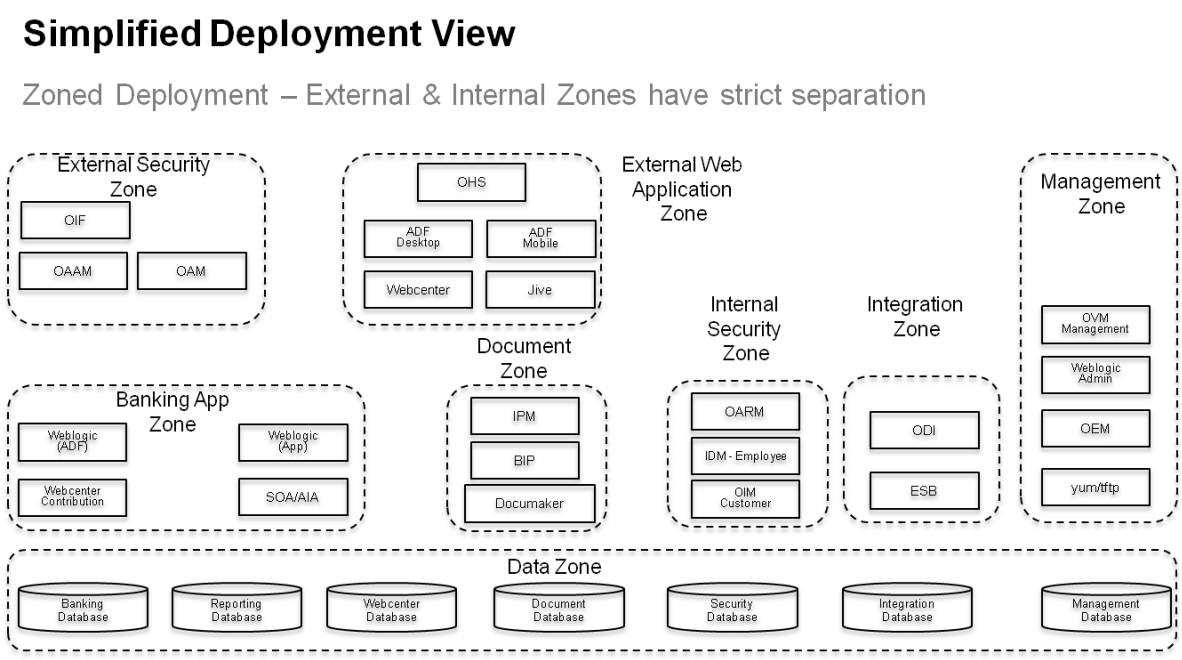
This chapter provides an overview of the recommended deployment topologies and describes the installation and configuration procedure for the infrastructure and product components of Oracle Banking Platform.

2.1 Recommended Deployment Topologies

This section describes the recommended deployment topologies for Oracle Banking Platform.

The simplified deployment view is as shown below:

Figure 2–1 Simplified Deployment View



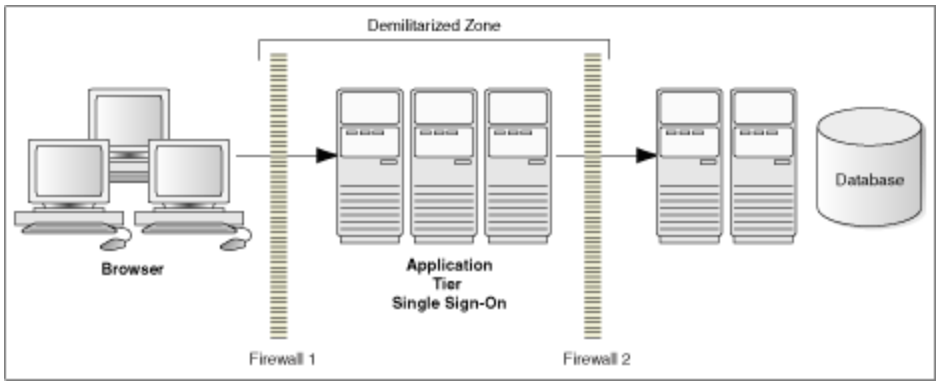
The deployment view for Oracle Banking Platform as shown in Figure 2–1 has the following features:

- Each zone is typically a separate network segment or subnet.
- Firewalls exist between each of these zones.

- The Document Zone and Integration Zones are shown for illustration purposes. Banks choose to typically deploy integration and document zones in the same Banking App Zone.
- Management Zone, Internal Security Zone and Banking Zone are typically an internal zone.
- Data is a separate zone.
- External Tiers have limited access to Data Zone.
  - This is for any personalization information that needs to be stored.
  - Banks may choose to deploy an external data zone which houses the personalization database.
- Access to core banking data (direct database access) is not allowed directly from the External Web Application Zone.
  - This would violate the defence in depth principle.
  - Access to core banking data is through services on HTTP protocol.

The general architectural recommendation is to use the well-known and generally accepted Internet-Firewall-DMZ-Firewall-Intranet architecture shown in Figure 2–2.

Figure 2–2 Traditional DMZ View



Note: The term Demilitarized Zone (DMZ) refers to a server that is isolated by firewalls from both the Internet and the intranet, thus forming a buffer between the two.

Firewalls separating DMZ zones provide two essential functions:

- Blocking any traffic types that are known to be illegal
- Providing intrusion containment, should successful intrusions take over processes or processors

2.2 Installing Linux

For installation of Oracle Banking Platform on RedHatEnterprise Linux 5, modify the default configuration following relevant instructions from the guide Hardening Tips for Default Installation of Red Hat Enterprise Linux 5 at the following location:

http://www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf

- Do not disable X Windows. It is needed for local administration and useful for troubleshooting.
- Do not disable SSH.

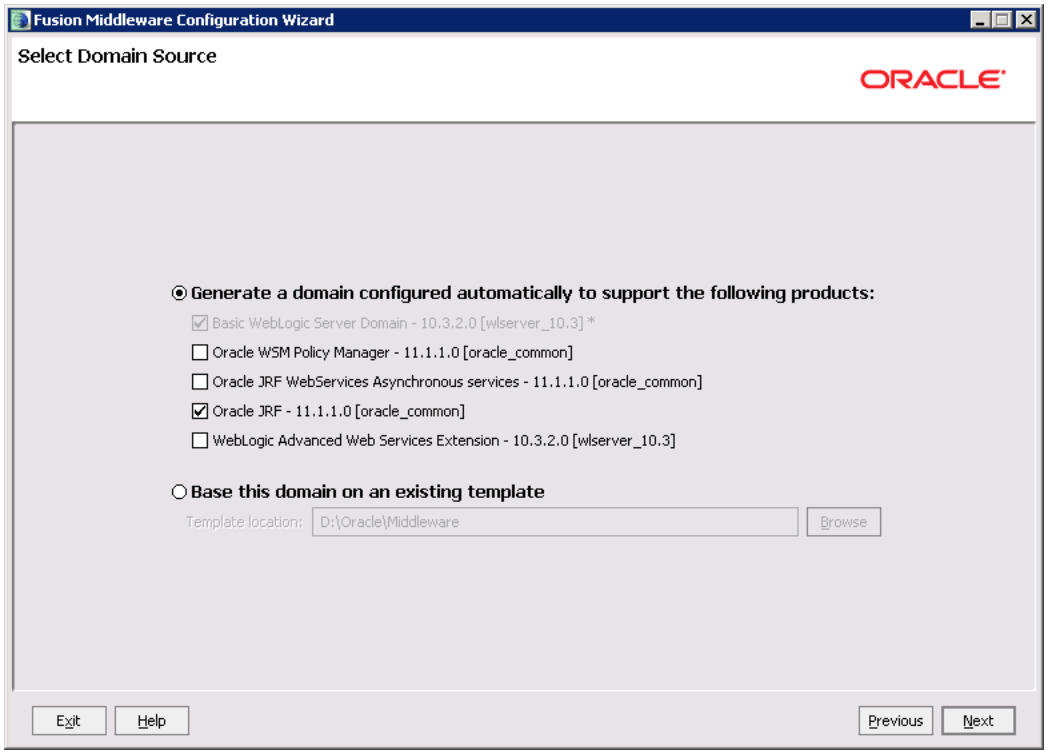
2.3 Installing WebLogic

Installation of WebLogic Server is done using the documentation as mentioned in the installation guide Oracle® Fusion Middleware Installation Guide for Oracle WebLogic Server at https://docs.oracle.com/middleware/11119/wls/WLSIG/toc.htm.

