

Oracle® Banking Deposits and Lines of Credit Servicing

Security Guide

Release 2.12.0.0.0

F42003-01

May 2021

Oracle Banking Deposits and Lines of Credit Servicing Security Guide, Release 2.12.0.0.0

F42003-01

Copyright © 2019, 2021 Oracle and/or its affiliates.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	7
Audience	7
Documentation Accessibility	7
Organization of the Guide	7
Related Documents	8
Conventions	8
1 About This Guide	10
2 Overview	12
2.1 Product Overview	12
2.2 General Security Principles	12
2.2.1 Restrict Network Access to Critical Services	12
2.2.2 Follow the Principle of Least Privilege	12
2.2.3 Monitor System Activity	12
2.2.4 Keep Up To Date on Latest Security Information	13
3 Secure Installation and Configuration	14
3.1 Recommended Deployment Topologies	14
3.2 Installing Linux	15
3.3 Installing WebLogic	16
3.4 Installing Oracle Banking Deposits and Lines of Credit Servicing	17
3.5 Configuring SSL	17
3.6 Post Installation Configuration	26
4 Security Features	28
4.1 Security Model	28
4.2 Security Architecture	28

4.2.1 OIM Based Security	28
4.2.2 Local Security	29
4.3 Approvals Architecture	30
4.4 Configuring and Using Authentication	31
4.5 Configuring and Using Access Control	32
4.6 Configuring and Using Security Audit	34
4.7 Configuring and Using TDE	34
4.8 Securing Outbound Interactions	36
4.9 Securing Key Store	36
4.9.1 Generation	36
4.9.2 Certificate Validity and Regeneration	37
4.9.3 Generation with 2048 Bit Key	37
5 Data Privacy and Security	38
5.1 Data Minimization	38
5.2 Data Portability	39
5.3 Encryption	40
5.3.1 Key Management	40
5.3.2 Secure Storage of Encryption Key	40
5.3.3 Secure Access of Encryption Key	41
5.4 Tracking Technologies	41
5.5 Separate Auditing and Detective Control Privileges	41
5.5.1 Application Logs	41
5.6 Logging	41
5.6.1 Application Logs	42
Appendix	44
Secure Deployment Checklist	44

List of Figures

Figure 3–1 Simplified Deployment View	14
Figure 3–2 Traditional DMZ View	15
Figure 3–3 Select Domain Source	16
Figure 3–4 Select Optional Configuration	17
Figure 3–5 Keystores	18
Figure 3–6 Keystores - Identity and Trust	19
Figure 3–7 SSL	20
Figure 3–8 SSL Configuration	21
Figure 3–9 SSL - Advanced	22
Figure 3–10 General	23
Figure 3–11 Presentation Domain Path	24
Figure 4–1 Security - Participating Systems	29
Figure 4–2 Local Security – High Level Architecture	30
Figure 4–3 Approvals - Participating Systems	31
Figure 4–4 Authentication and Single Sign On	32
Figure 4–5 OPSS Entitlements - Users / Roles / Services	32
Figure 5–1 Bank Admin as PII Data Originator	38
Figure 5–2 Bank Teller as PII Data Originator from Application Form	39
Figure 5–3 Bank Teller as PII Data Originator from Single Party View	39

List of Tables

Table 3–1 Keystore Configuration	19
Table 3–2 SSL Configuration	20
Table 3–3 Key fields in the XML token	26
Table 4–1 Factory-Shipped Roles	33
Table 5–1 Java Key Store Parameters	40
Table 5–2 Encryption Key Parameters	41
Table 5–3 Encryption Key Parameters	41

Preface

This document provides a comprehensive overview of security for Oracle Banking Deposits and Lines of Credit Servicing. 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 Deposits and Lines of Credit Servicing.

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Organization of the Guide](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This 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 that have purchased support 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 About This Guide](#)

This chapter provides details about applicability of this guide.

[Chapter 2 Overview](#)

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

[Chapter 3 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 the application.

[Chapter 4 Security Features](#)

This chapter outlines the specific security mechanisms offered by the application.

Chapter 5 Data Privacy and Security

This chapter explains the data privacy and security features offered by application.

Appendix

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure the application.

Related Documents

For more information, see the following documentation:

- Hardening Tips for Default Installation of Oracle Enterprise Linux 6 at https://docs.oracle.com/cd/E40518_01/server.761/es_security/src/csec_os_harden_linux.html
- 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 Deposits and Lines of Credit Servicing Localization Installation Guide - Silent Installation guide.
- For the complete list of licensed products and the third-party licenses included with the license, see the Oracle Banking Deposits and Lines of Credit Servicing Licensing Guide.
- For information related to setting up a bank or a branch, and other operational and administrative functions, see the Oracle Banking Deposits and Lines of Credit Servicing Administrator Guide.
- For information related to customization and extension, see the Oracle Banking Deposits and Lines of Credit Servicing Extensibility Guides for HOST, SOA, and UI.
- For information on the functionality and features, see the respective Oracle Banking Deposits and Lines of Credit Servicing Functional Overview document.
- For recommendations of secure usage of extensible components, see the Oracle Banking Deposits and Lines of Credit Servicing Secure Development Guide.

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 About This Guide

This guide is applicable for the following products:

- Oracle Banking Platform
- Oracle Banking Enterprise Originations
- Oracle Banking Enterprise Default Management
- Oracle Banking Loans Servicing
- Oracle Banking Deposits and Lines of Credit Servicing

References to Oracle Banking Platform or OBP in this guide apply to all the above mentioned products.

2 Overview

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

2.1 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

2.2 General Security Principles

The following principles are fundamental for using any application securely.

2.2.1 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.

2.2.2 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.

2.2.3 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.

2.2.4 Keep Up To Date on Latest Security Information

Oracle continually improves its software and documentation.

3 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 Deposits and Lines of Credit Servicing.

3.1 Recommended Deployment Topologies

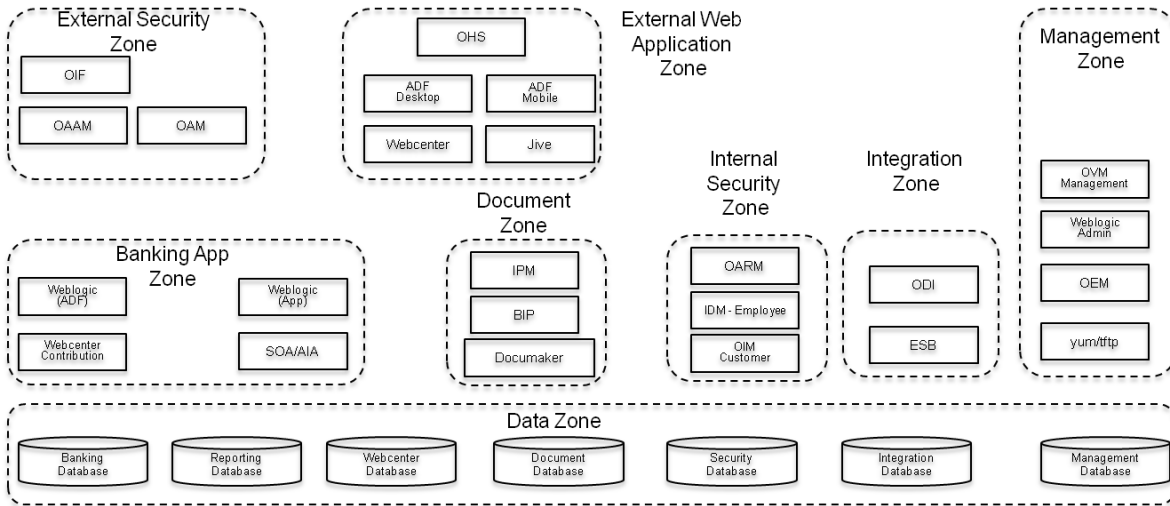
This section describes the recommended deployment topologies for Oracle Banking Deposits and Lines of Credit Servicing.

The simplified deployment view is as shown below:

Figure 3–1 Simplified Deployment View

Simplified Deployment View

Zoned Deployment – External & Internal Zones have strict separation



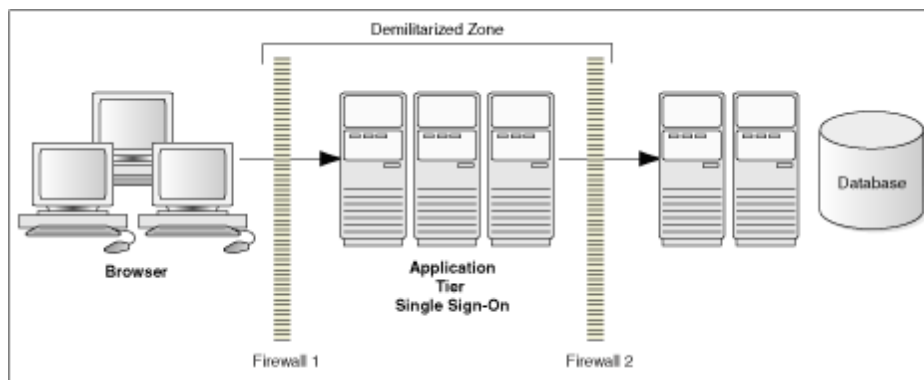
The deployment view for Oracle Banking Deposits and Lines of Credit Servicing as shown in [Figure 3–1](#) has the following features:

- 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 3–2](#).

