covisint®

Enabling information ecosystems.

appcloud™

Federation Integration Guide

April 24, 2013

Table of Contents

About This Guide

Purpose / Background

The purpose of this document is to describe what is required by an Application Provider to be able to establish a federation connection with AppCloud™ for the purpose of making an application or applications available to users of Covisint hosted Sponsor Portals via Single Sign-On (SSO). This document is a supplement to the *SSO and Application Provisioning Integration Guide* and assumes that the reader is familiar with that guide (definitions for acronyms and abbreviations used in this guide are contained there).

Upon completion of this document the reader will have established a base understanding of the key technical requirements and decisions that an Application Provider must make to be able to initiate a federation integration with AppCloud.

Audience

IT personnel who will be responsible for completing the tasks required to support the integration process described in this guide. It is assumed that the reader is familiar with federation technologies.

Scope

Included Topics

- Logical Architecture
- Application Provider Requirements

Excluded Topics

- Enabling Federation at the Application Provider's site.
- Integration of the Application Provider's federation solution with their provisioning and web access management systems.

Overview

Logical Architecture and Basic Data Flow

The following section describes the logical architecture and basic data flow that is utilized to enable a user who has authenticated to a Sponsor's Portal, to click on a link to a secured AppCloud™ application in the Portal, and then gain access to the application which is hosted at the Application Provider's site. Using federation terminology, the Application Provider is the Service Provider (SP) and the application is the Target Service.

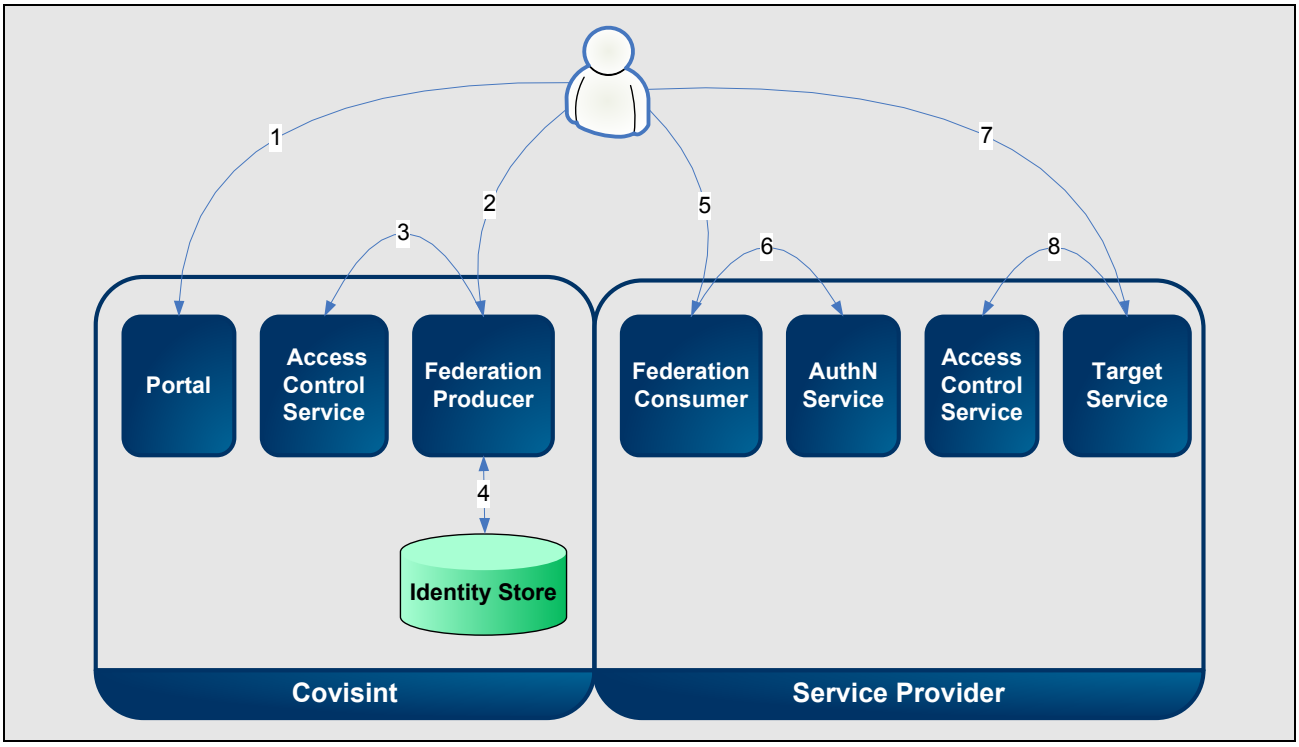

Figure 1: Logical Architecture and Basic Data Flow

1. A user who has already authenticated to a Sponsor's Portal selects a link to the SP's application (the Target Service).
2. The user's browser is redirected to the Federation Producer.
3. The Federation Producer makes a call to the Covisint Access Control Service which verifies the user's identity and then determines if the user has been granted access to the SP's application.
4. If the user has been granted access to the SP's application the Federation Producer makes a call to the Covisint Identity Store and gets the user's identity information which will be included in the federation assertion. It then generates the assertion (an assertion is an XML message that has a format which complies with an industry federation standard).
5. The assertion and the application URL are sent as form variables to the SP's Federation Consumer via an HTTP-POST. The Federation Consumer validates the assertion by verifying the associated digital signature.