Following options need to be selected during the installation process:

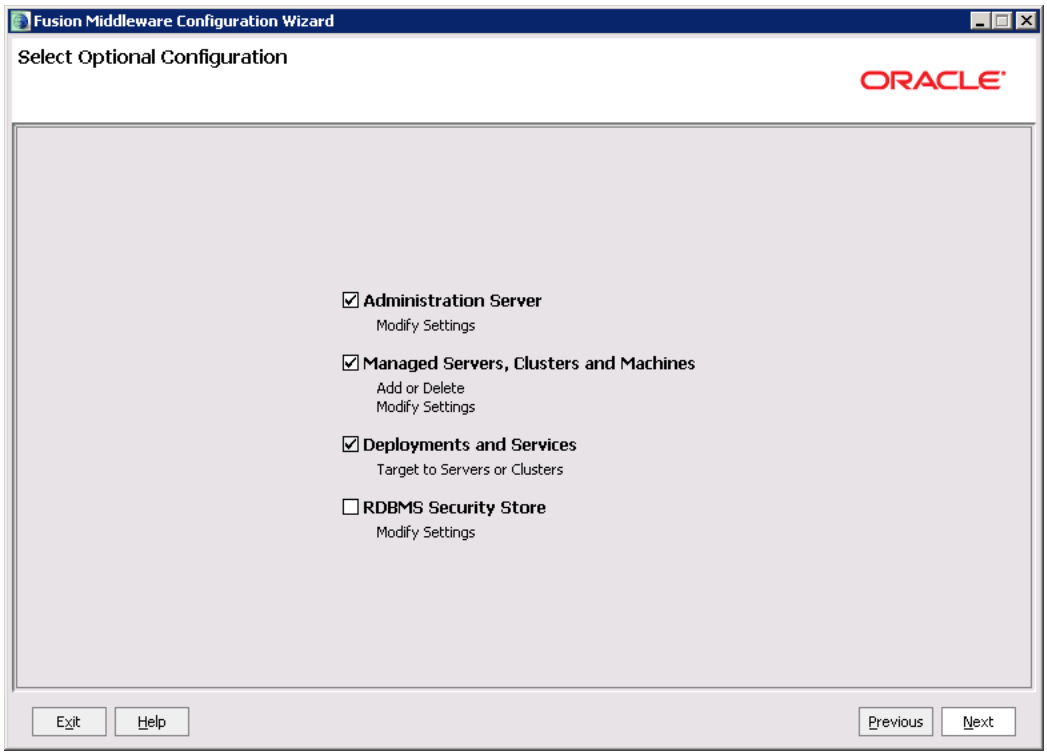
1. Select the option **Generate a domain configured automatically to support the following products:**
2. From the above option, select the **Oracle JRF - 11.1.1.0 [oracle_common] checkbox**.

Figure 2–3 Select Domain Source



3. Click **Next**.
4. Select the check box against the following options:
   - Administration Server
   - Managed Servers, Clusters and Machines
   - Deployments and Services

Figure 2–4 Select Optional Configuration



2.4 Installing Oracle Banking Platform

The detailed installation steps are present in the Oracle Banking Platform Installation Guide - Silent Installation.

2.5 Configuring SSL

One way SSL between the presentation and application WebLogic server is supported. The detailed configuration is explained below:

> **Note:** Procure an external CA signed certificate before proceeding further. Follow the instructions below to install the certificate once the certificate is available.

Step 1 Import the Certificate into a Java Trust Keystore

Execute the following command:

```
keytool -import -trustcacerts -alias sampletrustself -keystore
SampleTrust.jks -file SampleSelfCA.cer.der -keyalg RSA
```

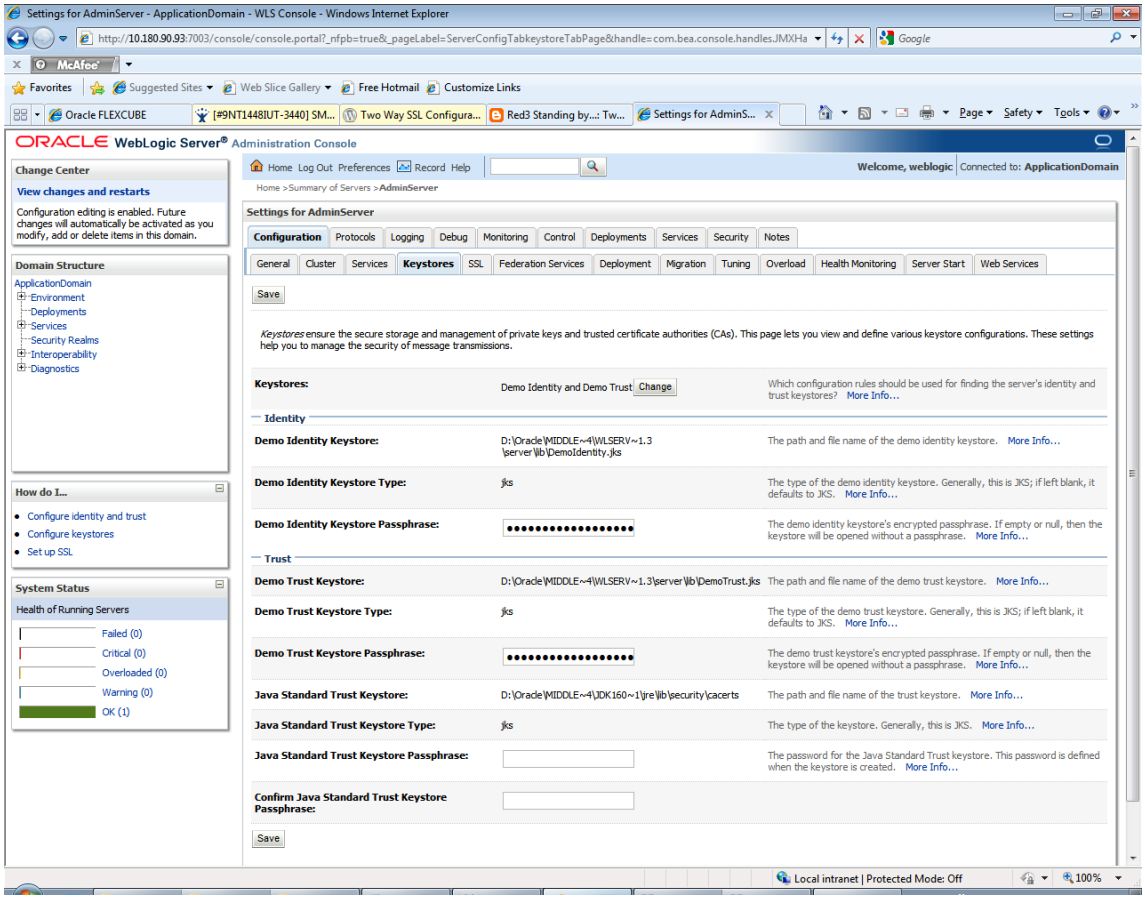
```
keytool -import -alias `hostname -f` -file `hostname -f`.cer -keystore
<JAVA_HOME>/jre/lib/security/cacerts -storepass changeit -noprompt
```

Step 2 Configure Application Domain's WebLogic with Custom Identity and Trust Keystores

To configure the application domain's WebLogic:

