CA SiteMinder®

Federation in Your Enterprise

12.52 SP1

CA Technologies Product References

This document references the following CA Technologies products:

- CA SiteMinder®
- CA SiteMinder® Web Agent Option Pack
- CA SiteMinder® SPS

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

No updates have been made to the 12.52 SP1 documentation as a result of issues found in previous releases.

The following updates have been made to the 12.52 documentation, as a result of issues found in previous releases.

- Removed all references to the SAML Affiliate Agent, which is no longer supported.
- Federation use cases and solutions (see page 11) — Use cases have been updated to better reflect the concept of federated partnerships.
- Federation transaction process flows (see page 69)—Transaction diagrams and flows have been revised and updated to include the new checkpoint log messages.

Contents

Chapter 1: CA SiteMinder® Federation Deployments

Federation Deployment Models

CA CA SiteMinder® Federation has two deployment models:

- Partnership Federation

  Partnership federation is based on configuring partnerships between enterprises based on federation standards. The partnership model does not require configuration of CA SiteMinder®-specific objects, such as domains, realms, and policies. This model is recommended for new configurations using CA SiteMinder® Federation.

- Legacy Federation

  Legacy Federation (formerly Federation Security Services).

  Legacy federation is based on configuring CA SiteMinder® objects, such as affiliate domains, authentication schemes, and policies to protect federated resources. This model is primarily for backward compatibility with older deployments.

Both deployments provide user authentication data in the form of a SAML assertion. The entity that consumes the assertion uses the assertion to identify the user. Upon successful authentication, the consuming entity makes the requested resources available. The result is a seamless experience for the user.

Install the CA SiteMinder® Policy Server, the Administrative UI, and the Web Agent Option Pack to use either model.

Note: Federation is separately licensed from CA SiteMinder®.

Federation Specifications

CA SiteMinder® supports the following federation specifications:

Security Assertion Markup Language (SAML)

The Security Assertion Markup Language (SAML) is a standard from the Organization for the Advancement of Structured Information Standards (OASIS). This industry standard defines an XML framework for exchanging authentication and authorization information.

SAML defines assertions as a means to pass security information about users between entities. SAML assertions are XML documents that contain information about a specific subject, such as a user. An assertion can contain several different internal statements about authentication, authorization, and attributes.

SAML defines two browser-based protocols that specify how SAML assertions are passed between partners to facilitate single sign-on.

The profiles are:

- Browser/artifact profile—defines a SAML artifact as a reference to a SAML assertion.
- Browser/POST profile—returns a response that contains an assertion.

Note: For SAML 2.0, the artifact and POST profiles are referred to as HTTP bindings.

For SAML specifications and information about SAML profiles, refer to the Organization for the Advancement of Structured Information Standards (Oasis).

CA SiteMinder® supports the following SAML standards and profiles:

- SAML 1.0 Artifact profile only (legacy federation only)
- SAML 1.1 Artifact and POST profile
- SAML 2.0 Artifact and POST profile

WS-Federation

Active Directory Federation Services (ADFS) is web services-based solution from Microsoft for federated single sign-on (SSO). ADFS runs on a Windows server and accomplishes SSO by letting partners securely share user identity information and access rights across a secure network. ADFS extends SSO functionality to internet applications, letting users have a seamless web SSO interaction when they access web-based applications of the organization.

ADFS uses the WS-Federation specification for communication. For WS specifications and background documentation, and information about ADFS profiles, go to the Microsoft website.

Entities in a Federated Network

In a federated network, one entity generates a SAML assertion or a WS-Federation token containing an assertion. Assertions contain information about a user whose identity is maintained locally at the site that generates them. The other entity uses the assertions to authenticate a user and to establish a session for the user.

Depending on the protocol, these two entities are named differently, but their functions are the same.

Protocol	Generates Assertions	Consumes Assertions
SAML 1.0 and 1.1	Producer	Consumer
SAML 2.0	Identity Provider (IdP)	Service Provider (SP)
WS-Federation (Partnership)	Identity Provider (IP)	Resource Partner (RP)
WS-Federation (Legacy)	Account Partner (AP)	Resource Partner (RP)

A single site can be the asserting party and the relying party.

Chapter 2: Federation Use Cases and Solutions

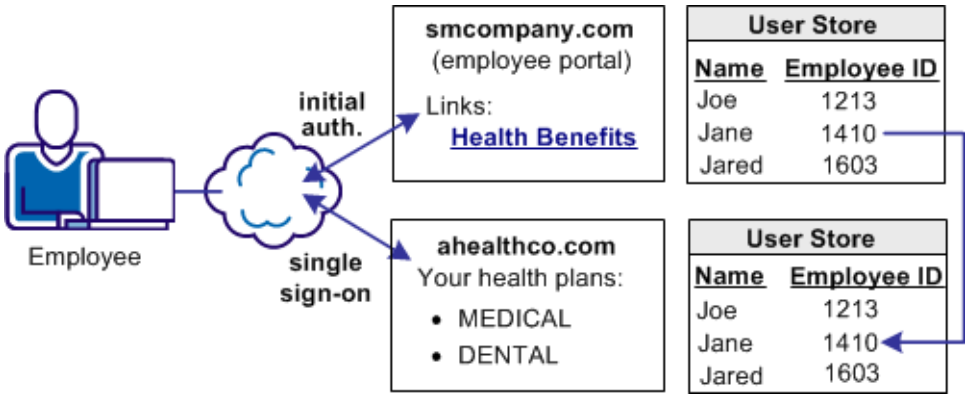
This section contains the following topics:

Use Case: Single Sign-on Based on Account Linking

In this use case, smcompany.com contracts with a partner, ahealthco.com to manage employee health benefits.