6. The Federation Consumer makes a call to the SP's Authentication (AuthN) Service which provides the user's browser with the SP's standard security token (the security token is used in conjunction with the web access management system to protect the application using session security).
7. The user's browser is redirected to the application (using the application URL that was included in the original POST to the Federation Consumer).
8. The application makes a call to the Access Control Service which verifies the user's identity and then provides access to the application as appropriate.

Note: Steps 2 through 7 are all invisible to the user. From the user's perspective a link to the application is selected and access to the resource is provided.

Technology Assumptions

- SAML HTTP-POST Profile will be used for all federation assertions.
- The Service Provider is capable of consuming an AppCloud supplied assertion using one of the supported federation standards.
- The attributes included in the attribute section of the SAML assertion will consist of a set of standard attributes that will be provided to the Service Provider.
- All federation assertions created by AppCloud will include the SP's Target Service (application) URL in a form POST variable named Target or RelayState.
    - The purpose of the Target or RelayState URL is to instruct the SP's Federation Consumer where to send the user's browser after the SP's security token has been provided to the browser.

Requirements

Systems

To complete a standard integration with AppCloud™, The SP's Federation Consumer will need to support the ability to consume HTTP-POST Profile assertions (i.e. Ping Identity PingFederate, Sun Java System Access Manager, RSA FIM, HP OpenView Select Federation, and Shibboleth are examples of solutions that provide support for this). Once the assertion is validated, the SP's system will provide the user's browser with the standard security token that the SP uses to provide session security to the SP's applications. Generally speaking, solutions that provide support for consuming federation assertions can be implemented independent of any changes to the SP's application(s). AppCloud provides standard support for SAML 2.0, SAML 1.1 and WS-FED 1.1 and can also provide support for custom federation integrations which will be defined and priced on a per project basis.

If SP's application is or will be integrated with an existing Single Sign-On (SSO) infrastructure, the recommendation is to utilize that system's support for federation. For most Commercial-Off-The-Shelf (COTS) solutions, this will require little to no development effort and should support an implementation that is independent of any changes to the application.

If SSO has not already been planned or implemented in your environment, several stand alone products are available (i.e. Ping Identity PingFederate, RSA FIM, etc.) as well as open source APIs from Shibboleth at http://shibboleth.net/ that will provide the necessary functionality to integrate an application with AppCloud.

A technology/product decision will need to be made by the SP. The SP can choose any compatible solution preferred but the relationship between the SP and any chosen vendor(s) is the SP's responsibility including but not limited to any negotiations around licensing and/or implementation services.

Protocols

Covisint will work with the SP to establish and manage the federation connection between the SP and AppCloud™. Based on the protocol that will be used, the following information will be required from the SP to establish this connection. The configuration information will be provided and managed through the self-service features available in AppCloud.

SAML 2.0

The SP provides a SAML 2.0 meta-data file or the specific meta-data required for the protocol. SAML 2.0 defines a standard for meta-data files and most products that support SAML 2.0 have the ability to create a meta-data file.

If a meta-data file can not be generated the following information is required.

- Protocol Version: SAML 2.0
- SP Name: A Service Provider name that can be easily identified by the end user (this is a "user friendly" name).
- Entity ID: The Entity ID is the identifier that uniquely represents every SAML Service Provider.
- Consumer URL: URL that the assertion will be posted to.
- SAML assertion will be signed: Integrity of a message between providers is insured by using digital signatures. Covisint will provide the SP with the Public Key that will be used to verify the assertion signature.
- Does the SP require the response to be signed (Yes or No): Integrity of a message between providers is insured by using digital signatures. By enabling response signing, a digital signature is generated for the

SAML response document. If yes, Covisint will provide the SP with the Public Key that will be used to verify the response signature.
- Does the SP require the assertions to be encrypted (Yes or No): All assertions will be encrypted during transport using HTTPS. Additionally, if required the assertion itself can be encrypted. If this level of encryption is required, the SP will need to provide Covisint with the Public Key that will be used to encrypt the assertions.

SAML 1.1

The SP provides a SAML 1.1 meta-data file or the specific meta-data required for the protocol. Some products that support SAML 1.1 have the ability to create a meta-data file.

If a meta-data file can not be generated the following information is required.

- Protocol Version: SAML 1.1
- SP Name: A Service Provider name that can be easily identified by the end user (this is a "user friendly" name).
- Entity ID: The Entity ID is the identifier that uniquely represents every SAML Service Provider.
- Does the SP require the assertions to be signed (Yes or No): Integrity of a message between providers is insured by using digital signatures. By enabling assertion signing, a digital signature is generated for the SAML assertion document. If yes, Covisint will provide the SP with the Public Key that will be used to verify the assertion signature.
- SAML response will be signed: Integrity of a message between providers is insured by using digital signatures. By enabling response signature signing, the digital signature is generated for the SAML response document. Covisint will provide the SP with the Public Key that will be used to verify the response signature.

