Oracle® Revenue Management and Billing Cloud Services

Release 8

Federated Identity Configuration Guide

Revision 2.0

F31148-01

May, 2020

ORACLE

Oracle Revenue Management and Billing Cloud Services Federated Identity Configuration Guide

F31148-01

Hazardous Applications Notice

This software 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 in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software and documentation may provide access to or information on 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.

Preface

About This Document

This document provides details of ORMB cloud federated identity to configure SSO (Single Sign On) with OpenID/OAuth Connect Protocol. This document will help you to understand how to configure federated identity with Identity Cloud Service (IDCS) for Oracle Revenue Management and Billing (ORMB) cloud service.

Intended Audience

This document is intended for customers using Oracle Revenue Management and Billing (ORMB) cloud service and assumes that you have administrative privileges on the host where you want to install the software.

Contents

1. Federated Single Sign On With SAML 2.0 1
1.1 OpenID Connect Terminology 1
1.2 Federated SSO Login Overview 1
1.2.1 Option 1 2
1.2.2 Option 2 3
1.3 OpenID Connect Implementation 3
1.4 SAML 2.0 Implementation (Only for Option1) 4
1.4.1 Configure SAML 2.0 Compliant Identity Provider 4
1.4.2 SAML Metadata 4
1.4.3 User Provisioning 5
1.5 Why SAML? 5

1. Federated Single Sign On With SAML 2.0

Federated single sign-on (SSO) standards such as OpenID Connect (OIDC) provide secure mechanisms for passing credentials and related information between different web applications that have their own authorization and authentication systems. OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol. It allows clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner. OpenID Connect has become the leading standard for single sign-on and identity provision on the Internet. Its formula for success: simple JSON-based identity tokens (JWT), delivered via OAuth 2.0 flows designed for web, browser-based and native, mobile applications.

ORMB uses 'Authorization code' flow. This flow is the most commonly used flow, intended for traditional web apps. Involves an initial browser redirection to / from the OpenID Provider/Identity Provider for user authentication and consent, then a second back-channel request to retrieve the ID token. This flow offers optimal security, as tokens are not revealed to the browser and the client can be authenticated.

ORMB provides two type of configuration option:

- Customer can integrate IDCS (as SP) with their on-premise Identity provider using SAML protocol
- Customer can leverage IDCS as their identity provider

1.1 OpenID Connect Terminology

The SAML 2.0 specification provides a Web Browser SSO Profile, which describes how web applications can achieve Single Sign On. Following are the main players in OpenID Connect:

- **Client** - This is how the user is interacting with the Resource Server, like a web application being served through a web browser.
- **Identity Provider (Authorization Server)** – This server owns the user identities and credentials, and authenticates the user.
- **SAML token** - The term SAML token refers to SAML Assertion, often compressed, encoded, possibly encrypted. SAML Assertion is just an XML node with certain elements.
- **Metadata*:*** Metadata defines how SAML 2.0 shares configuration information between two communicating entities. You can access and share the Access Manager Metadata information with the federated application. You can also access and share the federated application metadata with Access Manager.

1.2 Federated SSO Login Overview

With federated login, an external Identity provider (IDP), such as an on premise corporate login system, is used to authenticate the user's Id and password and, if successful, a token (SAML assertion) is generated by the IDP and used to grant access to the target application.

1.2.1 Option 1

The login process is as follows:

1. User accesses the ORMB application through the OHS/Webgate URL.
2. Webgate on ORMB app server intercepts the request and identifies the user is not authenticated.
3. Webgate redirects the user to the configured IDCS.
4. An external identity provider as configured in IDCS should do the authentication. It creates a SAML 2.0 request and responds to the browser with a redirect to the IDP.
5. The IDP is invoked with the SAML request and the IDP challenges the user with a login prompt.
6. The IDP authenticates the user and responds with a SAML 2.0 assertion and IDCS validates the assertion.
7. IDCS server accepts the resource consent and gets an 'Authorization Code'. IDCS internally sends authorization code at the ‘redirect_url’ specified.
8. Webgate intercepts the authorization code and a HTTP POST request is sent to IDCS to receive an 'Access Token'. This request is performed at the back channel and is not redirected from the browser.
9. IDCS generates an access token and returns to WebGate in the HTTP response.
10. WebGate redirects the user back to the originally requested resource.
11. Weblogic validates the JWT assertion token and then redirects to the ORMB application.