An employee of smcompany.com authenticates at an employee portal at the company website, smcompany.com and clicks a link to view her health benefits at ahealthco.com. The employee is taken to the ahealthco.com web site and is presented with the correct health benefit information without having to sign on to the website.

The following illustration shows this use case.

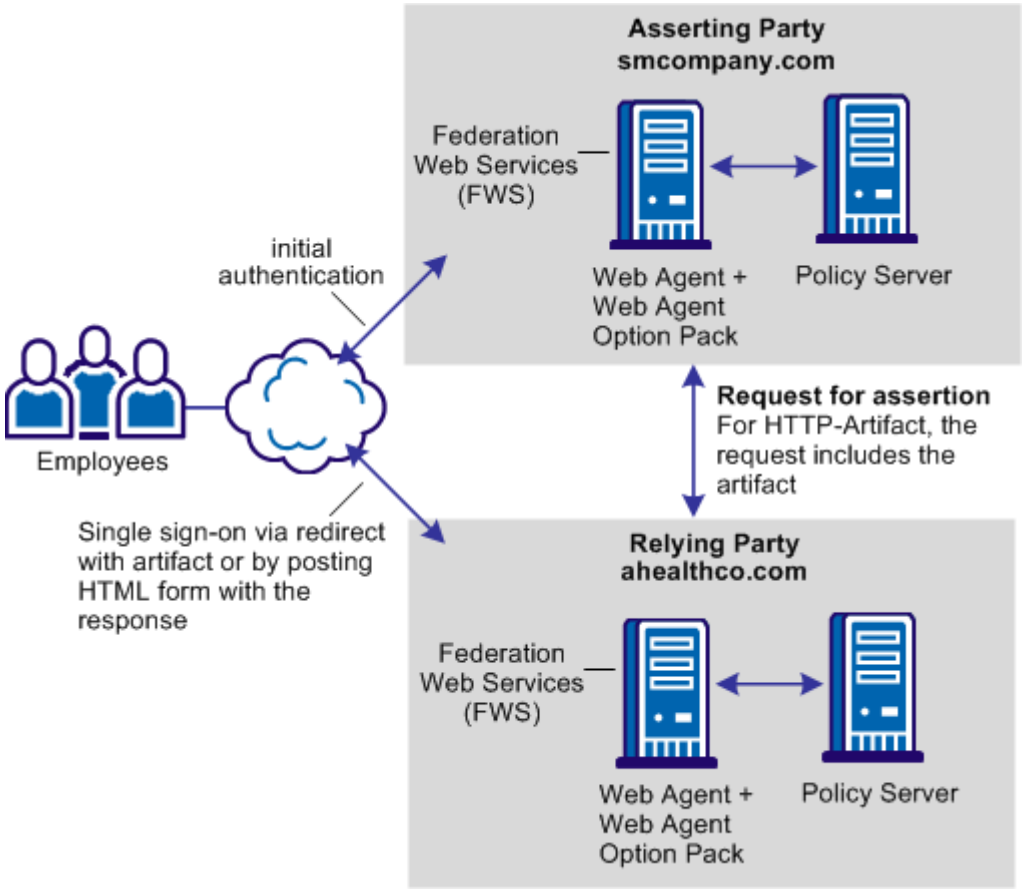


Account linking can be used for browser-based single sign-on, where each partner maintains separate user accounts for the same user. Account linking uses the SAML assertion to associate a federated identifier with the local identity at a partner.

In this use case, ahealthco.com maintains all health-related information and user identities for every employee at smcompany.com. When an employee of smcompany.com accesses ahealthco.com, an identifier for the employee is passed from smcompany.com to ahealthco.com in a secure manner. This identifier allows ahealthco.com to determine who the user is and the level of access to allow for that user.

Solution: Single Sign-on based on Account Linking

Federation can be deployed at smcompany.com and ahealthco.com to solve Use Case: Single Sign-on Based on Account Linking (see page 11).



CA SiteMinder® is deployed at both sites. The Web Agent with the Web Agent Option Pack are installed on a webserver system and the Policy Server is installed on another system. The installations are the same for smcompany.com and ahealthco.com.

The FWS application provides the servlets which retrieve assertions for the HTTP-Artifact profile and which consume assertions.

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

Account Linking Solution: SAML 1.1 HTTP-Artifact Profile

In this example, smcompany.com is the producer. The administrator at smcompany.com configures a SAML 1.1 producer-to-consumer partnership between smcompany.com and ahealthco.com. This partnership uses the HTTP-Artifact profile for single sign-on.

The partnership at smcompany.com has the following information:

- The location of the assertion consumer service at ahealthco.com.
- The unique Name ID.
- Assertion attributes to be added to the assertion.