Figure 3–2 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

3.2 Installing Linux

For installation of Oracle Banking Deposits and Lines of Credit Servicing on Oracle Enterprise Linux 6, modify the default configuration following relevant instructions from the guide [Hardening Tips for Default Installation of Oracle Enterprise Linux 6](#) at the following location:

https://docs.oracle.com/cd/E40518_01/server.761/es_security/src/csec_os_harden_linux.html

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

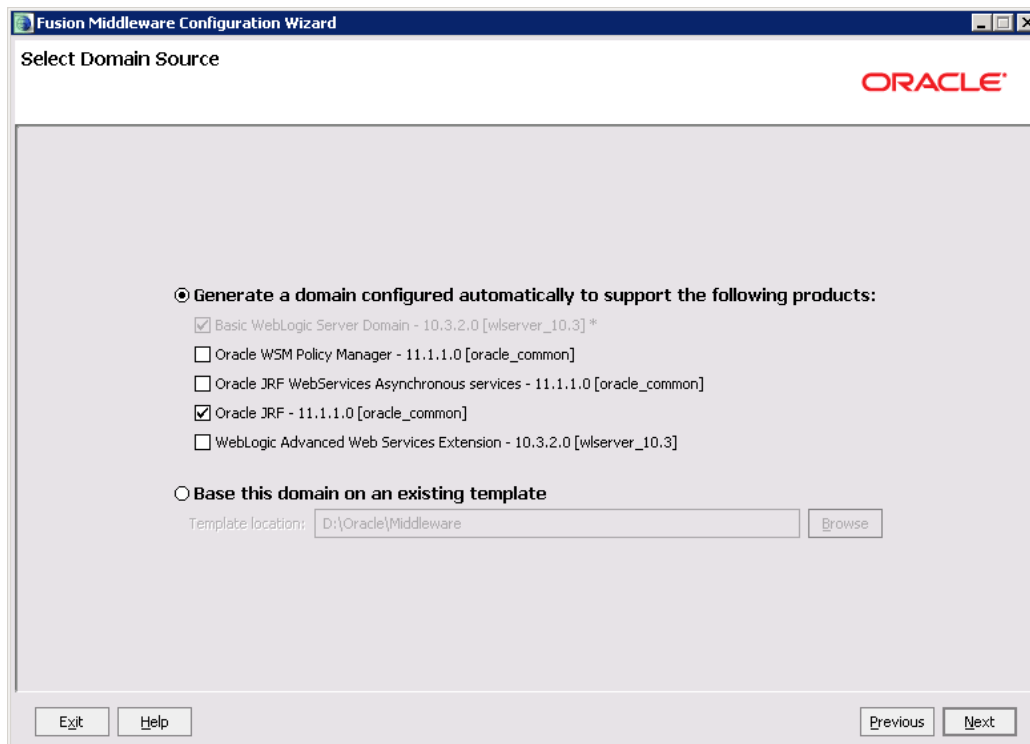
3.3 Installing WebLogic

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:

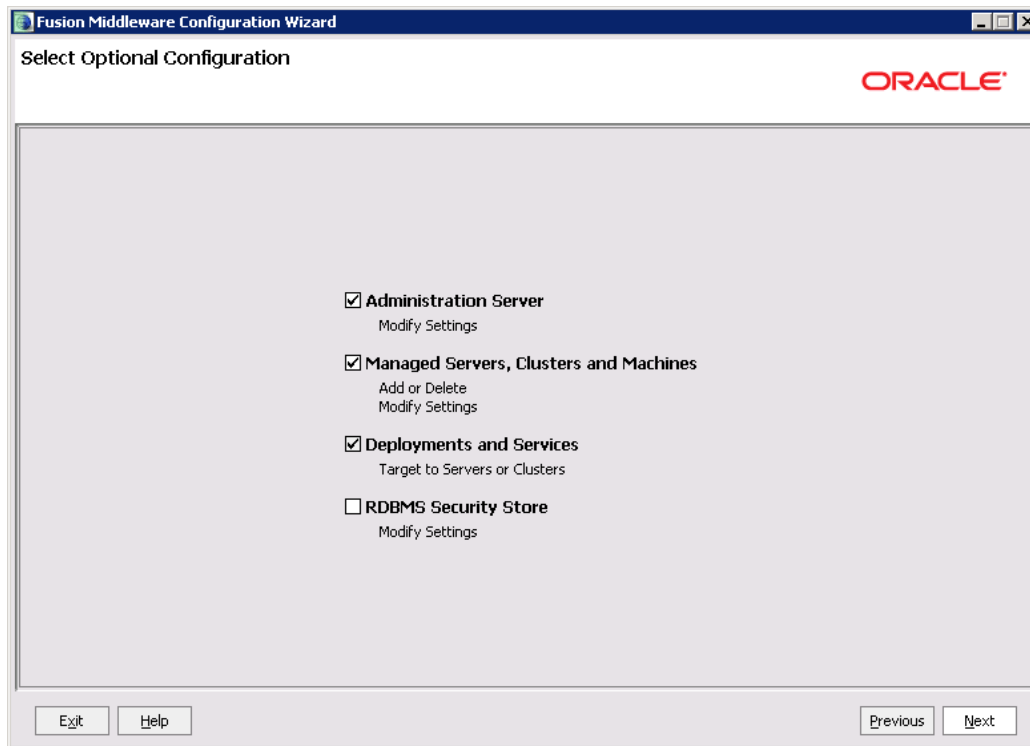
1. Select the option **Generate a domain configured automatically to support the following products:**
2. From the above option, select the **Oracle JRF - 12.2.1.4 [oracle_common]** check box.

Figure 3–3 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 3–4 Select Optional Configuration



3.4 Installing Oracle Banking Deposits and Lines of Credit Servicing

The detailed installation steps are present in the Oracle Banking Deposits and Lines of Credit Servicing Installation Guide - Silent Installation.