Refer to the image below to for better understanding of Screen Login flow:

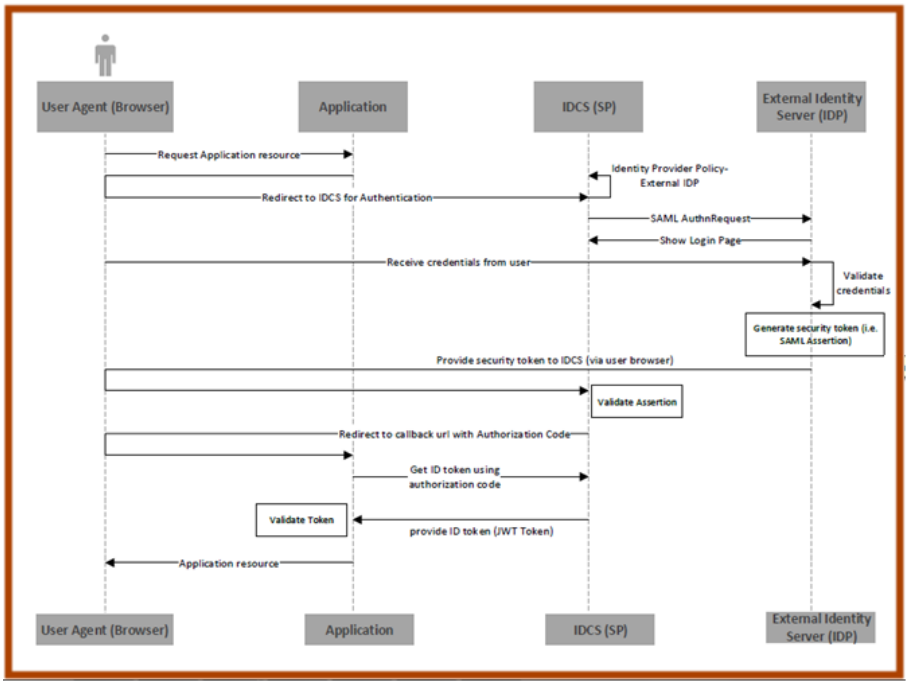

Fig 1: Screen Login Flow

1.2.2 Option 2

The login process is as follows:

1. User accesses the ORMB application through the OHS/Webgate URL.
2. Webgate on ORMB app server intercepts the request and identifies the user is not authenticated.
3. Webgate redirects the user to the configured IDCS. IDCS challenges the user to enter the credentials.
4. The user logs in to the IDCS server accepts the resource consent and gets an 'Authorization Code'. IDCS internally sends authorization code at the ‘redirect_url’ specified.
5. Webgate intercepts the authorization code and a HTTP POST request is sent to IDCS to receive an 'Access Token'. This request is performed at the back channel and is not redirected from the browser.
6. IDCS generates an access token and returns to WebGate in the HTTP response.
7. WebGate redirects the user back to the originally requested resource.
8. Weblogic validates the JWT assertion token and then redirects to the ORMB application.

Refer to the image below to for better understanding of Login flow:

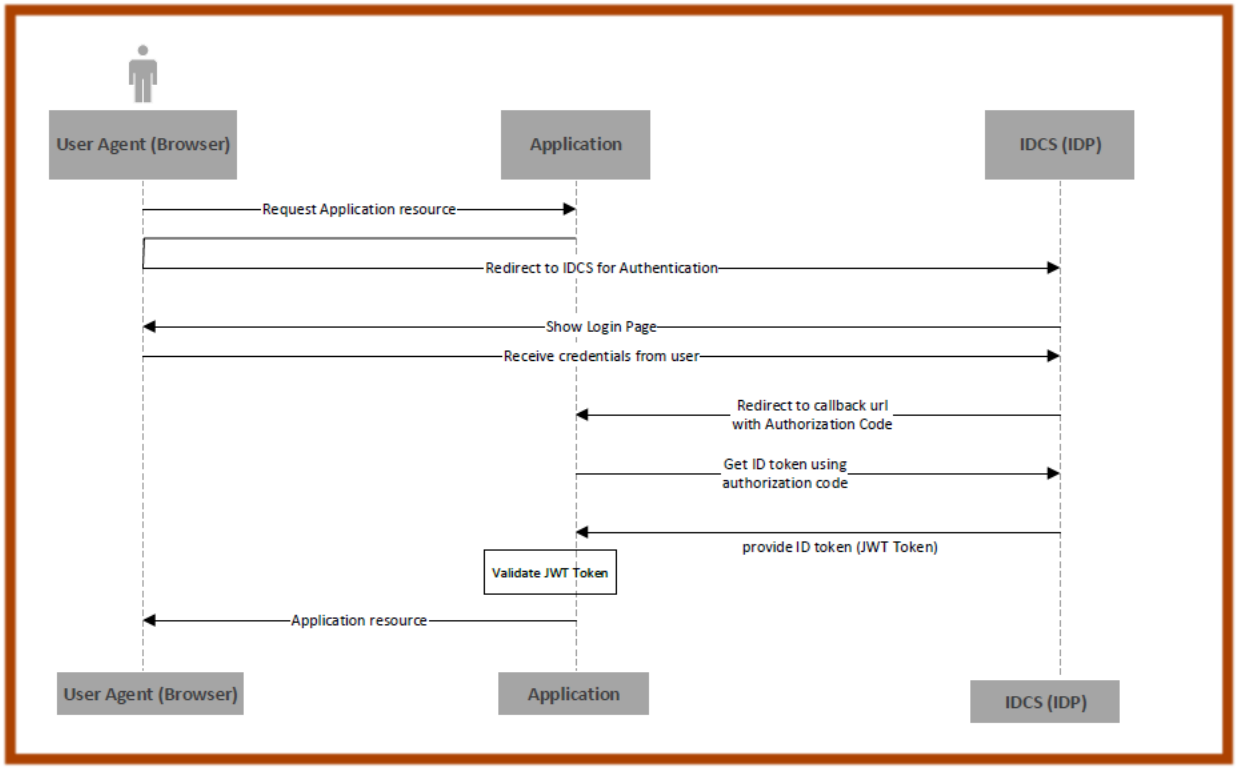

Fig 2: Screen Login Flow

1.3 OpenID Connect Implementation

IDCS OAuth key used for signing the JWT Access Token is imported in ORMB App Server (Weblogic) using chef automation script. These exchanges of signing keys will happen over sftp or email. ORMB application server in order to validate JWT Access Token will use JWT Identity Assertor using JWT signing certificate.

1.4 SAML 2.0 Implementation (Only for Option1)

External Identity Provider (IDP) will handle the sign-in process and will eventually provide the authentication to ORMB application users. Users are authenticated through SAML Assertion at IDCS ends. Any changes you perform on Premise accounts (namely first name, last name, and email) is synced back to the ORMB account through external REST services. The only user data necessary for ORMB is a user id for each user, the user's first name, last name and email. ORMB does not store passwords.

1.4.1 Configure SAML 2.0 Compliant Identity Provider

This section contains guidelines on how to configure SAML 2.0 Identity Provider to federate with ORMB application server to enable Single Sign-On access to one or more ORMB cloud services using the OpenID Connect protocol. The SAML 2.0 relying party for ORMB cloud service used in this scenario is External IDP and service provider is IDCS.