An employee of smcompany.com logs into the company site and is initially authenticated by the Web Agent. The employee of smcompany.com accesses the employee portal web site and the sequence of events is as follows:

1. The employee clicks a link at smcompany.com to view health benefits at ahealthco.com. The link makes a request to the Intersite Transfer Service at smcompany.com.
2. The Intersite Transfer Service calls the Policy Server and sends a request that the Policy Server generate an assertion and artifact. The Policy Server generates an assertion and places assertion in the session store. It also generates and returns the artifact to the service.
3. The Web Agent redirects the user to ahealthco.com with the SAML artifact.

Ahealthco.com is the consumer site. The administrator at ahealthco.com configures a consumer-to-producer partnership with smcompany.com. This partnership uses the HTTP-Artifact profile for single sign-on.

The partnership configuration has the following information:

- The location of the artifact retrieval service at smcompany.com.
- Which attribute in the assertion to use for locating a user in the user directory.
- The search string to locate the user record in the local directory. This record must match the value in the assertion.
- The target resource.

Ahealthco.com receives the assertion and the sequence of events follows:

1. The browser posts the response to the SAML credential collector URL.
2. The service sends a request with the SAML artifact to the assertion retrieval service at smcompany.com. The assertion retrieval service extracts the session ID from the artifact.
3. The assertion retrieval service obtains the assertion from the session store. It sends the assertion as an artifact response to the SAML credential collector at ahealthco.com.
4. The SAML credential collector validates the assertion. The Policy Server creates a session and places a session cookie in the browser for the ahealthco.com domain.
5. The SAML credential collector redirects the user to the target resources at ahealthco.com.

Account Linking Solution: SAML 1.x POST Profile

In this example, smcompany.com is the producer. The administrator at smcompany.com configures a producer-to-consumer partnership. The partnership uses SAML 1.x POST profile for single sign-on.

The partnership configuration has the following information:

- The location of the assertion consumer service at ahealthco.com.
- The unique Name ID.
- Assertion attributes to be added to the assertion.

When an employee of smcompany.com accesses the employee portal site, the sequence of events is as follows:

1. The Web Agent provides the initial authentication.
2. The employee clicks a link at smcompany.com to view the health benefits at ahealthco.com. The link makes a request to the Intersite Transfer Service at smcompany.com.

3. The Intersite Transfer Service calls the assertion generator, which creates a SAML assertion and signs the SAML response.
4. The signed response is placed in an auto-POST HTML form and sent to the user's browser.
5. The browser posts a form containing the response to the Assertion Consumer Service at ahealthco.com..

Ahealthco.com is the consumer site. The SAML credential collector service at ahealthco.com handles the SAML response. The administrator at ahealthco.com configures a consumer-to-producer partnership with smcompany.com that uses the SAMl 1.1 HTTP-POST profile for single sign-on.

The partnership configuration has the following information:

- The location of the artifact retrieval service at smcompany.com.
- Which attribute in the assertion to use for locating a user in the user directory.
- The search string to locate the user record in the local directory. This record must match the value in the assertion.
- The target resource.

The sequence of events is as follows:

1. The SAML credential collector receives the assertion from the producer.
2. The SAML credential collector calls the Policy Server at ahealthco.com.
3. The Policy Server verifies the signature of the assertion then uses it to authenticate the user.
4. After successful authentication, the Policy Server creates an SMSESSION cookie and places it in the browser
5. The browser redirects the user to the target resource at ahealthco.com.

Account Linking Solution: SAML 2.0 Artifact Profile

In this example, smcompany.com is the Identity Provider. The administrator at smcompany.com configures an IdP-to-SP partnership with ahealthco.com as the remote SP.

The partnership configuration contains the following information:

- The location of the assertion consumer service at ahealthco.com.
- The unique Name ID.
- The assertion attributes to be added to the assertion.

An employee accesses the employee portal site and the following sequence of events occurs:

1. The Web Agent provides the initial authentication.
2. The user clicks a link to view health benefits at ahealthco.com. Because the request is initiated at the Identity Provide, the request triggers an unsolicited response.
3. Federation Web Services (FWS) requests the SAML artifact from Policy Server.
4. The Policy Server generates a SAML assertion and an artifact. The Policy Server stores the assertion in the session store, and the artifact as a URL parameter.
5. The Policy Server returns the response containing the SAML artifact to FWS.
6. The Web Agent redirects the user with the SAML artifact to ahealthco.com.

Ahealthco.com is the Service Provider. The administrator at ahealthco.com configures an SP-to-IdP partnership with smcompany.com, which uses the artifact profile. The partnership configuration has the following information:

- The location of the Single Sign-on Service at smcompany.com.
- Which attribute in the assertion to use for locating a user in the user directory.
- The search string to locate the user record in the local directory. This record must match the value in the assertion.
- The target resource.

The sequence of events is as follows:

1. The Assertion Consumer Service receives the artifact. The service obtains the location of the Artifact Resolution Service at smcompany.com from its partnership configuration.
2. The Assertion Consumer Service makes a call across the back channel to the Artifact Resolution Service at smcompany.com.
3. The Policy Server retrieves the assertion from the session store and returns the response to the Assertion Consumer Service at ahealthco.com.
4. The Assertion Consumer Service validates the response and creates the session for ahealthco.com. A session cookie is written to the browser.
5. The browser redirects the user to the target resource at ahealthco.com.

Account Linking Solution: SAML 2.0 POST Profile

In this example, smcompany.com is the Identity Provider. The administrator at smcompany.com configures an IdP-to-SP partnership. The partnership uses SAML 2.0 HTTP-POST profile for single sign-on.