3.5 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
```

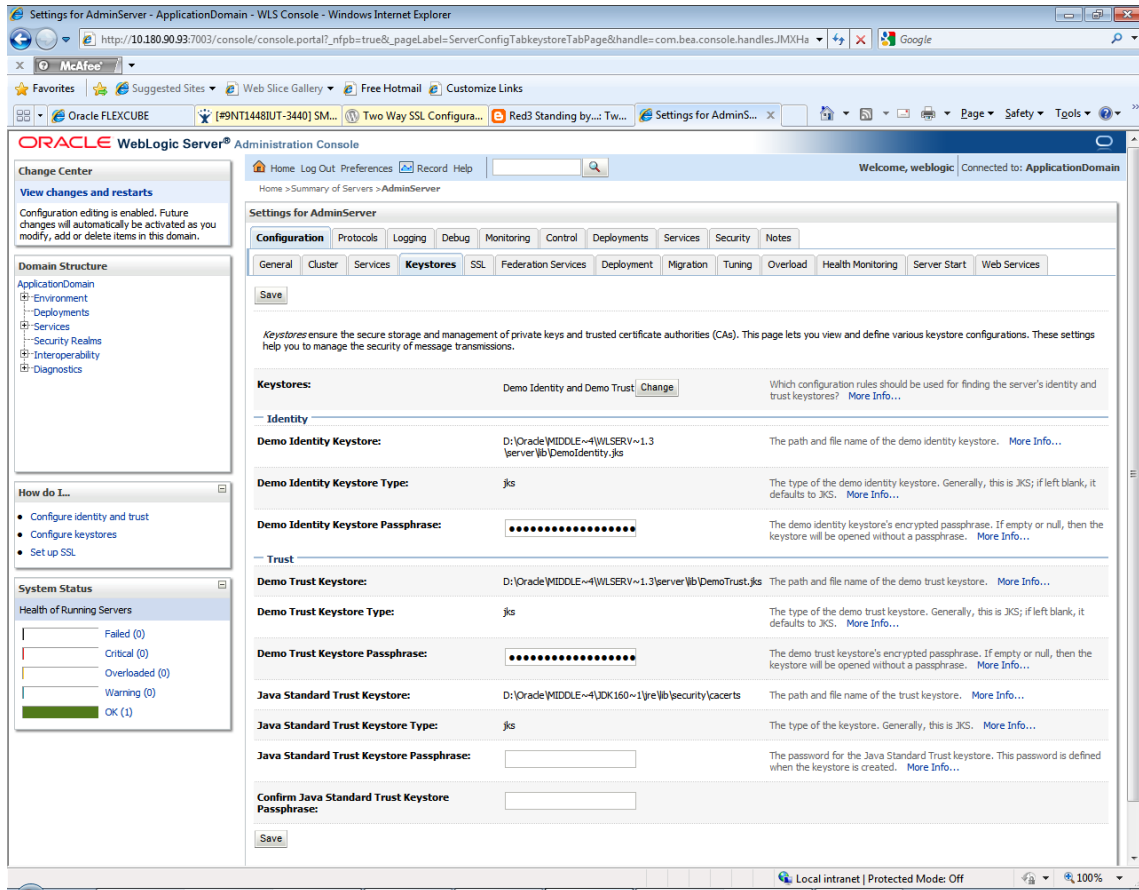
```
keytool -import -alias `hostname -f` -file `hostname -f`.cer -keystore <JAVA_
HOME>/jre/lib/security/cacerts -storepass changeit -noprompt
```

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

To configure the application domain's WebLogic:

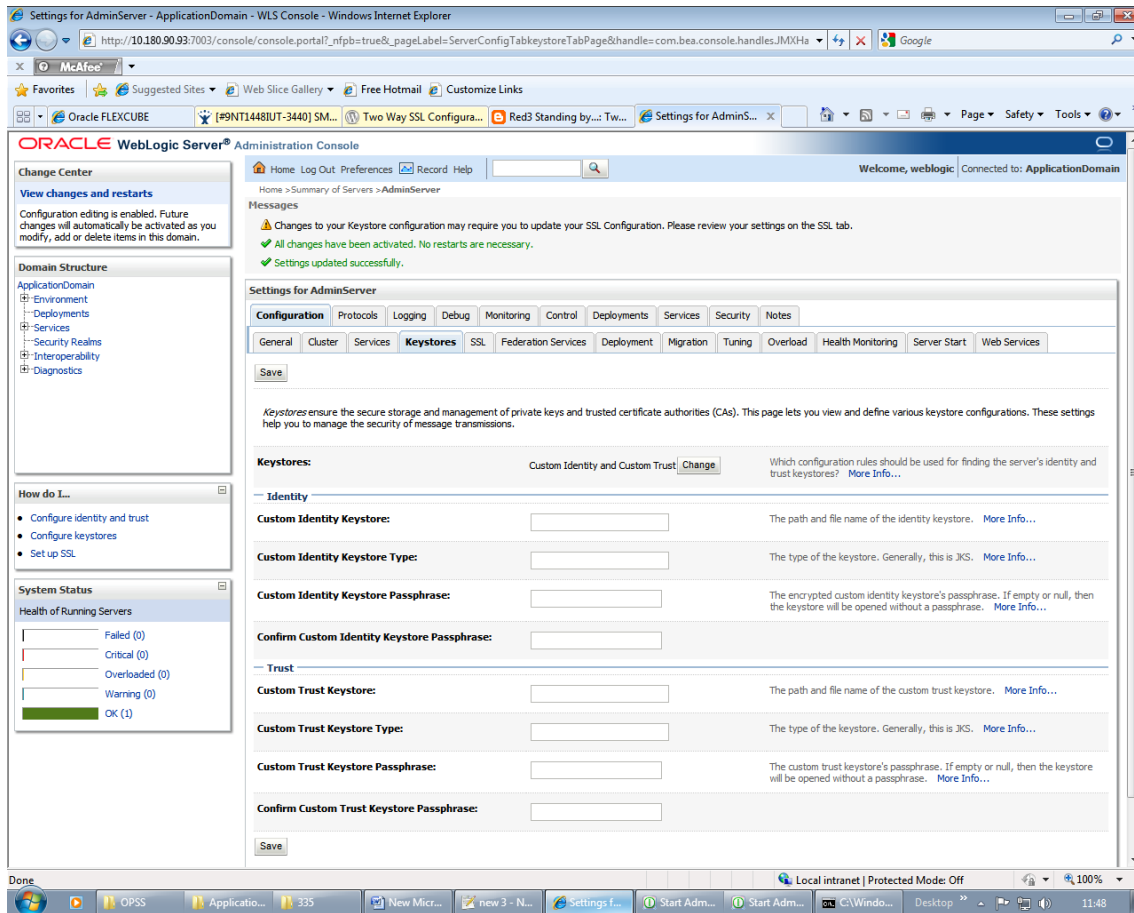
1. Open WebLogic admin console and navigate to **Home --> Summary of Servers --> AdminServer**. Click the **Keystores** tab.

Figure 3–5 Keystores



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

Figure 3–6 Keystores - Identity and Trust



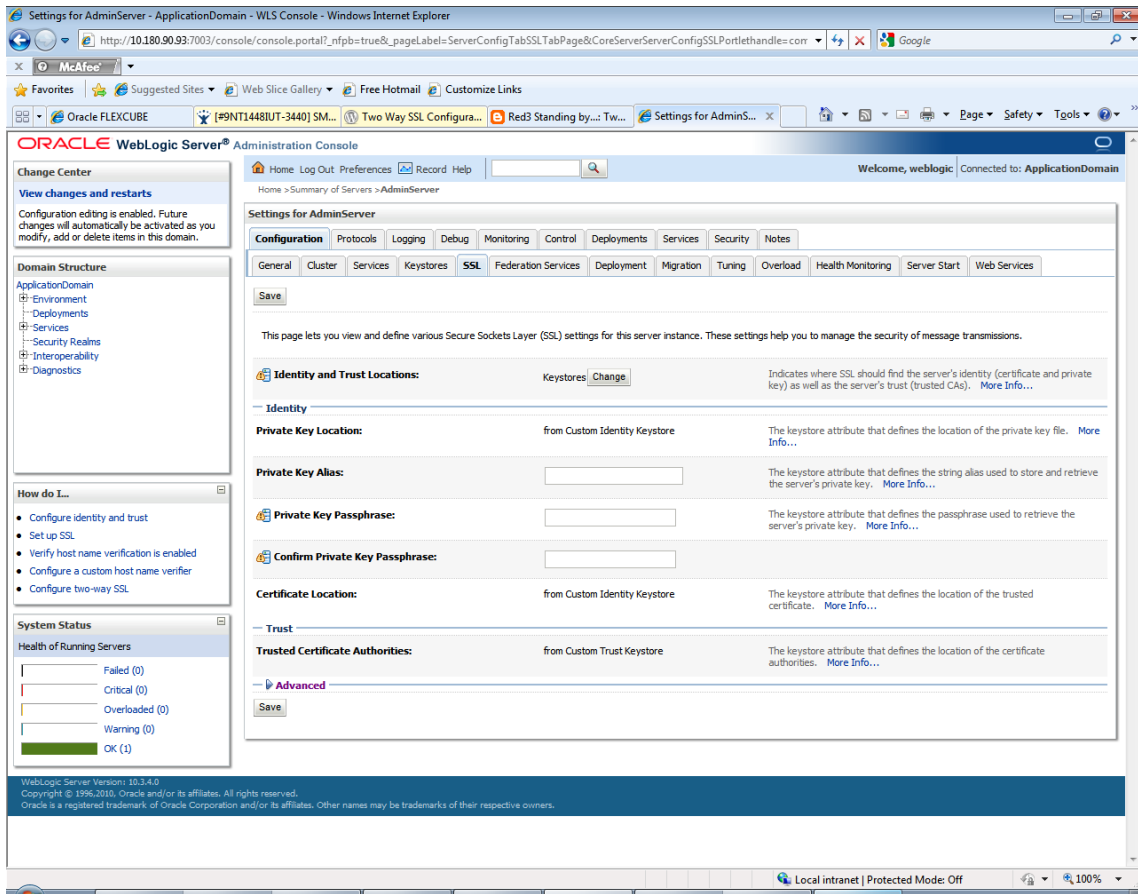
5. Enter the following details in the **Identity** and **Trust** sections:

Table 3–1 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 3–7 SSL



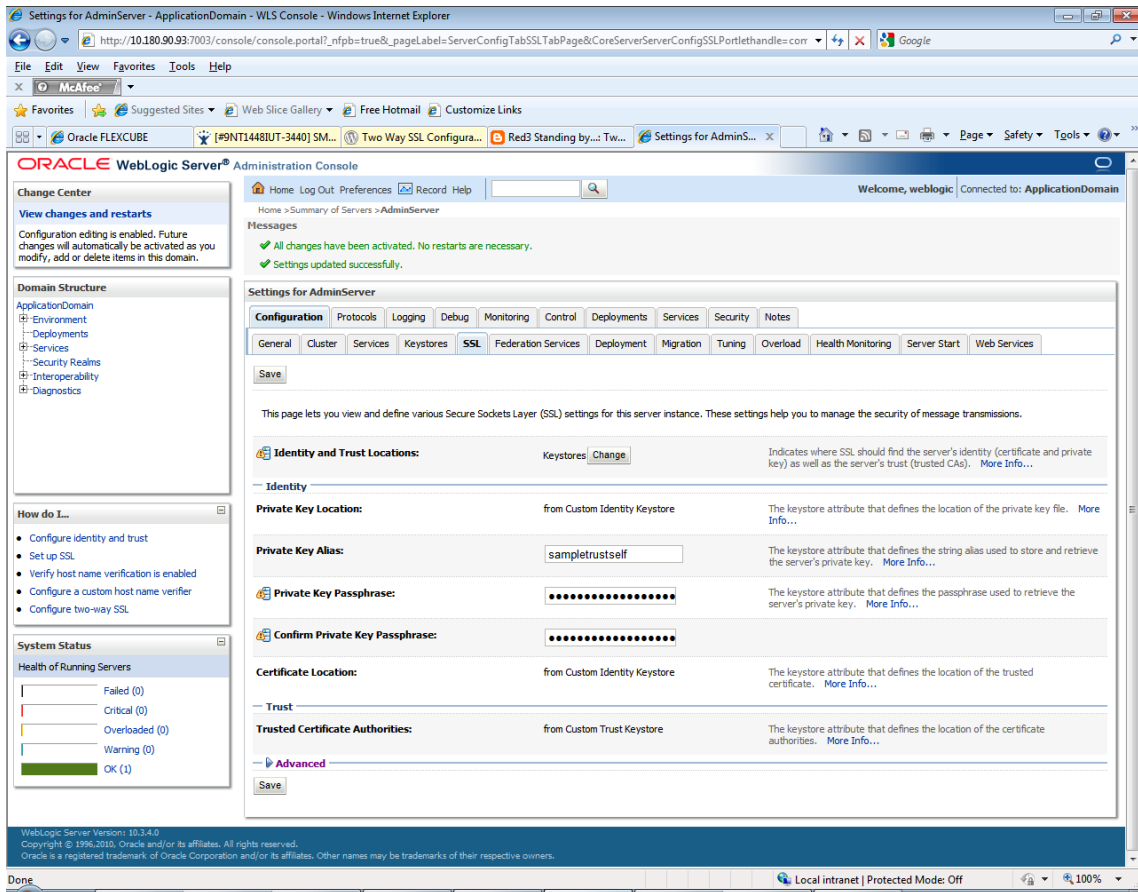
9. Enter the following details in the **Identity** section:

Table 3–2 SSL Configuration

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

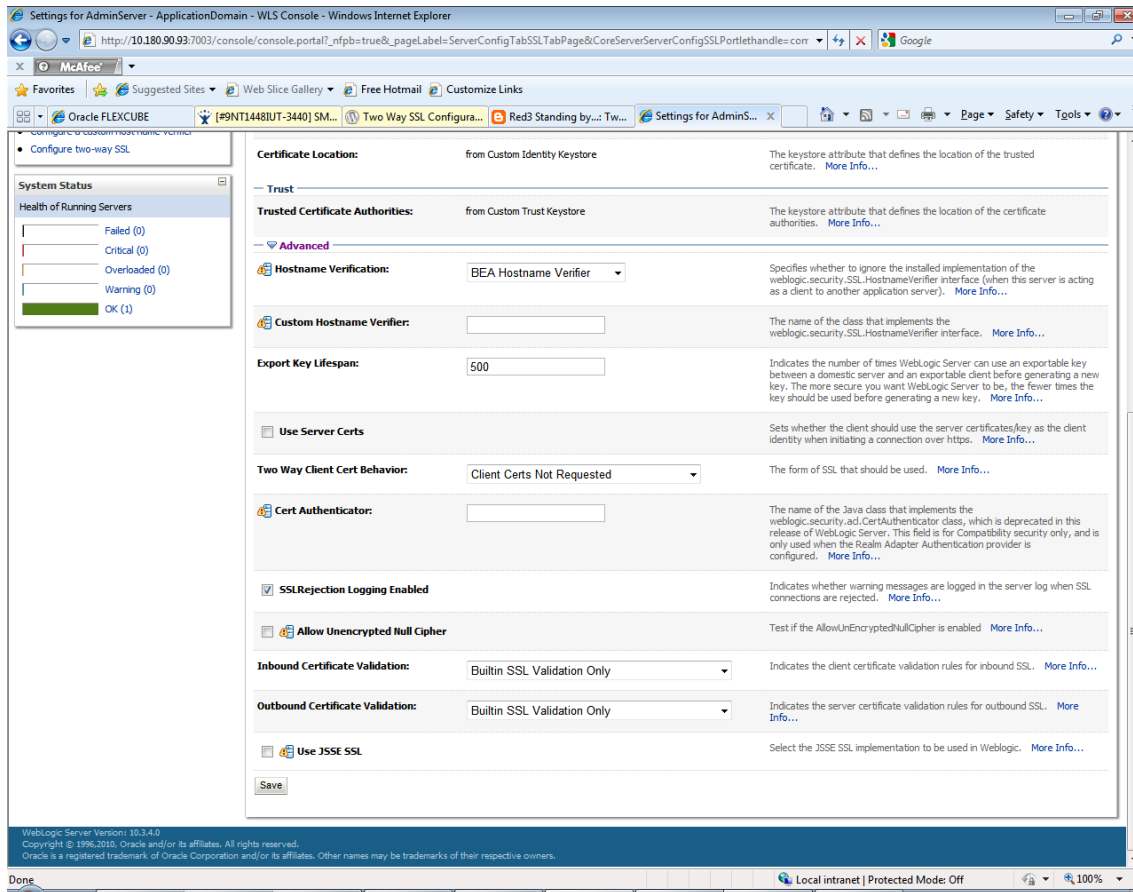