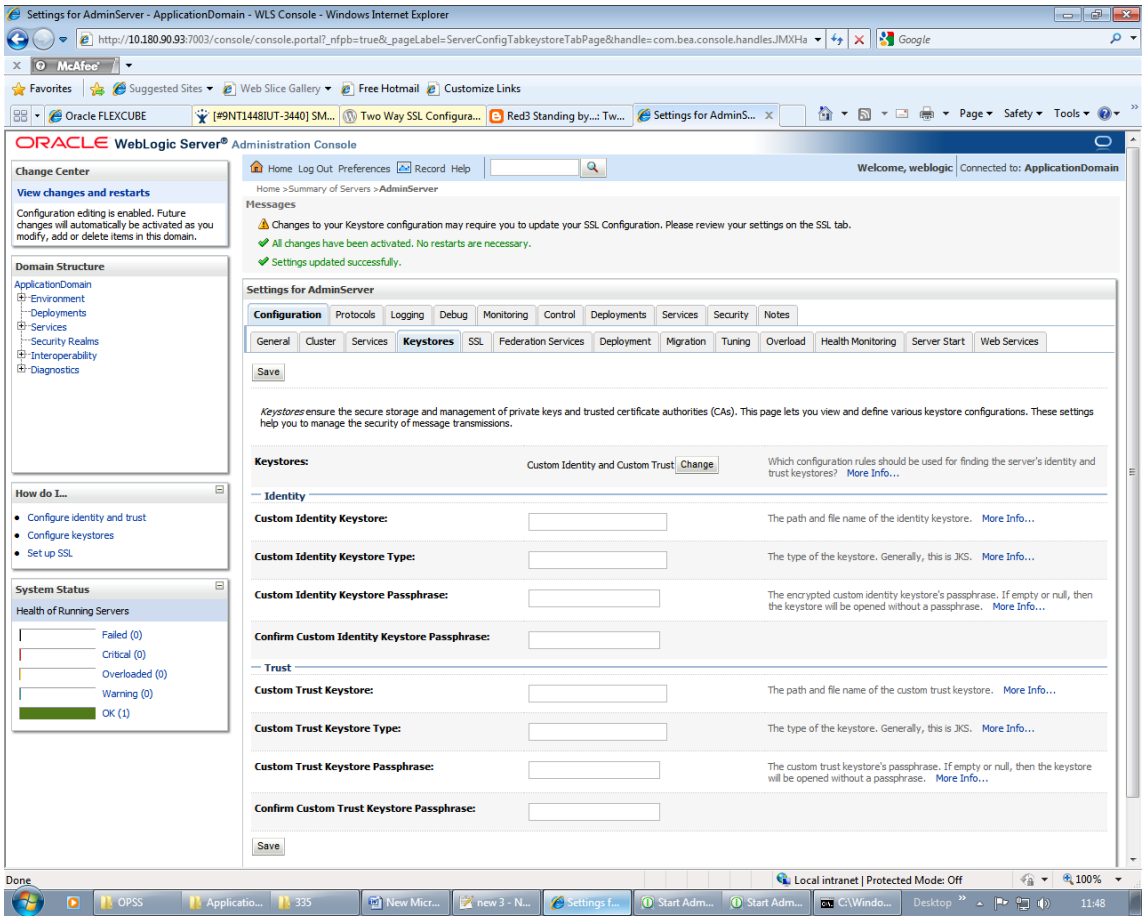
1. Open WebLogic admin console and navigate to **Home --> Summary of Servers --> AdminServer**. Click the **Keystores** tab.

Figure 2–5 Keystores



2. Click the **Change** button.
3. Select **Custom Identity and Java Standard Trust** option from the list.
4. Click the **Save** button.

Figure 2–6 Keystores - Identity and Trust



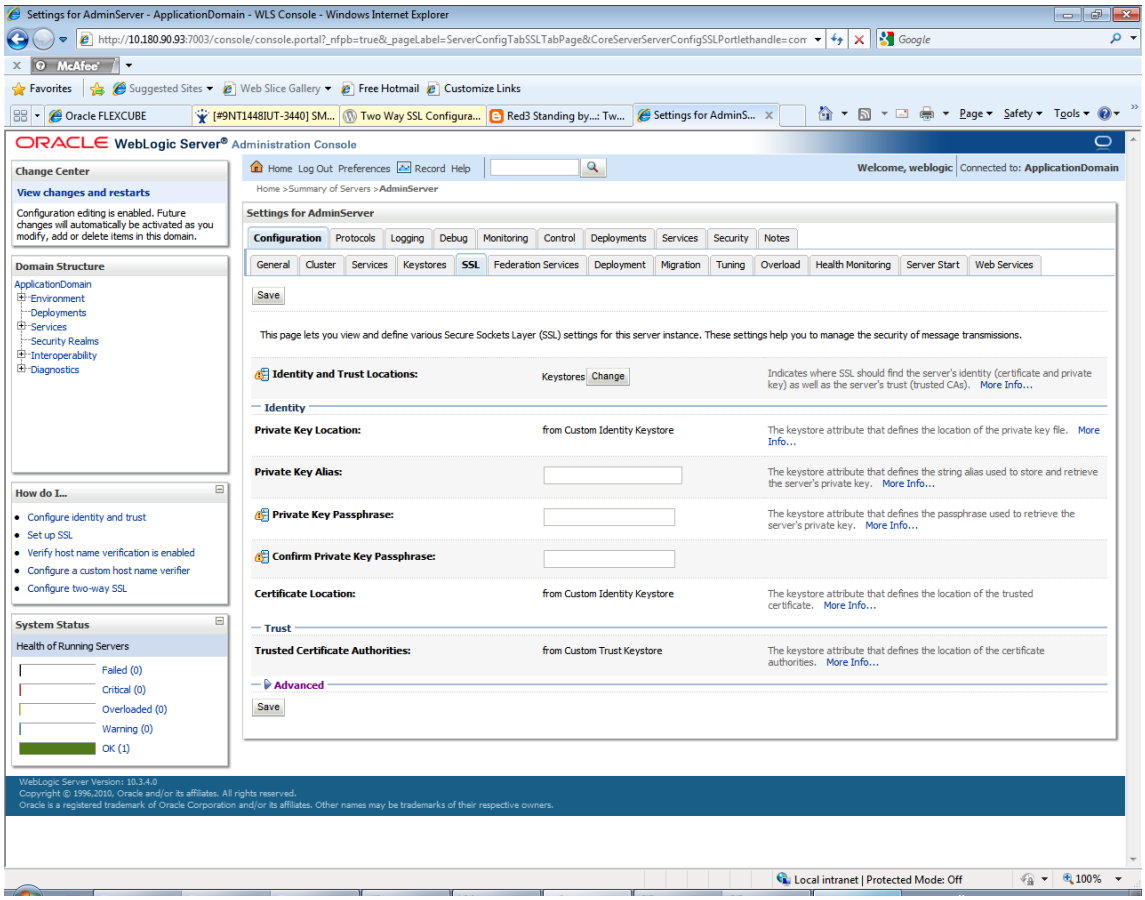
5. Enter the following details in the **Identity** and **Trust** sections:

Table 2–1 Keystore Configuration

Field	Value
Identity	
Custom Identity Keystore	Absolute path of `hostname -f`_identity.jck file
Custom Identity Keystore Type	JCKES
Custom Identity Keystore Passphrase	***
Confirm Custom Identity Keystore Passphrase	***

6. Enter the passphrases that were used while creating Identity Keystore and certificate.
7. Click the **Save** button.
8. Click the **SSL** Tab.