WS-FED 1.1

The SP provides the specific meta-data required for the protocol.

- Protocol Version: WS-Federation 1.1 Passive Profile
- SP Name: A Service Provider name that can be easily identified by the end user (this is a "user friendly" name).
- WTREALM (same as Entity ID in SAML): The WTREALM is the identifier that uniquely represents every WS-FED Service Provider.
- Consumer URL: URL that the assertion will be posted to.
- Does the SP require the assertions to be signed (Yes or No): Integrity of a message between providers is insured by using digital signatures. By enabling assertion signing, a digital signature is generated for the assertion document. If yes, Covisint will provide the SP with the Public Key that will be used to verify the assertion signature.

Custom

Custom solutions can be supported and will be defined and priced on a per project basis.

Attributes

The assertion which is sent to the Service Provider will always contain the user's Covisint Unique ID (an identifier that uniquely identifies any user in Covisint systems). Optionally, The Service Provider is able to use the self-service features available in AppCloud™ to view a list of standard user and group attributes that can be selected and included in the assertions that the Service Provider will be receiving from AppCloud. In addition to these standard attributes that will be available for every user and group across all Sponsors, Sponsor specific attributes may also be made available so that they can be included in the assertion.

Additional custom attributes that are required by the Service Provider can also be supported and will be defined and priced on a per project basis.

The tables below show the standard user and group attributes that are available to a Service Provider. The group attributes are the attributes associated with the group that the user is member of. A group can represent an Organization, Practice, Company, Site, Plant, HQ, etc.

There will always be an attribute value associated with information that a user or group is required to provide. There may not always be an attribute value associated with information that is optional for a user or group to provide. Attribute values will be included in the assertion whenever they are available.

User Attributes

NAME	DESCRIPTION	MAX LENGTH	ALWAYS A VALUE
CovisintUniqueID	Uniquely identifies a User in Covisint systems	80	Y
Prefix	Prefix to the User's Name (i.e. Mr, Ms)	20	N
FirstName	First (given) name of the User	150	Y
MiddleName	Middle name of the User	60	N
LastName	Last (family) name of the User	150	Y
Suffix	Suffix to the User's Name (i.e. Sr, Jr)	20	N
Address1	First line of the User's Address	255	Y
Address2	Second line of the User's Address	255	N
Address3	Third line of the User's Address	255	N
City	City or Region where the User is located	60	Y
State	State or Province where the User is located	60	Y
PostalCode	Postal code where the User is located	10	Y
CountryCode	Country code of the Country where the User is located	3	Y
EmailAddress	User's email address	1000	Y
PhoneNumber	User's phone number	100	Y
FaxNumber	User's fax number	100	N

Group Attributes

NAME	DESCRIPTION	MAX LENGTH	ALWAYS A VALUE
CovisintGroupID	Uniquely identifies a Group in Covisint systems	80	Y
GroupName	Name of the Group	150	Y
GroupAddress1	First line of the Group's Address	255	Y
GroupAddress2	Second line of the Group's Address	255	N
GroupAddress3	Third line of the Group's Address	255	N
GroupCity	City or Region where the Group is located	60	Y
GroupState	State or Province where the Group is located	60	Y
GroupPostalCode	Postal code where the Group is located	10	Y

GroupCountryCode	Country code of the Country where the Group is located	3	Y
GroupEmailAddress	Email address associated with the Group	1000	N
GroupPhoneNumber	Phone number associated with the Group	100	N
GroupFaxNumber	Fax number associated with the Group	100	N

Custom Attributes

Custom attributes that are required by the Service Provider can be supported and will be defined and priced on a per project basis.

Assertion Details

Details for the industry standard federation protocols that are supported by AppCloud™ are available at the following locations:

SAML 2.0

OASIS document “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0” which is located at:
http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

SAML 1.1

OASIS document “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1” which is located at:
http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

WS-FED 1.1

WS-Federation: Passive Requestor Profile V1.1 which is located at:
http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf

Application Context

Application context is implemented as a federation extension and is used to provide additional information, other than the information about the user, to an application when a user federates to the application. It provides additional "context" which relates to the action the user is engaged in prior to federating to the application. Application context is sent in the assertion to the Service Provider and then needs to be provided to the application when the user accesses the application.

A typical scenario, and the associated process flow, would be the following:

1. A user that is reviewing information about a specific person or product in a portal wants more information about that person or product (which is located in an application at a SP's site).
2. The user selects a link associated with the person or product of interest.
3. An identifier or additional information associated with the person or product is included in the federation assertion that is sent to Service Provider.
4. The application receives the identifier or additional information and uses it to provide the user with the specific information of interest, in the appropriate context, when the user accesses the application.

The result of the process is a seamless experience that provides the user with the specific information of interest directly upon accessing the application.

As an example, AppCloud™ supports a standard application context extension for providing patient context to Healthcare applications (which is described in the *AppCloud™ Healthcare Patient-Sync integration Guide*).

Custom application context extensions can be supported and will be defined and priced on a per project basis.