10. Enter the passphrases that were used while creating the certificate.

Figure 3–8 SSL Configuration



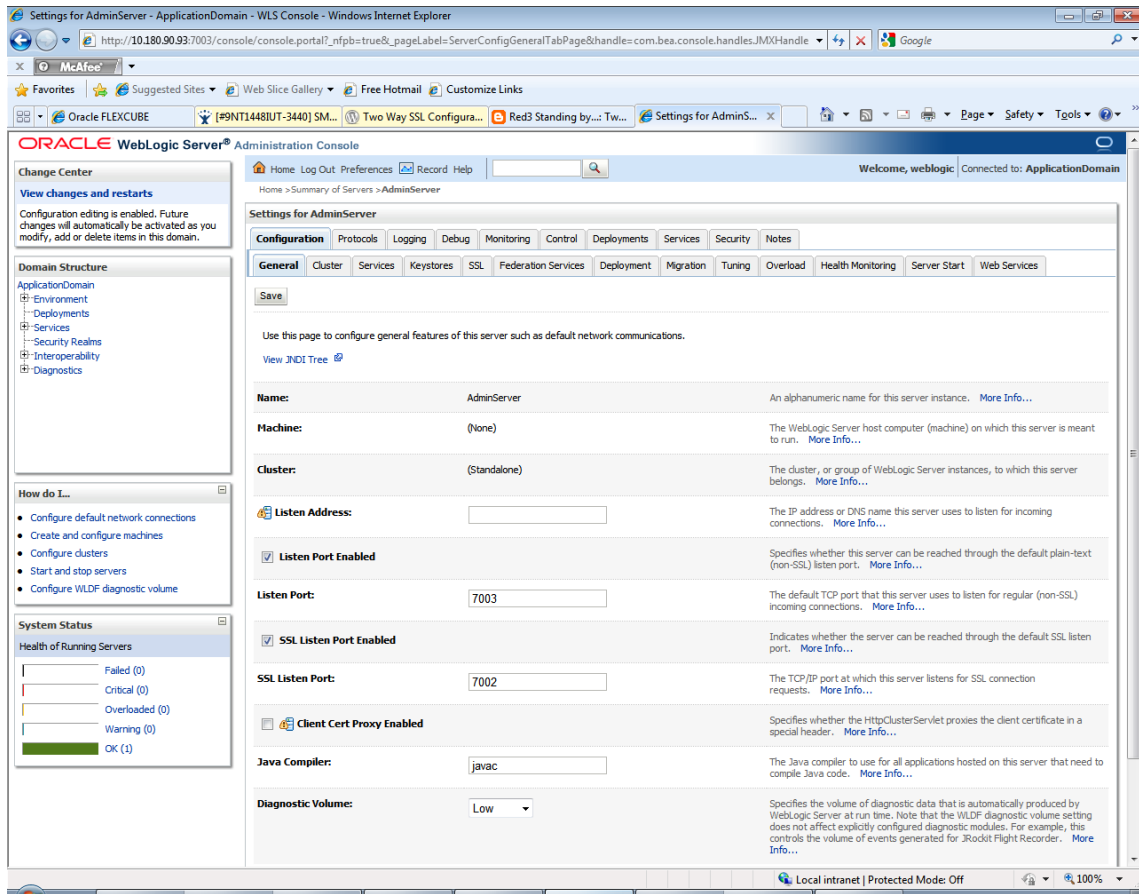
11. Click the **Save** button.
12. Click the **Advanced** link. Ensure that **Two Way Client Cert Behavior** is set to **Client Certs Not Requested**.

Figure 3–9 SSL - Advanced



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

Figure 3–10 General



15. Click the **Save** button.

Step 3 Restart Admin Server

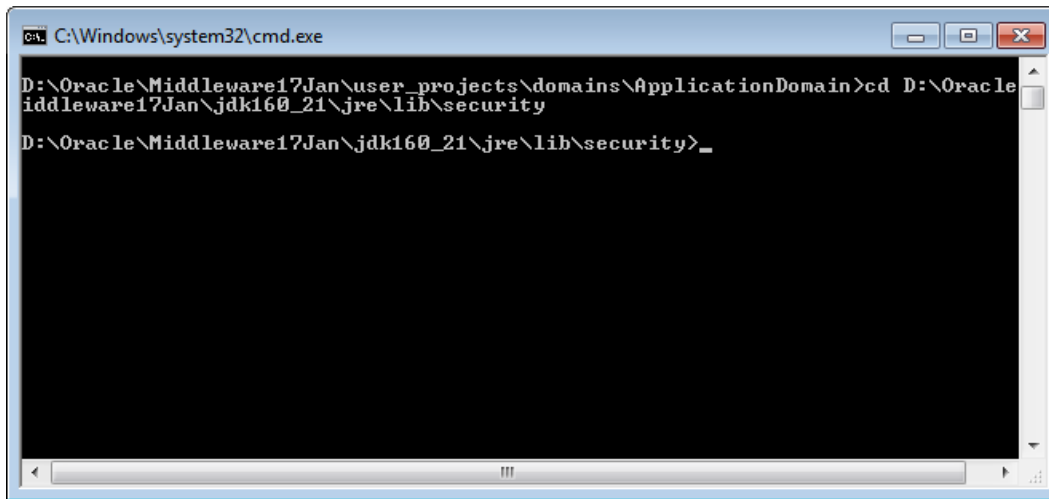
Restart the admin server of the Application Domain. Check the log file of admin server to ensure successful loading of the SSL configuration.

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 3–11 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. 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.

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

Step 5 Web Services Authentication 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 entries are 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.

OR