Figure 2–7 SSL



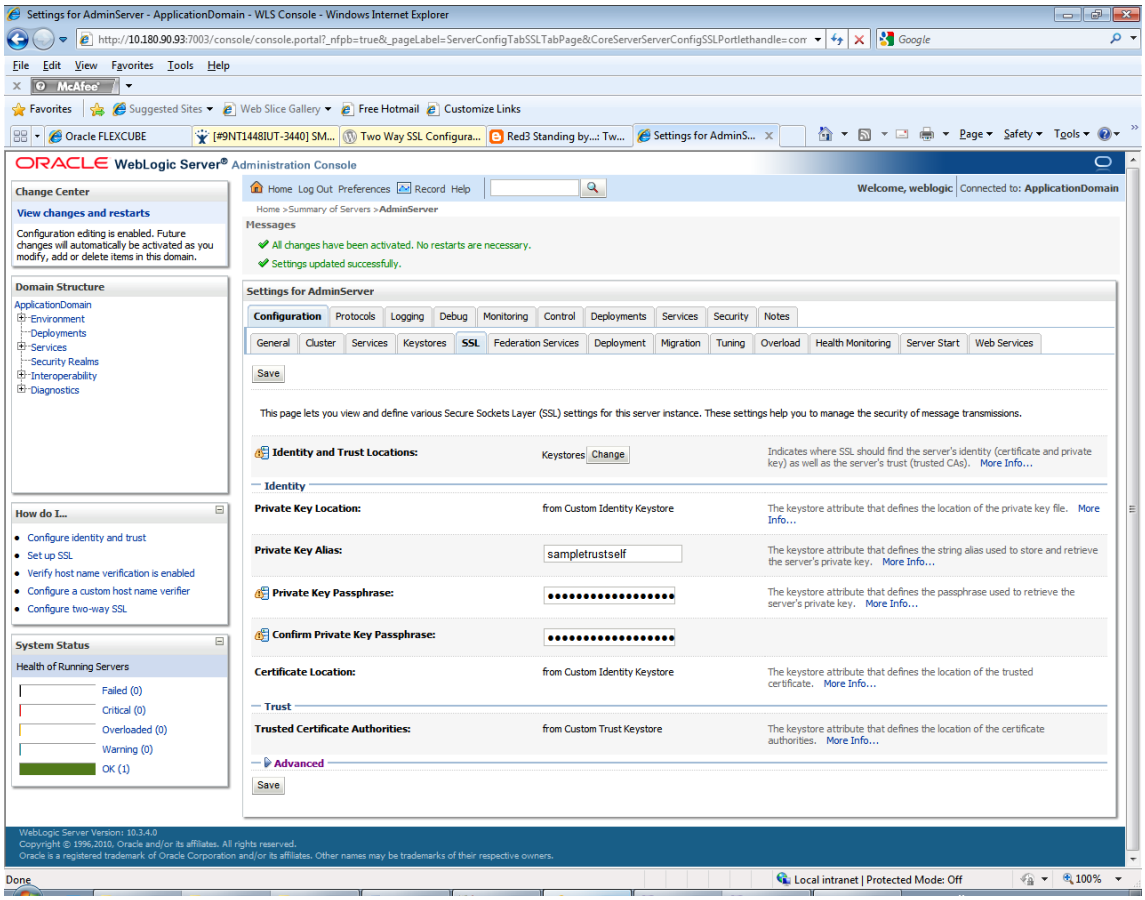
9. Enter the following details in the **Identity** section:

Table 2–2 SSL Configuration

Field	Value
Private Key Alias	`hostname -f`
Private Key Passphrase	***
Confirm Private Key Passphrase	***

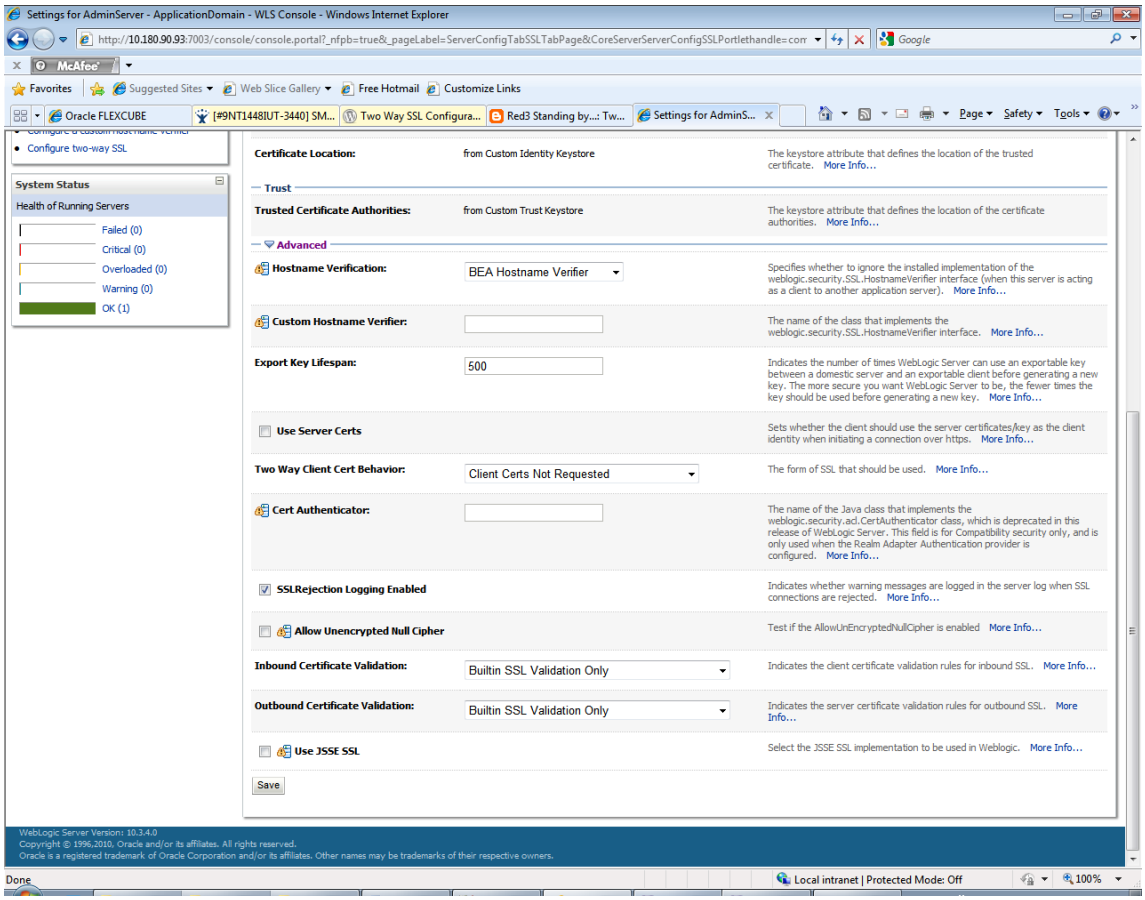
10. Enter the passphrases that were used while creating the certificate.