The partnership configuration has the following information:

- The location of the assertion consumer service at ahealthco.com.
- The unique Name ID.
- The assertion attributes to be added to the assertion.

An employee of smcompany.com logs in to the employee portal site.

After a successful initial authentication, the following sequence occurs:

1. The Web Agent at smcompany.com initially authenticates the user.
2. The employee clicks a link to ahealthco.com to view health benefits. The Policy Server reads the SAML 2.0 SP configuration.

   The Identity Provider initiates the request, which triggers an unsolicited response.
3. A request is sent to the Single Sign-on (SSO) service at smcompany.com.
4. The SSO service makes a request to policy server to generate a SAML 2.0 assertion/artifact based on the selected profile. For HTTP-POST, the Policy Server generates a SAML assertion.
5. The SSO service receives the assertion response for the selected profile.
6. The signed response is placed in an auto-POST HTML form and sent to the browser.
7. The browser POSTs the response to the Assertion Consumer Service at ahealthco.com.

Ahealthco.com is the Service Provider. The administrator at ahealthco.com configures an SP-to-IdP partnership with smcompany.com. The configuration uses the SAML 2.0 HTTP-POST profile for single sign-on.

The partnership configuration has the following information:

- The location of the Artifact Retrieval Service at smcompany.com.
- Which attribute in the assertion to use for locating a user in the user directory.
- The search string to locate the user record in the local directory. This record must match the value in the assertion.
- The target resource.

The sequence of events at ahealthco.com is as follows:

1. The Assertion Consumer Service obtains the response message from the post data.
2. The Assertion Consumer Service reads the IdP configuration to get the target URL.
3. The Assertion Consumer Service passes the signed SAML response as credentials to the Policy Server at ahealthco.com.
4. The Policy Server verifies the signature and then authenticates the user.
5. The login is successful.
6. The Policy Server creates an SMSESSION cookie for the ahealthco.com domain and places the cookie in the browser.
7. The browser redirects the user to the target resource at ahealthco.com.

Account Linking Solution: WS-Federation Passive Requestor Profile

In this example, smcompany.com is the Identity Provider. The administrator at smcompany.com configures a WSFED IP-to-RP partnership. The partnership uses the WS-Federation Passive Requester profile for single sign-on. In this use case, ahealthco.com, the Resource Partner, initiates single sign-on.

The SAML token type is SAML 1.1. This part of the IP entity configuration.

The partnership configuration has the following information:

- The location of the Security Token Consumer Service at ahealthco.com.
- The unique Name ID.
- The assertion attributes to be added to the assertion.

When an employee of smcompany.com accesses the employee portal, the following sequence of events occurs:

1. The user visits an unprotected site selection page at ahealthco.com. The Web Agent provides the initial authentication.
2. The user clicks a link that points to the Single Sign-on Service at smcompany.com. The browser redirects the user to smcompany.com.
3. The SSO Service calls the Policy Server. The Policy Server generates the assertion.
4. The Policy Server signs the assertion element of the Request Security Token Response and returns a response.
5. The browser POSTS the response in an auto-POST HTML form to the Security Token Consumer Service at ahealthco.com.

Ahealthco.com is the Resource Partner.

The partnership configuration has the following information:

- The location of the single sign-on service at smcompany.com.
- Which attribute in the assertion to use for locating a user in the user directory.
- The search string to locate the user record in the local directory. This record must match the value in the assertion.
- The target resource.

The sequence of events is as follows:

1. The Security Token Consumer Service extracts the assertion from the security token consumer response.
2. The service determines the target resource.
3. The Security Token Consumer Service passes the signed assertion as credentials to the Policy Server at ahealthco.com.
4. The Policy Server verifies the signature and then authenticates the user.
5. After successful authentication, the Security Token Consumer Service creates an SMSESSION cookie.
6. The service then places the cookie in the browser and redirects the user to the target resource at ahealthco.com.

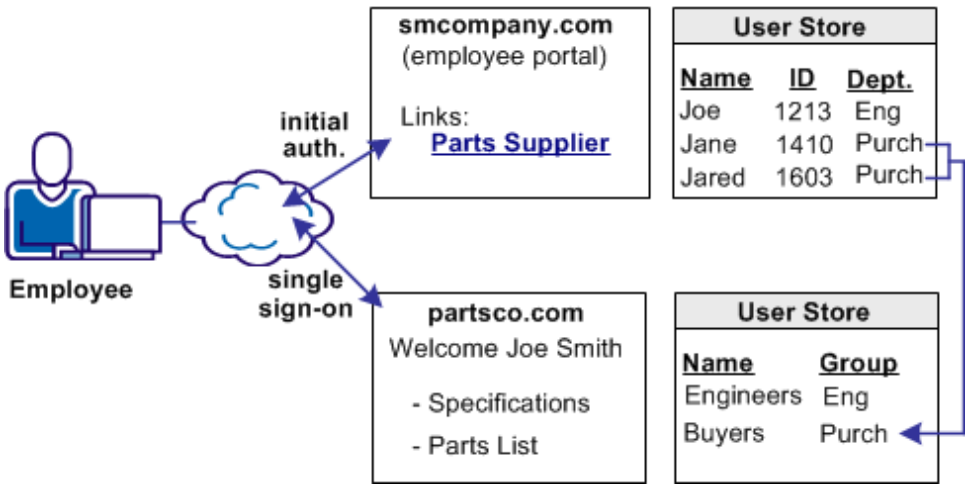
Use Case: Single Sign-on Based on User Attributes

In Use Case 2, smcompany.com buys parts from a business partner named partsco.com.