```
com.ofss.fc.app.party.service.core.MDMPartyApplicationService=policy:oracle/http_saml20_token_bearer_
service_policy
```


SAML Token Strategy for Third Party Applications

The following sample enumerates one of the SAML token specifications that third party applications can use:

Note

The signature, certificate, digest and other encryption related values are changed.

```
<saml:Assertion Version="2.0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="SAML-
nW7XYMp231fHjvTtM0JxFA22" IssueInstant="2016-08-
25T14:40:34Z"><saml:Issuer>www.oracle.com</saml:Issuer><dsig:Signa
ture
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"><dsig:SignedInfo><
dsig:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" /><dsig:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1" /><dsig:Reference URI="#SAML-
nW7XYMp231fHjvTtM0JxFA22"><dsig:Transforms><dsig:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" /><dsig:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" /></dsig:Transforms><dsig:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" /><dsig:DigestVa
lue>vKTL+kQYWTadssdsdxl4dt6kXvc=</dsig:DigestValue></dsig:Referenc
e></dsig:SignedInfo><dsig:SignatureValue>Wdsdadasdsa8addasdaadasda
daddasdasdasdadasdadasdadsdsdds1hmg1f0s98kfLtfREOpRRGn4xNO2z/Ju+KC
TtA5Y4E0ZuHZN5DF2no2mXwTOVZR00moTR1t5woUfi62iXnLLky+UTpVW5boi3QXdt
qsMI6oscbkgbrrigx5SMbJiR+kNni7vpg7UB2EBI5nLTGsRu4+383zggK5ETWRCV9
O7Zp/it5m0KuY0XctLEDAlSuM4069xrJgviMvuH9F3dgMjN/Dwy2pMr3VRsQ5gkMyY
IRNJ0vr4DzilckTSORU3chXja7CQDxjGm44mX84yL7OuRaRwfOql8HaA==</dsig:S
ignatureValue><dsig:KeyInfo><dsig:X509Data><dsig:X509Certificate>M
IIC7TCCAdWgAwIBdasdasdasdasdasdadadasdasdasdasdadSHxUe75mI51BSbDim
hMz4TprGhxG7jKDsthcnlWqxlCtJPgZeSR76HI/JGYIqozccKk303Dnc9y1YfqV73v
A/o2opXjzNSBC33ruovq9SiZz4F7v8clmp9wChI6V4AcC00jpb8</dsig:X509Certi
ficate><dsig:X509IssuerSerial><dsig:X509IssuerName>CN=orakey,
O="oracle
C=us"</dsig:X509IssuerName><dsig:X509SerialNumber>473970469</dsig:
X509SerialNumber></dsig:X509IssuerSerial><dsig:X509SubjectName>CN=
orakey, O="oracle
C=us"</dsig:X509SubjectName><dsig:X509SKI>SG/lnWm3TKwkxoW6KmkBPUyE
0C4=</dsig:X509SKI></dsig:X509Data></dsig:KeyInfo></dsig:Signature
><saml:Subject><saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">nikhilt</saml:NameID><saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer" /></saml:Subject><sam
l:Conditions NotBefore="2016-08-25T14:40:34Z" NotOnOrAfter="2016-
```

```
08-29T02:00:34Z"/><saml:AuthnStatement AuthnInstant="2016-08-25T14:40:34Z"><saml:AuthnContext><saml:AuthnContextClassRef>urn:oa
sis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassR
ef></saml:AuthnContext></saml:AuthnStatement></saml:Assertion>
```

This XML token is required to be compressed and Base64 encoded and pre-pended with the following string "oit ". This token is added to the HTTP header attribute 'Authorization'. The final token looks as follows:

Note

The actual string so generated is shortened.