Figure 2–8 SSL Configuration



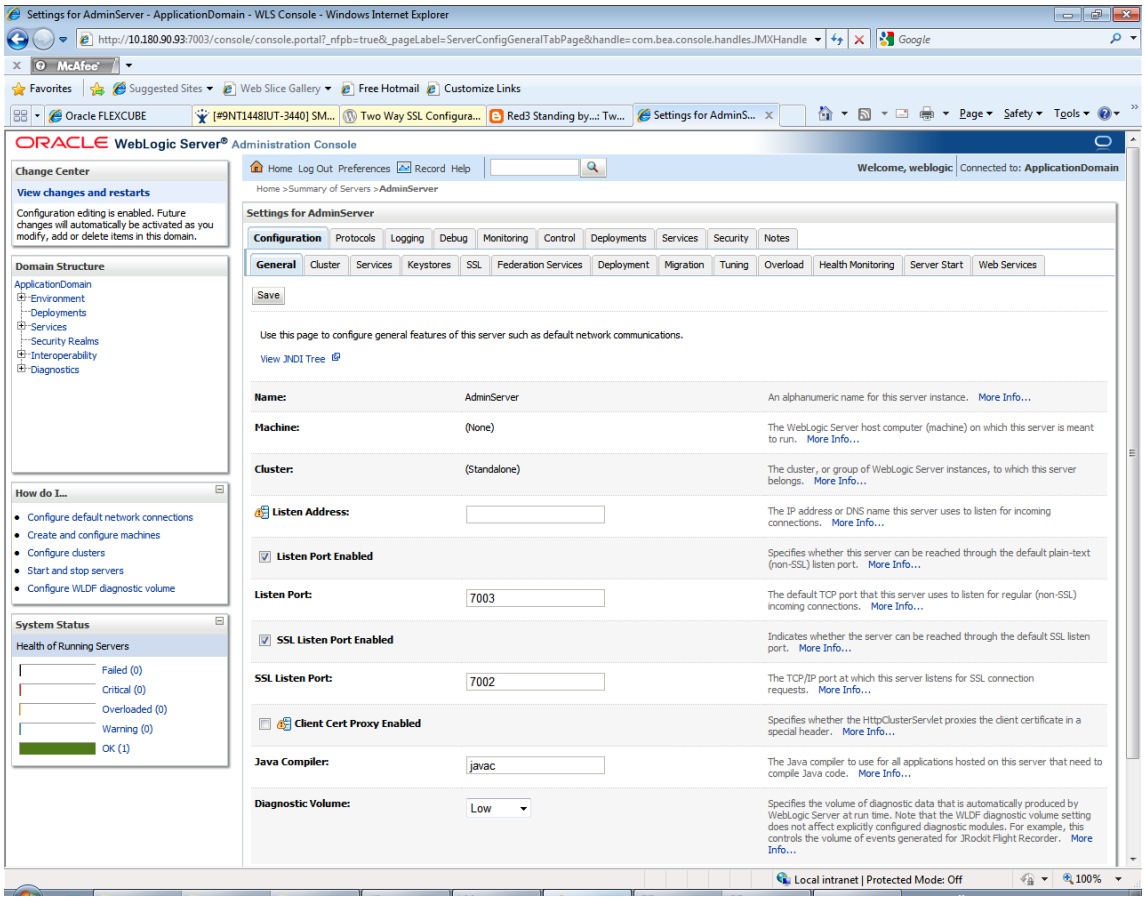
11. Click the **Save** button.
12. Click the **Advanced** link. Ensure that **Two Way Client Cert Behavior** is set to **Client Certs Not Requested**.

Figure 2–9 SSL - Advanced



13. Click the **General** tab. Select the **SSL Listen Port Enabled** check box.
14. Select the **Use JSSE SSL flag**.

Figure 2–10 General



15. Click the **Save** button.

Step 3 Restart Admin Server

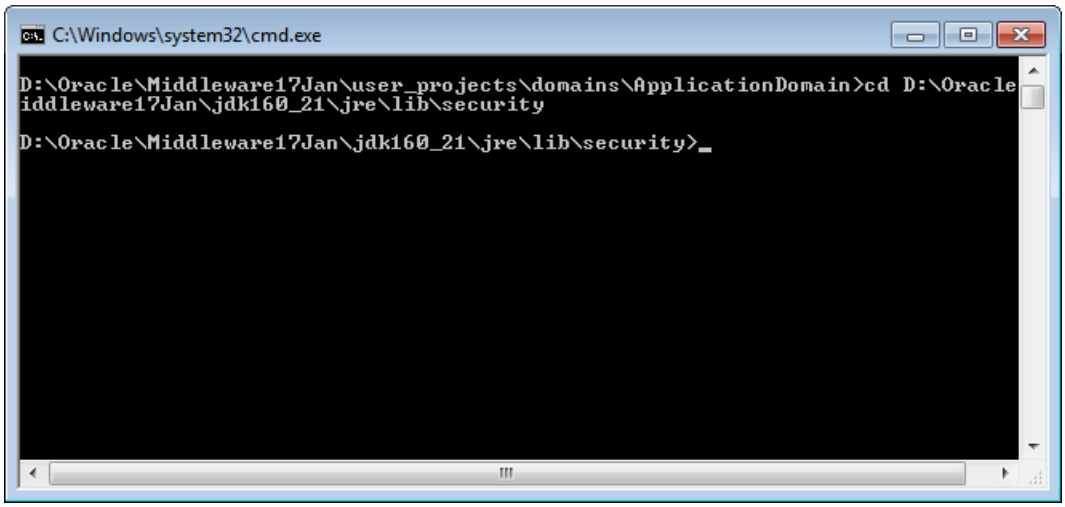
Restart the admin server of the Application Domain. Check the log file of admin server to ensure successful loading of the SSL configuration.

Step 4 Import Certificate in the JRE of Presentation Domain

To import the certificate:

1. Go to <MIDDLEWARE_HOME>\<JDK_HOME>\jre\lib\security

Figure 2–11 Presentation Domain Path



2. Execute the following command:

   ```
   keytool -import -alias sampletrustself -file D:\SampleSelfCA.cer
   -keystore cacerts
   ```

   Enter the keystore password when prompted to import the certificate in the JRE of the presentation domain.

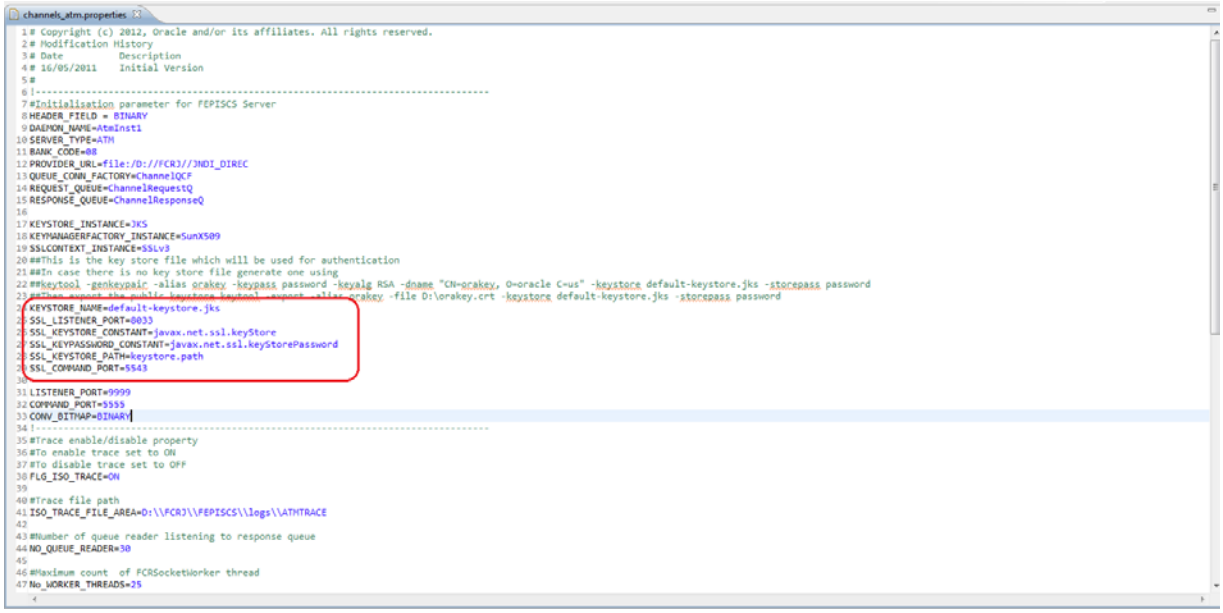
3. Finally, restart the admin server of the Presentation Domain.

Step 5 FEPI SSL Configuration

To enable SSL:

1. In the channel_atm.properties/channel_pos.properties, mention the keystore name as shown in the diagram below:

Figure 2–12 FEPI SSL Configuration



2. In the FEPI startup script, mention the keystore path as -Dkeystore.path="<ORACLE_MIDDLEWARE_HOME>/user_projects/domains/<DOMAIN_NAME>/config/fmwconfig ".
3. Executing the startup script, would prompt for host WebLogic username/password as well as the key and keystore password.
4. Check if FEPI has been started successfully using grep fepi.