An engineer authenticates at smcompany.com and clicks a link to access information at partsco.com. As an engineer at smcompany.com, the user is taken directly to the Specifications portion of the partsco.com website without having to log in.

A buyer for smcompany.com authenticates and clicks a link for partsco.com. The buyer is taken directly to the Parts List portion of the partsco.com website. The buyer does not have to log in.

The following graphic shows the relationship between the two partners.



Other attributes, such as the user name are passed from smcompany.com to partsco.com to personalize the interface for the individual user.

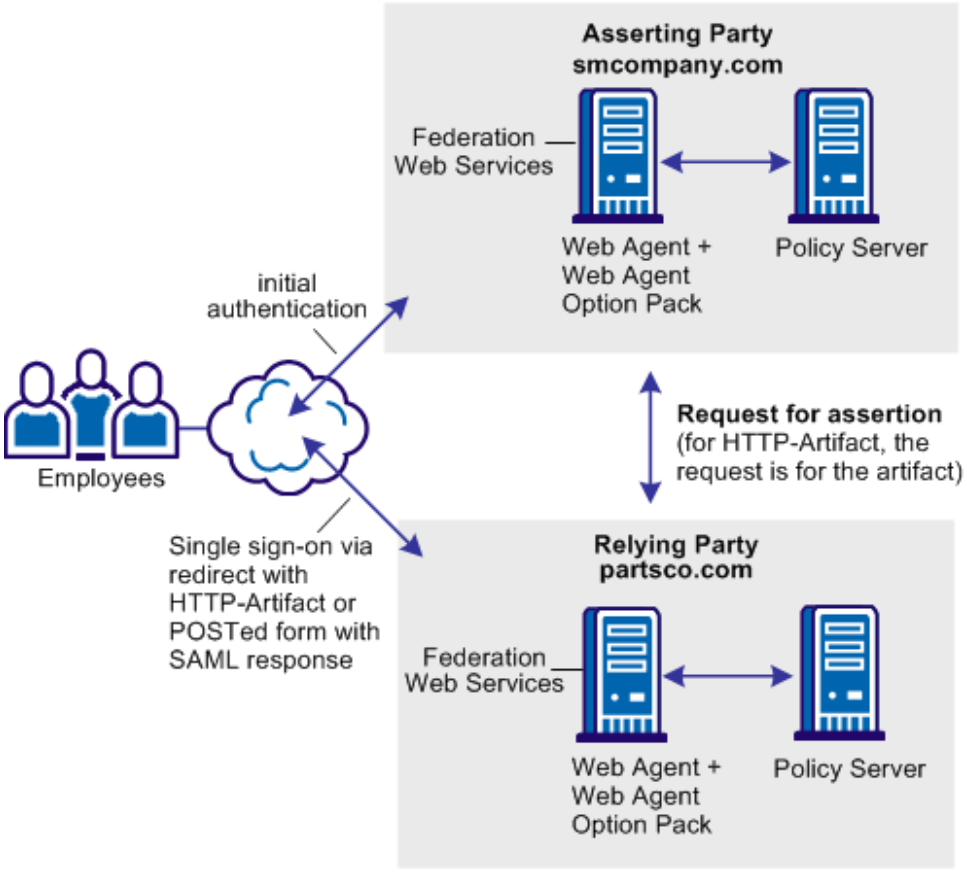
Partsco.com does not want to maintain user identities for all smcompany.com employees, but the company wants to control access to sensitive portions of the website. To control the access, partsco.com maintains a limited number of identities for users at smcompany.com. One identity is maintained for Engineers and one identity is maintained for Buyers.

When an employee of smcompany.com accesses partsco.com, smcompany.com sends user attributes in a secure manner to partsco.com. Partsco.com uses the attributes to determine which identity controls access for the user.

Solution: Single Sign-on based on User Attributes

Federation can be deployed at smcompany.com and partsco.com to solve Use Case: Single Sign-on Based on User Attribute Profiles (see page 19).

The illustration is similar for SAML 1.1, SAML 2.0, and WS-Federation.



CA SiteMinder® is deployed at both sites. The interaction between the user and each site is similar, where partsco.com is acting as the relying party. The Federation Web Services application contains all the necessary servlets to process the transaction.