```
oit
H4sIAAAAAAAAAAJ1XWZOiyhr8Kx2eR28Pu4gxdgSrIIvNIqJviMUim0KByK+/2O30sW
d6JubeJ6is/LKyFsr0e+3n2Yyta1DBpCyeXFDVw3M+wr+ho6cuz4p6dqPMR01VzEq/
TupZ4eegnsFgZrO6NhuIM/9H/ehJEeajG/5cbGhVqHyLLyx/TezALbrwBfh0e17I63
Of+pdQXfZ+wj218oD/+Hbz8FwaCLGtuDAAA
```

Some of the key fields in the XML token are enumerated below:

Table 3–3 Key fields in the XML token

XML-tag / attribute	Description
<saml:Issuer>	Default value 'www.oracle.com'
<dsig:DigestValue>	Digest is computed for the entire token, minus the Signature node.
<dsig:SignatureValue>	Signature is calculated for the entire token, with the digest value also being signed.
<dsig:X509Certificate>	The public key to be used in signature verification.

Step 6 Web Service SSL configuration

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
 - Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2

Note

If you do not specify the above property, it takes SSLv3 by default.

3.6 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.

4 Security Features

This chapter outlines the specific security mechanisms offered by Oracle Banking Deposits and Lines of Credit Servicing.

4.1 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 Deposits and Lines of Credit Servicing Security Architecture is explained in detail in the next section.

4.2 Security Architecture

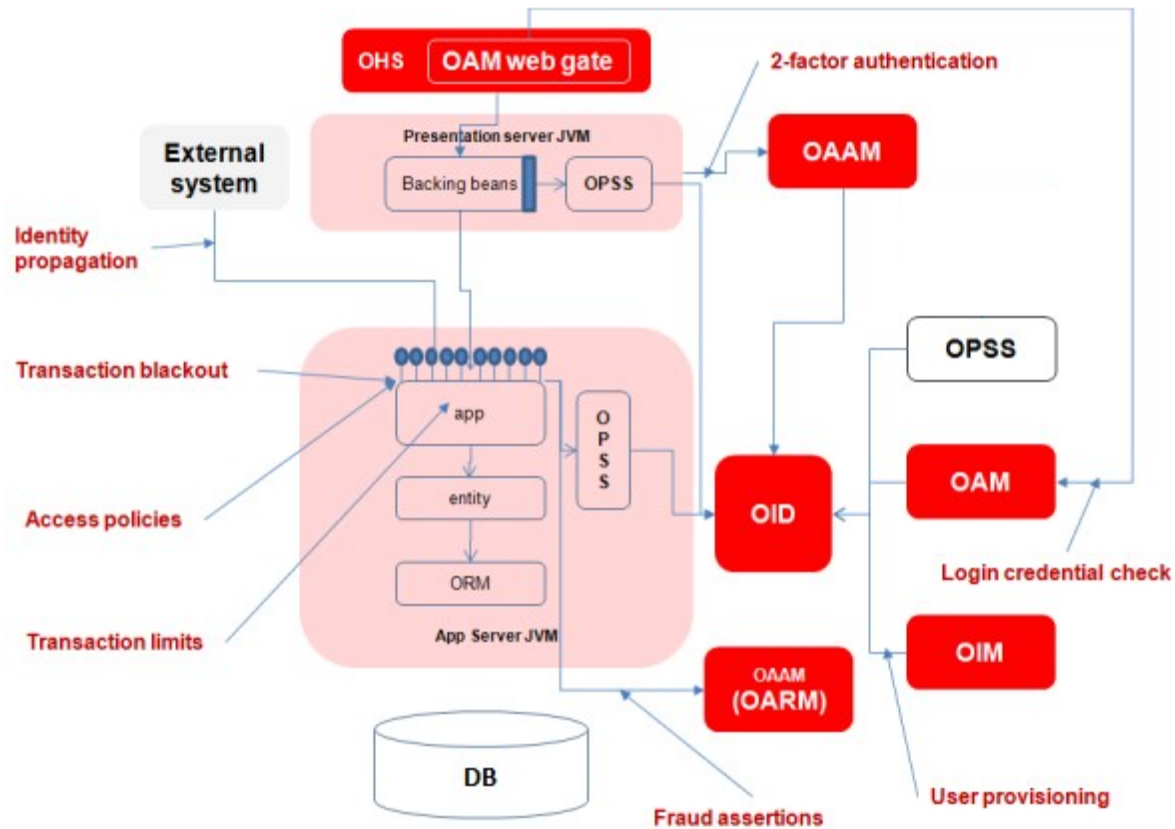
Oracle Banking Deposits and Lines of Credit Servicing comprises of several modules that interface with various systems in an enterprise to transfer or share data. This data is generated during business activity that takes place during teller operations or processing. While managing the transactions that are within OBDLOCS'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.

The security is provided in the product using either OIM or the Local Security option. The following sections explain the high-level architecture of OIM Based Security and Local Security.

4.2.1 OIM Based Security

The following diagram shows the high-level architecture of OIM Based Security.

Figure 4–1 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.
- 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.

4.2.2 Local Security

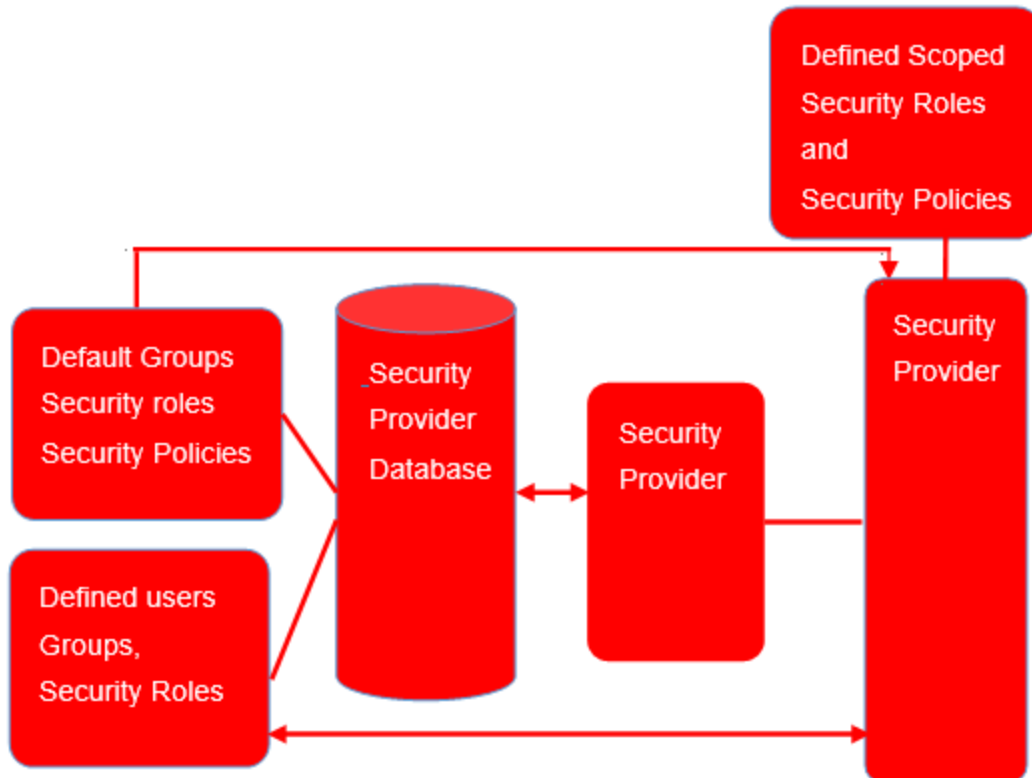
The Local Security option is provided using Oracle database as follows:

- SQLAuthenticator is configured in weblogic security realm.
- Users, groups, and roles are stored in the database.

- User, role, and entitlement management are done using the internal screens developed as a part of local Security module.

The following diagram shows the high-level architecture of Local Security.

Figure 4–2 Local Security – High Level Architecture

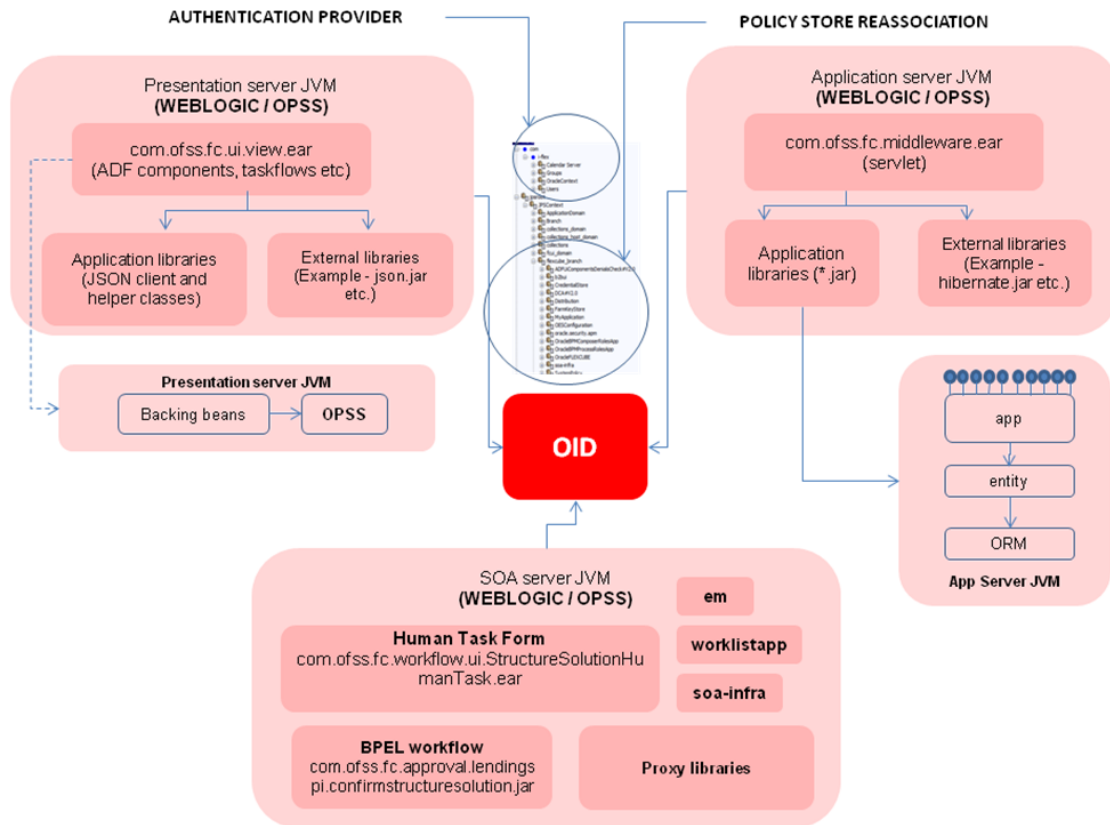


4.3 Approvals Architecture

Oracle Banking Deposits and Lines of Credit Servicing 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 via 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 via 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 4–3 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:

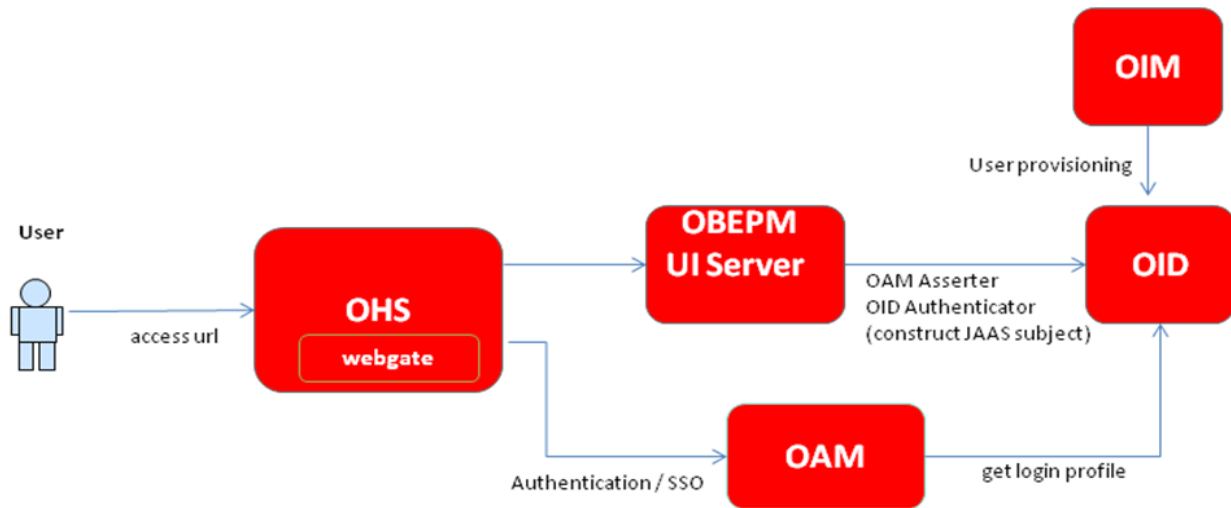
- **Dual Control:** Any transaction can be set up for approvals (2-eyes principle).
- **User Limits:** User Limits assenter evaluates whether transaction amount is within limits available to the user (role).
- **Matrix Based:** Matrix assenter evaluates a matrix of facts available in the context of the transaction. This assenter 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.