Step 6 Web Service SSL configuration

All the host application web services are secured using the OWSM security policies.

The policy to be applied to the web service is defined in config/properties/SecurityAnnotations.properties

Sample entry is as follows:

com.ofss.fc.app.party.service.core.MDMPartyApplicationService=policy:oracle/wss_saml_token_over_ssl_service_policy

- In an SSL enabled environment, oracle/wss_saml_token_over_ssl_service_policy is used.
- @Policy annotation is added at the server startup in BootstrapServlet.

By default, SSLv3 should be disabled. The steps to disable SSLv3 protocol on Weblogic are as follows:

1. The weblogic.security.SSL.protocolVersion command-line argument lets you specify which protocol is used for SSL connections.
2. After enabling/configuring the SSL for weblogic server, append the following option to the JAVA_OPTIONS variable.

   -Dweblogic.security.SSL.protocolVersion=TLS1

> **Note:** If you do not specify the above property, it takes SSLv3 by default.

2.6 Post Installation Configuration

The security practices that should always be followed are listed below:

- Set the proper permissions for users accessing databases. You could also implement roles to manage privileges. Check whether permissions are correctly set in operating system. If these are not correctly set, there may be a security loophole.
- Implement TDE column encryption on the sensitive data.

3

Security Features

This chapter outlines the specific security mechanisms offered by Oracle Banking Platform.

3.1 Security Model

Application security requirements arise from the need to protect data, first, from accidental loss and corruption, and second, from deliberate unauthorized attempts to access or alter that data.

Secondary concerns include protecting against undue delays in accessing or using data, or even against interference to the point of denial of service.

The global costs of such security breaches run up to billions of dollars annually, and the cost to individual companies can be severe, sometimes catastrophic.

The critical security features that provide these protections are:

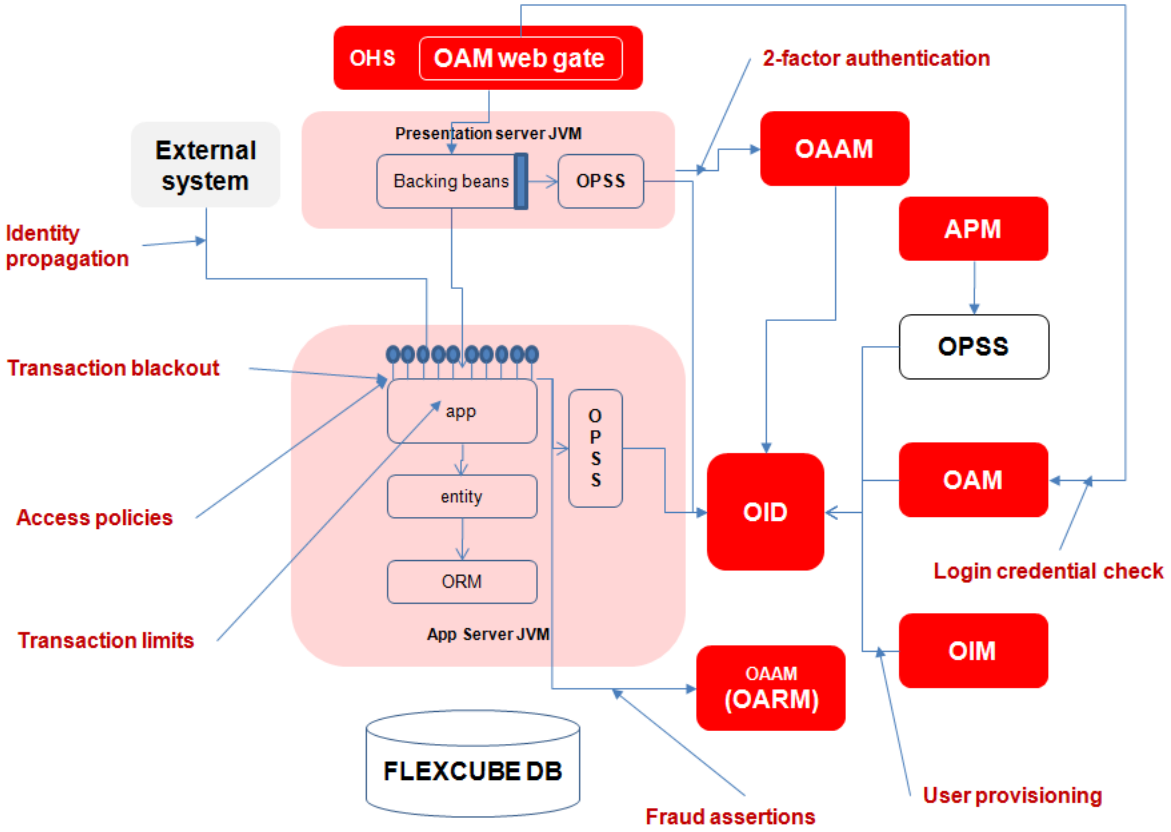
- **Authentication** – Ensures that only authorized individuals get access to the system and data.
- **Authorization** – Ensures access control to system privileges and data. This builds on authentication to ensure that individuals only get appropriate access. Oracle Database Vault will be used for this purpose.
- **Audit** – Allows administrators to detect attempted breaches of the authentication mechanism and attempted or successful breaches of access control.

The Oracle Banking Platform Security Architecture is explained in detail in the next section.

3.2 Security Architecture

Oracle Banking Platform comprises of several modules that interface with various systems in an enterprise to transfer or share data. This data is generated during business activity that takes place during teller operations or processing. While managing the transactions that are within OBP's domain, it also needs to consider security and identity management, and the uniform way in which these services need to be consumed by all applications in the enterprise. This is possible if these capabilities can be externalized from the application itself and are implemented within products that are specialized to handle such services. Examples of these services include authentication against an enterprise identity-store, creating permissions and role-based authorization model that controls access to not only the components of the application, but also the data that is visible to the user based on fine-grained entitlements.

Figure 3–1 Security - Participating Systems



The participating systems are as follows:

- Oracle Identity Manager (OIM) to be used for managing user provisioning.
- Oracle Access Manager (OAM) to be used for managing declarative authentication and SSO.
- Oracle Platform Security Services (OPSS) to be used for runtime evaluation of authentication/authorization.
- Oracle Adaptive Access Manager (OAAM)/Oracle Adaptive Risk Manager (OARM) to be used for step-up authentication and fraud management.
- Authorization Policy Manager (APM) to be used to manage access policy definitions.
- Oracle Internet Directory (OID) is used as the identity/policy store.

See the document Oracle® Collaboration Suite Security Guide at `http://docs.oracle.com/cd/B25553_01/collab.1012/b25494/toc.htm` for configuration details of the mentioned applications.