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is similar to the solution for Single Sign-on based on Account Linking (see page 12), except for the following items:

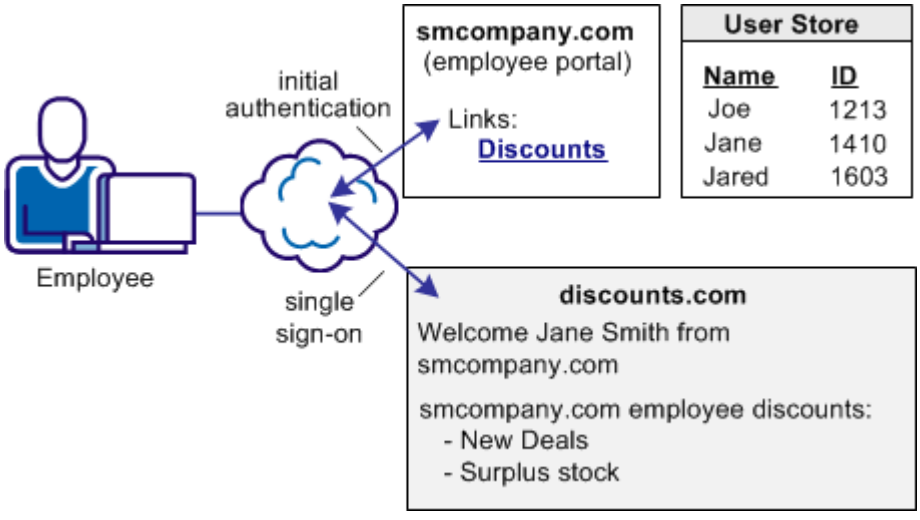
- The administrator at smcompany.com defines the partnership with partsco.com.
- The partnership configuration includes an assertion attribute named *department*. This attribute specifies the department to which the user belongs. The Policy Server includes this attribute in the assertion it generates for the requesting user.
- The administrator defines one user record for each department that is allowed to access the partsco.com website.
- The administrator at partsco.com defines a partnership with smcompany.com.
- The Assertion Consumer Service extracts the department attribute from the assertion. The Policy Server searches the user directory at partsco.com for the user record that has the matching value for the department attribute.

Use Case: Single Sign-on with No Local User Account

In this use case, smcompany.com offers employee discounts by establishing a partnership with discounts.com.

An employee authenticates at smcompany.com and clicks a link to access discounts.com. The employee is taken to the discounts.com website and is presented with the discounts available for smcompany.com employees, without logging in to the discounts.com website.

The following illustration shows this use case.

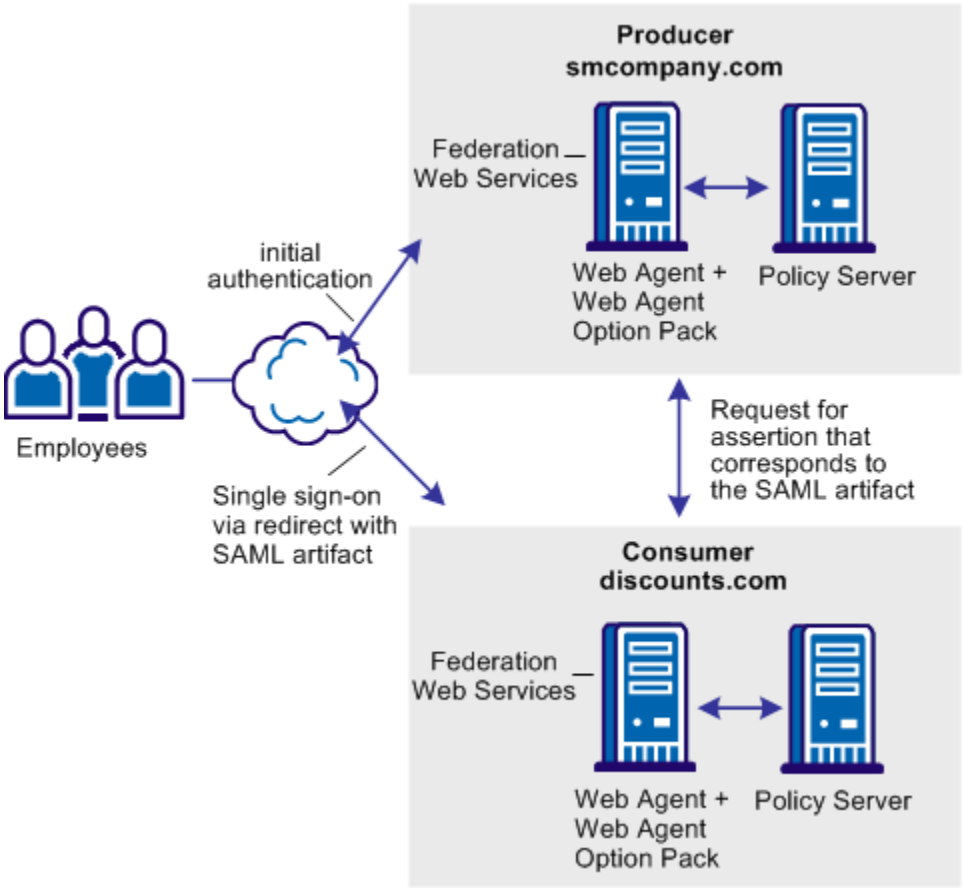


Discounts.com does not maintain any identities for smcompany.com. The company allows all employees of smcompany.com to access discounts.com as long as they have been authenticated at smcompany.com. Smcompany.com sends authentication information about the user requesting a resource to discounts.com in a secure manner so access is permitted.

Solution: Single Sign-on with no Local User Account

Federation is deployed at smcompany.com and discounts.com to solve Use Case: Single Sign-on with No Local User Account (see page 22).

The following figure shows single sign-on with no local user account. SAML 1.1 is the SSO profile in use.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

In this deployment, CA SiteMinder® is at both sites. Smcompany.com is the SAML 1.1 producer. The administrator at smcompany.com configures a SAML 1.1 partnership that includes a remote entity representing discounts.com. Any attributes that are configured in the partnership get included in the assertion.