4.4 Configuring and Using Authentication

Oracle Banking Deposits and Lines of Credit Servicing uses OAM to authenticate users.

Figure 4–4 Authentication and Single Sign On



Data flow is as follows:

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

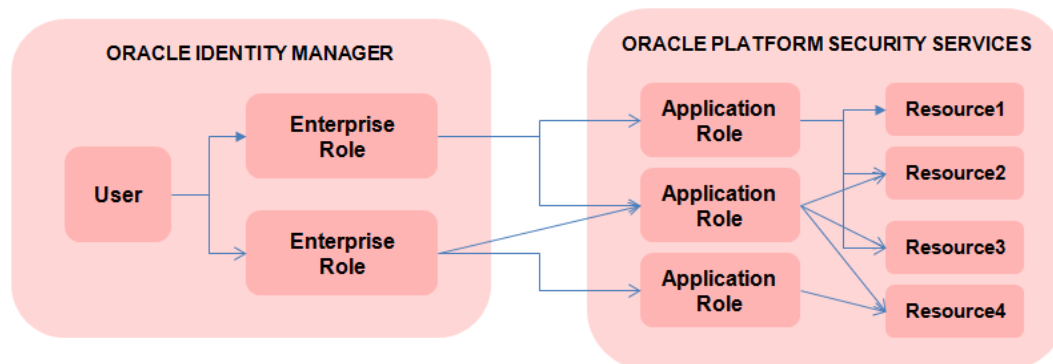
4.5 Configuring and Using Access Control

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 Deposits and Lines of Credit Servicing uses OPSS Entitlements for authorization.

Figure 4–5 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.

The application roles are mapped to the enterprise roles (or OID groups) that are managed within the identity store. The association between enterprise role and application role is many-to-many.

Oracle Banking Deposits and Lines of Credit Servicing' security solution implements the following factory-shipped roles:

Table 4-1 Factory-Shipped Roles

Role	Description
OBP - read only	This is a read-only role that has inquiry privileges for a number of screens. For screens that have capabilities to change data, these users will have access only to inquiry services.
System Administrator	This role has privileges to perform and approve configuration parameters. The configuration is mostly done during implementation such as technical configuration and server configurations such as managing resource adapters.
Loans Officer	This role has privileges to evaluate, authorize, or recommend approval of commercial, real estate, or credit loans. This user advises borrowers on financial status and methods of payments. Entitlements include loans servicing.
Broker	This role is a customer-facing role involved in selling products to customer. All party-related inquiries and origination capabilities are given to these set of users. Entitlements include party identification, account details, party relationship inquiries, and so on.
Customer	This role provides entitlements for internet banking capabilities. Entitlements include funds transfer, installment payment, partial payoff, and so on.
Product Manager	This role provides entitlements for defining products and offers. Entitlements include offers and products creation.
Due Diligence Officer	This role provides entitlements for performing Know Your Customer (KYC), due diligence, onboarding or recertification process. Entitlements include collecting, analyzing, verifying and archiving legal and KYC documentation.

Industry experience feeds the OBDLOCS security model and the entitlements thereof. The latest set of factory-shipped access policies is available in the host mediapack that can be downloaded from e-delivery. These policies are seeded by the Policy-Store setup utility (PIT) during installation. At run time, these are managed using Authorization Policy Manager (APM), which is the GUI of the Oracle Entitlements Server (OES).

4.6 Configuring and Using Security Audit

Oracle Banking Deposits and Lines of Credit Servicing 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.

4.7 Configuring and Using TDE

Oracle Banking Deposits and Lines of Credit Servicing relies on Oracle® Database Advanced Security for encrypting sensitive data.

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

OBDLOCS 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
'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcmdp' -auto_login
```