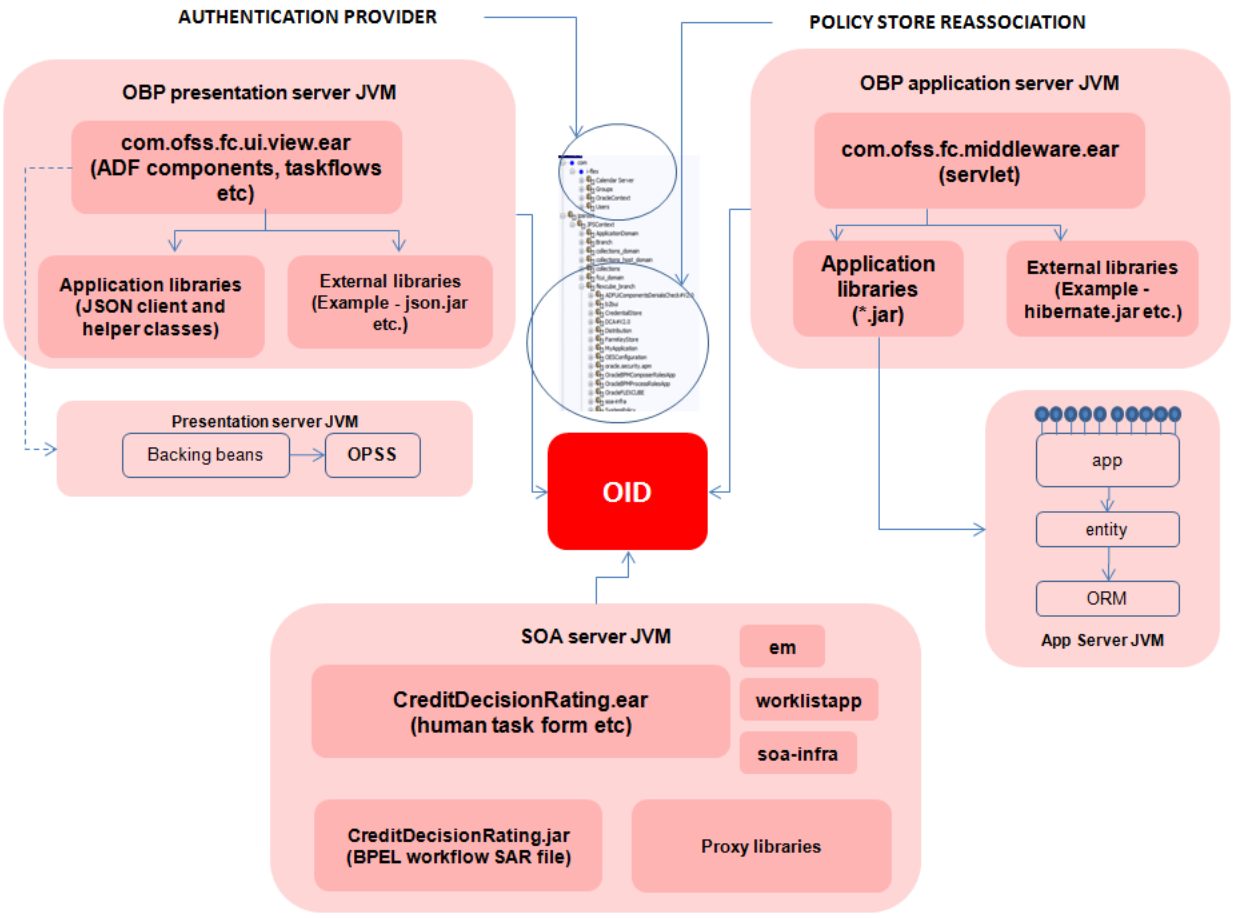
3.3 Approvals Architecture

Oracle Banking Platform is pre-integrated with the Oracle SOA Suite for executing its business workflows. The Originations module uses several process or human workflow features to originate customers and accounts. The Approvals module makes

use of the sophisticated participant assignment, routing or work-list features to fulfil the approvals use cases.

- The SOA suite identifies its users vide authentication provider pointing to OID. The OBP UI and app servers also point to the same identity store to provide authentication rights to its users.
- Work-list users or process users are protected vide access policies set up in OPSS. The SOA server domain is also re-associated to the same domain that the OBP UI and app-servers use to get the benefits of a centrally set up policy store.

Figure 3–2 Approvals - Participating Systems



Whenever a transaction is submitted by a user (banker, customer, and so on), security access check interceptors assert role-based access and fraud policies added on the service executed. Additionally, these interceptors also evaluate whether there are approvals configured on the service.

Approval checks are of the following types:

- **Dual Control** - Any transaction can be set up for approvals (2-eyes principle).
- **User Limits** - User Limits asserter evaluates whether transaction amount is within limits available to the user (role).

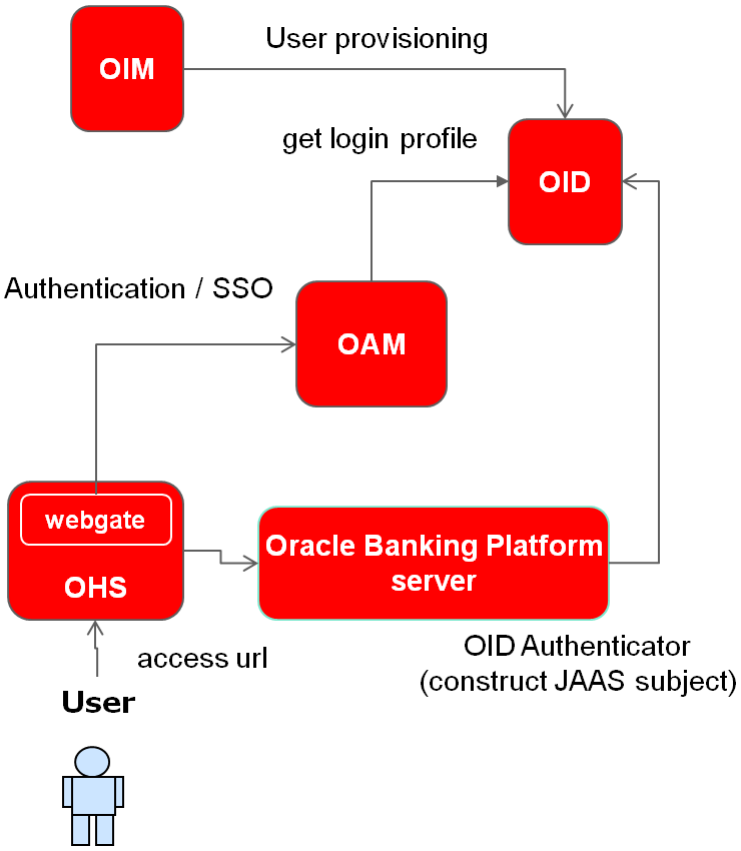
- **Matrix Based** - Matrix asserter evaluates a matrix of facts available in the context of the transaction. This asserter is used to evaluate the delegated commitment authority and discretionary pricing facts.

The output of these asserters is a decision on whether approvals are required or not. If approvals are required, system executes the process (BPEL) configured on the transaction. Thereon, the BPEL process takes the responsibility of routing the work-item to the configured assignees and seeking approvals from them. More details on this are available in the Static view, Dynamic view and inner mechanism chapters that follow.

3.4 Configuring and Using Authentication

Oracle Banking Platform uses OAM to authenticate users.

Figure 3–3 Authentication and Single Sign On



Data flow is as follows:

- OAM gets login profile from OID.
- OAM intercepts access call to Oracle Banking Platform and authenticates user.
- OAM ensures single sign-on across participating applications (configurable).
- SSO across various enterprise applications for internal users.

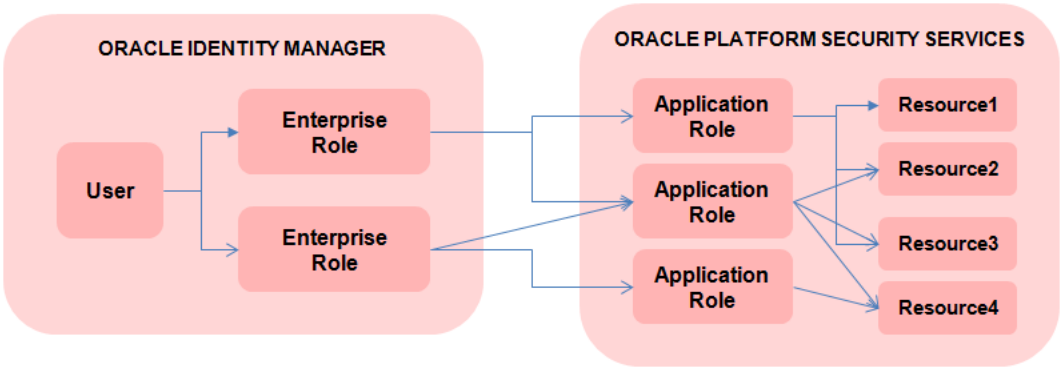
3.5 Configuring and Using Access Control

Authorization includes primarily two processes:

- Permitting only certain users to access, process, or alter transactions
- Applying varying limitations on user access or actions. The limitations placed on (or removed from) users can apply to transactions

Oracle Banking Platform uses OPSS Entitlements for authorization.