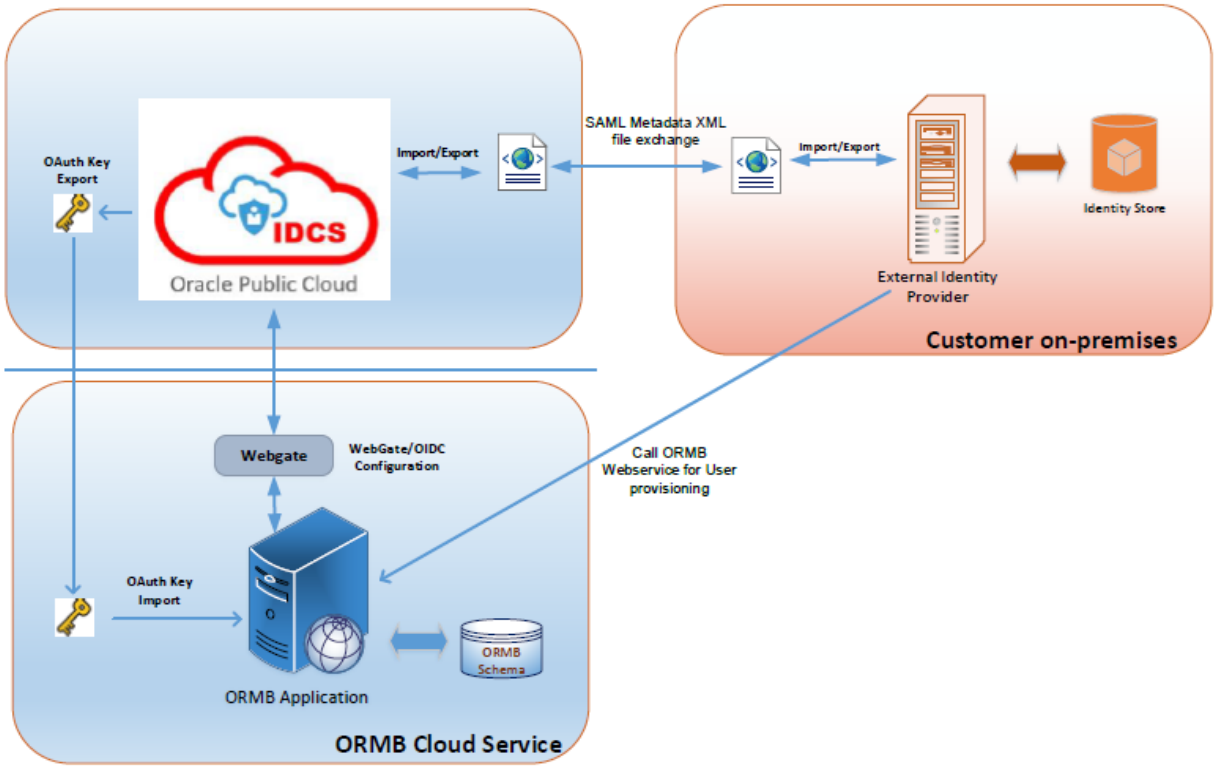

Fig2: SAML Configuration

1.4.2 SAML Metadata

IDP imports IDCS's SAML metadata and thereby exchange public keys, IP addresses and communication information. Thus, ORMB IDCS provides you with the SAML metadata XML file, including the correct X509 certificates. It is recommended that you always import the latest ORMB metadata when configuring SAML 2.0 identity provider.

The following image shows a sample SAML2.0 metadata XML:

Fig 3: Sample SAML Metadata XML

Please note that the metadata XML varies from server to server.

1.4.3 User Provisioning

For user provisioning, external identity server must compatible with JWT. Customer needs to create users into ORMB application through REST services. For detailed instructions on how to do this, refer to the document: R7_REST_Services_Federated_Identity_Configuration.doc. User must be present in ORMB application.

1.5 Why SAML?

The benefits of SAML include:

- **Platform neutrality:** SAML abstracts the security framework away from platform architectures and particular vendor implementations. Making security more independent of application logic is an important principle of Service-Oriented Architecture.
- **Loose coupling of directories**: SAML does not require user information to be maintained and synchronized between directories.
- **Improved online experience for end users:** SAML enables Single Sign-On by allowing users to authenticate at an Identity Provider and then access service providers without additional authentication. Additionally, identity federation (linking of multiple identities) with SAML allows for a better-customized user experience at each service while promoting privacy.