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

```
orapki wallet display -wallet /oracle/app/admin/IN5FMT/wallet
-pwd 'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcmdp'
ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY
'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcmdp";
```

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');
```

4.8 Securing Outbound Interactions

Oracle Banking Deposits and Lines of Credit Servicing interacts with external systems like Oracle Analytics Publisher (formerly know as Business Intelligence Publisher), 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 Oracle Analytics Publisher,

```
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.

4.9 Securing Key Store

This section describes the securing key store details.

4.9.1 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.

4.9.2 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.

4.9.3 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.

5 Data Privacy and Security

This chapter explains the data privacy and security features offered by Oracle Banking Deposits and Lines of Credit Servicing.

5.1 Data Minimization

The primary use cases depicting Personally Identifiable Information (PII) data flows are presented in the following diagrams:

Figure 5–1 Bank Admin as PII Data Originator

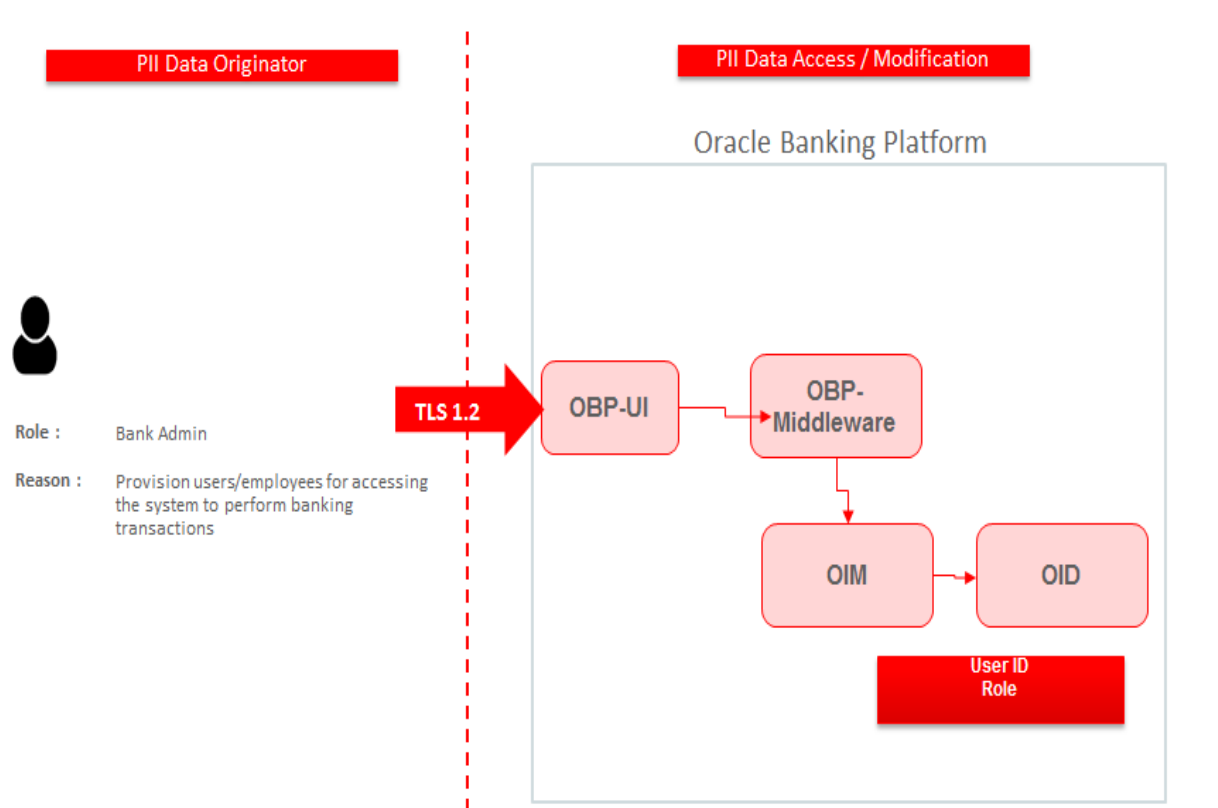


Figure 5–2 Bank Teller as PII Data Originator from Application Form

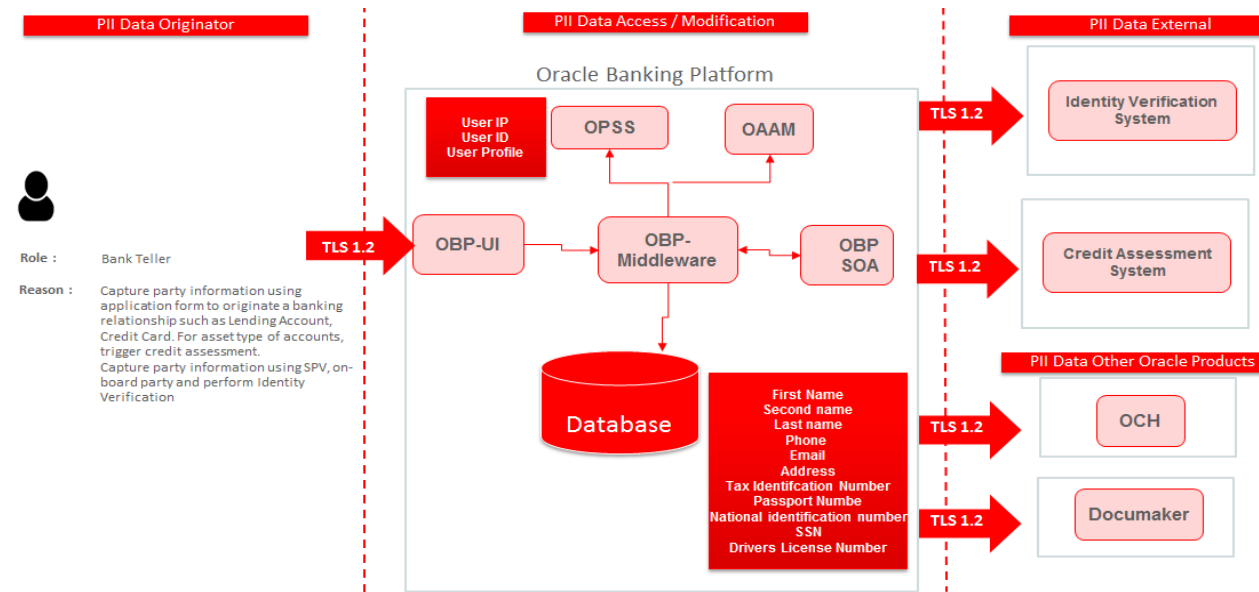
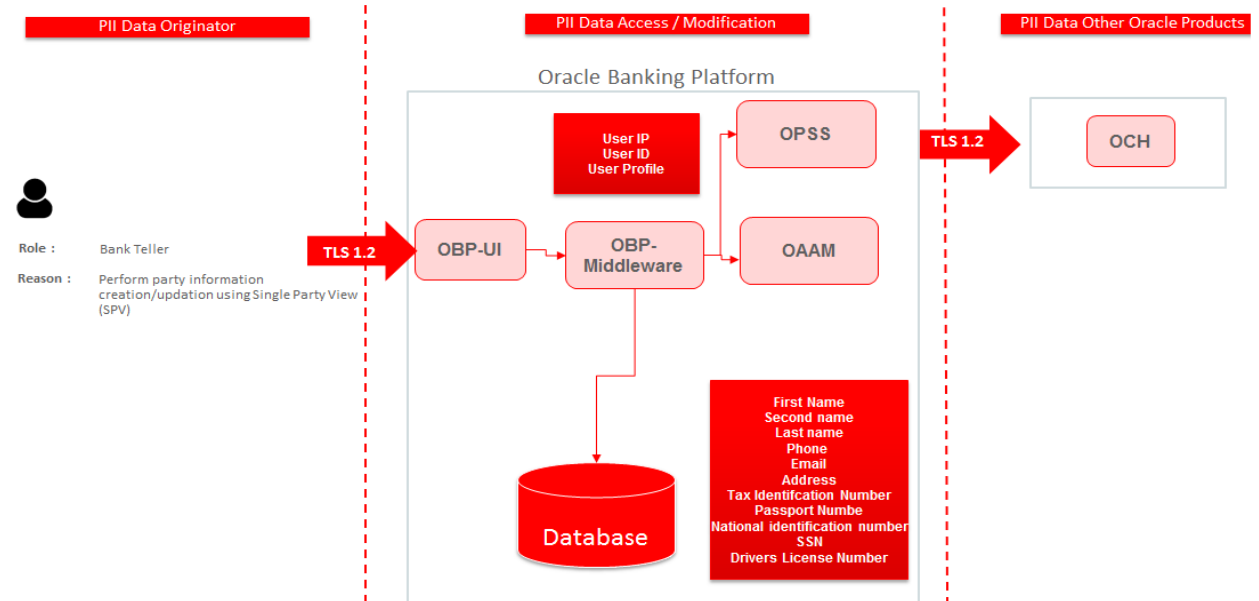


Figure 5–3 Bank Teller as PII Data Originator from Single Party View



5.2 Data Portability

Oracle Banking Deposits and Lines of Credit Servicing enables bank users to extract the audit log in an industry standard, so that the file can be provided to a customer or system in a machine-readable format for easy interpretation.

- Interface Logging (Fast Path: OPA008): The user can use this screen for viewing and extracting log of payloads to external interfaces.

- Audit Text Based Search (Fast Path: BAM56): The user can use this screen for viewing and extracting log of entity maintenance in OBDLOCS.

Both Interface Logging (Fast Path: OPA008) and Audit Text Based Search (Fast Path: BAM56) allow the user to do criteria based search. The search results can be viewed on the screen as well as exported as Excel file format and further saved as a CSV (comma separated values) file format. As PII details are not part of transactions, no enhancements are required in Financial Transactions Log View.

The Party Export Data service helps in extracting data for the parties mentioned in the request. The request is logged and during End of Day batch process, the data is extracted and stored in machine readable formats.

5.3 Encryption

This section explains about encryption of PII data.

5.3.1 Key Management

Oracle Banking Deposits and Lines of Credit Servicing encrypts and decrypts PII data using AES. Since it uses symmetric-key algorithm, key management is very critical.

The starting point in any private key management strategy is to create a comprehensive inventory of all keys, their locations and responsible parties. Private keys used must be kept secure as unauthorized individuals can intercept confidential communications or gain unauthorized access to critical systems. Failure to ensure proper segregation of duties means that administrator who generates the encryption keys can use them to access sensitive, regulated data.

Oracle Banking Deposits and Lines of Credit Servicing, by default, implements secure storage and access to encryption key.

5.3.2 Secure Storage of Encryption Key

Java Key Store (JKS) is used to hold the encryption key. JKS file is created for each encryption key (For example, for card number encryption, a separate JKS file is created). The key store file, type and corresponding mapping properties are factory shipped with product jar.

Following are the Java key store parameters that are used:

Table 5–1 Java Key Store Parameters

Parameter	Value
Type	Secret Key
Algorithm	AES
Store Type	JCEKS (Triple DES)
Key Size	128
Alias	<<alias>>
Key Password	<<password>>
Store Password	<<password>>
Domain Name	<<domain-name>>
Key Store	<<key store file name>> For example, cks-keystore.jceks for card number

5.3.3 Secure Access of Encryption Key

For accessing the encryption key, the JKS requires valid alias and password. The alias and password are maintained using credential store resource adapter (com.ofss.fc.connector). Connector Instance is created for each encryption key. For example, JNDI Name: ra/FCRJConnectorKEYSTORE_CARD, for card number.

Credential mapping should be done for each JNDI / encryption key as follows:

Table 5–2 Encryption Key Parameters

Property	Value	Mapping / Usage
EIS User	<<alias>>	Alias used for key store
EIS Password	<<password>>	Store / Key password used for the key store

The credential store JNDI name is maintained in the configuration factory (DB based). The property ID has the key lookup name.

Table 5–3 Encryption Key Parameters

Configuration Type	Category	Prop ID	Prop Value
DB Based	CredentialConnector	CKS_RA_JNDIKEY (Format: <<keyLookupName>>_RA_JNDIKEY)	a/FCRJConnectorKEYSTORE_CARD

5.4 Tracking Technologies

OBDLOCS components have externalized their authentication needs to the Oracle security stack. The applications in the OBDLOCS suite do not generate, manipulate, collect or interpret cookies. However, the underlying weblogic and OPSS infrastructure on which OBDLOCS is deployed does use cookies for its authentication needs.

5.5 Separate Auditing and Detective Control Privileges

This section explains about auditing and detective control privileges.

5.5.1 Application Logs

Following application logs are supported in OBDLOCS:

- Financial Transactions
- Entity Maintenance
- Payloads to External Interfaces

PII data for logs are masked prior to logging.

5.6 Logging

This section explains about logging.

5.6.1 Application Logs

Following application logs are supported in OBDLOCS:

- Financial Transactions
- Entity Maintenance
- Payloads to External Interfaces

PII data for logs are masked prior to logging.

Appendix

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure Oracle Banking Deposits and Lines of Credit Servicing.

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.