Figure 3–4 OPSS Entitlements - Users / Roles / Services



The features are:

- User belongs to the enterprise
- Users mapped to enterprise roles (used organization-wide)
- Enterprise roles mapped to application roles (application roles used within the application)
- Access policies defined for services defined on application roles

3.6 Configuring and Using Security Audit

Oracle Banking Platform relies on the Oracle Fusion Middleware Audit Framework for security audits.

The configuration and usage is explained in detail in the document Oracle® Fusion Middleware Application Security Guide - Configuring and Managing Auditing at http://docs.oracle.com/cd/E23943_01/core.1111/e10043/audpolicy.htm.

3.7 Configuring and Using TDE

Oracle Banking Platform relies on Oracle® Database Advanced Security for encrypting sensitive data.

The configuration is explained in detail in Oracle® Database Advanced Security Administrator's Guide.

OBP supports both TDE Tablespace Encryption as well as TDE Column Encryption.

Steps to perform TDE, with sample commands, as shown below:

1. Create Directories in all respective node servers.

```
mkdir -p -m 0700 /oracle/app/admin/IN5FMT/wallet
```

```
ssh orkxintdb10 "mkdir -p -m 0700 /oracle/app/admin/IN5FMT/wallet"
ssh orkxintdb11 "mkdir -p -m 0700 /oracle/app/admin/IN5FMT/wallet"
ssh orkxintdb12 "mkdir -p -m 0700 /oracle/app/admin/IN5FMT/wallet"

ssh orkxintdb10 "mkdir -p -m 0700 /oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet"
ssh orkxintdb11 "mkdir -p -m 0700 /oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet"
ssh orkxintdb12 "mkdir -p -m 0700 /oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet"
```

2. Create wallet on all nodes of server.

```
orapki wallet create -wallet /oracle/app/admin/IN5FMT/wallet -pwd
'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp' -auto_login
```

3. Restart database.
4. Set Master Key from sqlplus.

```
orapki wallet display -wallet /oracle/app/admin/IN5FMT/wallet -pwd
'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp'
ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY
"iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp";
```

5. Shutdown database.
6. Copy wallets into directories of all servers.

```
cd /oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb10:/oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb11:/oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb12:/oracle/app/admin/IN5FMT/wallet

cp -p * /oracle/app/database/11.2.0.2/dbhome_1/admin/IN5FMT/wallet/
scp -p * oracle@orkxintdb10:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb11:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb12:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
```

7. Startup database.
8. For TDE Tablespace encryption, create tablespace as <Original>_Encrypted and give quota to owner.

```
CREATE TABLESPACE "FMTAPP_ENCRYPTED" DATAFILE SIZE 512M
AUTOEXTEND ON NEXT 104857600 MAXSIZE UNLIMITED
LOGGING ONLINE PERMANENT BLOCKSIZE 8192
EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO
ENCRYPTION USING 'AES256' DEFAULT STORAGE(ENCRYPT);

alter user FMTAPP quota unlimited on FMTAPP_ENCRYPTED;
```

9. Move the tables with sensitive data in the encrypted tablespace.

```
alter table FMTAPP.SAVING_GOAL move tablespace FMTAPP_ENCRYPTED;
alter table FMTAPP.TXN_CATEGORY move tablespace FMTAPP_ENCRYPTED;
alter table FMTAPP.CUST_TXNS move tablespace FMTAPP_ENCRYPTED;
```

10. Rebuild the indexes.

```
alter index FMTAPP.TXN_DATE_AC_INDEX rebuild;
alter index FMTAPP.TXN_UID_IDX rebuild;
alter index FMTAPP.CUST_TXN_ID_IDX rebuild;
alter index FMTAPP.SG_CUSTOMER_NUM_IDX rebuild;
```

11. For TDE column encryption, check for foreign key usage. TDE cannot be used to encrypt columns that are used in a foreign key. Verifying whether a column is used as part of a foreign key can be accomplished by examining the Oracle data dictionary.
12. Encrypt column using TDE.

```
table customers modify (credit_card encrypt);
create table billing_information ( first_name varchar2(40) ,last_name
varchar2(40) ,card_number varchar2(19) encrypt using 'AES256');
```

3.8 Securing Outbound Interactions

Oracle Banking Platform interacts with external systems like Oracle BIP, Oracle Customer Hub (OCH). These interactions are synchronous and asynchronous in nature.

Synchronous communication is achieved using JAX-WS.

The outbound webservice configurations are present in flx_fw_config_out_ws_cfg_b.

The configurations include URL, Service ID, StubService, and timeout. The IP address and port of the external system is defined in flx_fw_config_var_b.

For example, in case of BIP,

url=http://{servername}:{serverport}/xmlpserver/services/PublicReportService?wsdl

timeOut=10000

stubService=com.oracle.xmlns.oxp.service.publicreportservice.PublicReportServiceService

The security credentials are stored in WebLogic connectors defined during installation.

Asynchronous communication is achieved using remote JMS queue.

The queue configurations are present in flx_fw_config_all_b, where category_id = 'EndpointConfig'. The IP address and port of the external system is defined in flx_fw_config_var_b.

For example, in case of OCH,

OCH.QUEUE_CONNECTION_FACTORY=jms/aia/AIA_CustomerJMSQueueCF

OCH.QUEUE=jms/aia/AIA_CustomerJMSQueue

OCH.PROVIDER.URL=t3:// {servername}:{serverport}/

The security credentials are stored in WebLogic connectors defined during installation.

3.9 Securing Key Store

This section describes the securing key store details.

3.9.1 Generation

The certificate is regenerated during installation, with a default password. Therefore, it needs to be regenerated post installation.

To generate keystore 'cks-keystore.jceks', following command should be used:

```
keytool -genseckey -alias orakey -keypass <Password> -keyalg RSA -keysize
2048 -dname "CN=orakey, O=oracle C=us" -storetype jceks -keystore
cks-keystore.jceks -storepass <Password>
```

The command generates a public/private key pair for the entity. It creates a self-signed certificate that includes the public key and the distinguished name information. The certificate is associated with the private key in a keystore entry.

By default, the keystore files are generated with 2048 bit key. These are required to be packaged as part of the **com.ofss.fc.ixface.sms.jar** file. These certificates are located within encr folder in the **com.ofss.fc.ixface.sms.jar** file.

3.9.2 Certificate Validity and Regeneration

The certificate is valid for 90 days. This is the default validity period, if the validity option is not specified explicitly. On certificate expiry, it has to be regenerated and replaced in the encr folder within the **com.ofss.fc.ixface.sms.jar** file.

3.9.3 Generation with 2048 Bit Key

In order to generate higher than 128 bit key size, **Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy** files are required. These are available at the **Java SE download** page at
http://www.oracle.com/technetwork/java/embedded/embedded-se/downloads/jce-7-download-432124.html

The zip file contains policy jars, which you need to copy to overwrite the jars present in the *{java.home}/jre/lib/security* directory. This allows for key strength above 128 bits.

A

Appendix

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure Oracle Banking Platform.

A.1 Secure Deployment Checklist

The following security checklist includes guidelines that help secure your installation:

1. Install only what is required.
2. Lock and expire default user accounts.
3. Enforce password management.
4. Practice the principle of least privilege.
    a. Grant necessary privileges only.
    b. Revoke unnecessary privileges from the PUBLIC user group.
    c. Restrict permissions on run-time facilities.
5. Enforce access controls effectively and authenticate clients stringently.
6. Restrict network access.
    a. Use a firewall.
    b. Never poke a hole through a firewall.
    c. Monitor who accesses your systems.
    d. Check network IP addresses.
    e. Encrypt network traffic.
    f. Harden the operating system.
7. Apply all security patches and workarounds.
8. Contact Oracle Security Products if you come across vulnerability in Oracle Database.