For this solution to work, every user must map to a single user account, making the single user essentially an anonymous user.

When an smcompany.com employee accesses the employee portal, the following process occurs:

1. The Web Agent provides initially authenticates.
2. The employee clicks a link to access deals at discounts.com. This link is referred to as the intersite transfer URL because it results in transferring the user to another site.
3. The intersite transfer URL makes a request to the Web Agent. This URL contains the location of the SAML credential collector and the target URL at the consumer site.
4. The Web Agent at smcompany.com calls the Policy Server. The Policy Server generates an assertion and an artifact, and stores the assertion in the session store.
5. The Policy Server returns the artifact to the FWS application, which in turn creates a response.
6. The browser redirects the user with the artifact response to discounts.com.

Discounts.com is the consumer site. The administrator at discounts.com configures an SP-to-IdP partnership. The partnership configuration specifies the location of the assertion retriever service at smcompany.com, and the protected target resources.

The user identification configuration for the partnership must specify a custom user search specification that looks up a single user. For example, if the user directory is LDAP, the search specification is uid=user1.

Important! To map every user to a single user, a user directory at discounts.com must exist. This user directory must contain a single user record. An alternative is to create a user directory using the Policy Server API that returns the same user record.

The following process occurs:

1. The browser posts the response to the SAML credential collector, which obtains the location of the assertion retrieval service at smcompany.com.
2. The SAML credential collector makes a back channel call to the Assertion Retrieval Service at smcompany.com. The session ID is extracted from the artifact.
3. The Policy Server retrieves the assertion from the session store and returns it to the SAML credential collector at discounts.com.

4. The SAML credential collector then validates the SAML assertion and issues a session cookie to the browser.
5. The browser redirects the user to the target resource at discounts.com.

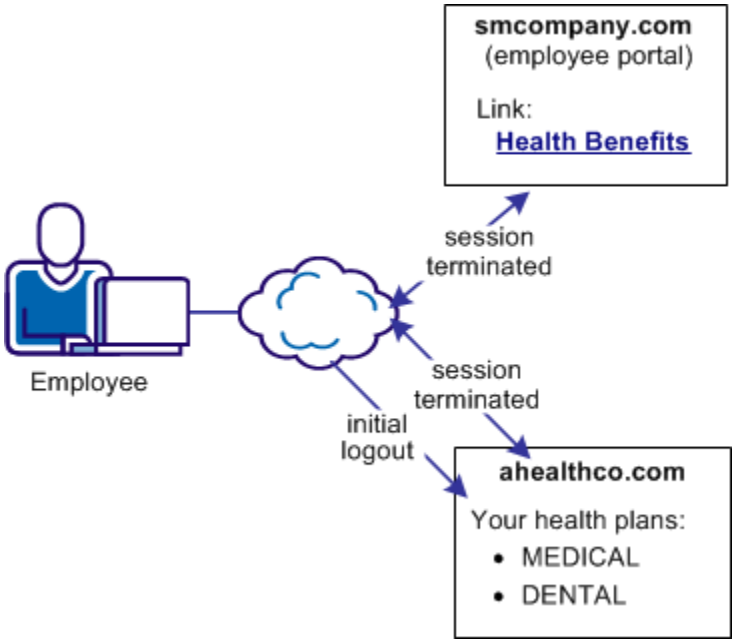
Use Case: SAML 2.0 Single Logout

In this use case, an employee of smcompany.com authenticates at the employee portal and selects a link to view the health benefits at ahealthco.com. The employee is taken to the ahealthco.com website and presented with the health benefit information without logging in to the site.

After the employee logs out from ahealthco.com, the site wants to verify the termination of the user session at ahealthco.com and at smcompany.com. Terminating both sessions prevents an unauthorized employee from using the existing session to access resources at smcompany.com or to view benefits of the authorized employee.

Note: In this case, the initial logout occurs at ahealthco.com and results in both sessions being terminated.

The following illustration shows the use case.



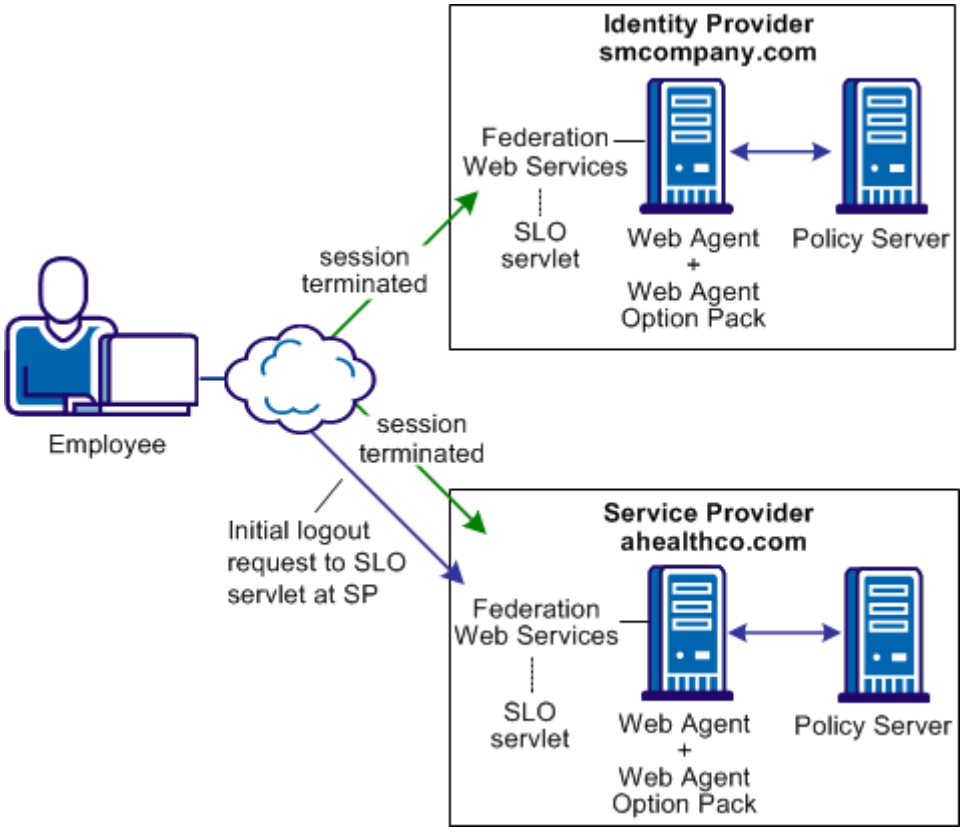
Solution: SAML 2.0 Single Logout

You can use federation to solve Use Case: SAML 2.0 Single Logout (see page 25).

In this solution:

- smcompany.com is the Identity Provider.
- ahealthco.com is the Service Provider that initiates the logout request.
- Single logout is enabled at the Identity Provider and the Service Provider.

The following figure shows the solution for single logout.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The following sequence of events for SP-initiated single logout occurs:

1. An employee authenticates at smcompany.com then accesses health benefits at ahealthcom.com by way of federated single sign-on. Smcompany.com places information about ahealthco.com in its session store. Ahealthco.com places information about smcompany.com in its session store.
2. The employee finishes reviewing health benefits and clicks a log-out link at ahealthco.com. The browser accesses the single logout servlet.
3. The FWS application at ahealthco.com renames the existing SMSESSION cookie to SESSIONSIGNOUT to invalidate the current session of the user.
4. The user session is terminated at the ahealthco.com.

   Note: The termination does not remove the session from the session store; it sets the state to LogoutInProgress.
5. The Policy Server generates a logout request to invalidate the user session at smcompany.com. The Policy Server also returns the provider ID of smcompany.com.
6. The browser redirects the log out request to the single logout servlet at smcompany.com, with the logout request message added as a query parameter.
7. The FWS application receives the log out request message, and renames the SMSESSION cookie to SESSIONSIGNOUT.
8. FWS invalidates the user session from all Service Providers that are associated with that user session. The only exception is the session at ahealthco.com, who initiated the log out request.
9. After all the Service Providers confirm the logout, smcompany.com removes the user session from its session store. FWS deletes the SESSIONSIGNOUT cookie.

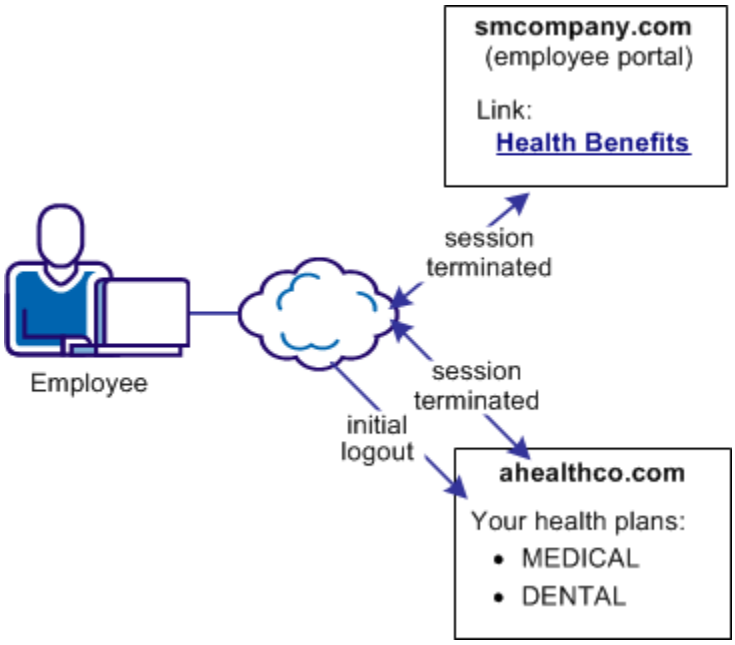
   Note: Other Service Providers are not identified in the illustration.
10. Smcompany.com returns a logout response message to ahealthco.com, the initiating Service Provider, and the user session is removed from its session store.
11. The user is finally sent to the SLO configuration page at ahealthco.com.

Use Case: WS-Federation Sign-out

In this use case, an smcompany.com employee authenticates at the employee portal. The employee selects a link to view health benefits at ahealthco.com. The employee is taken to the ahealthco.com website and presented with the health benefit information without having to sign on to the site.

When the employee logs out, ahealthco.com wants the user session at its site and at smcompany.com terminated. Terminating both sessions prevents an unauthorized person from using the existing sessions to access resources at smcompany.com or ahealthco.com.

The following illustration shows the use case.



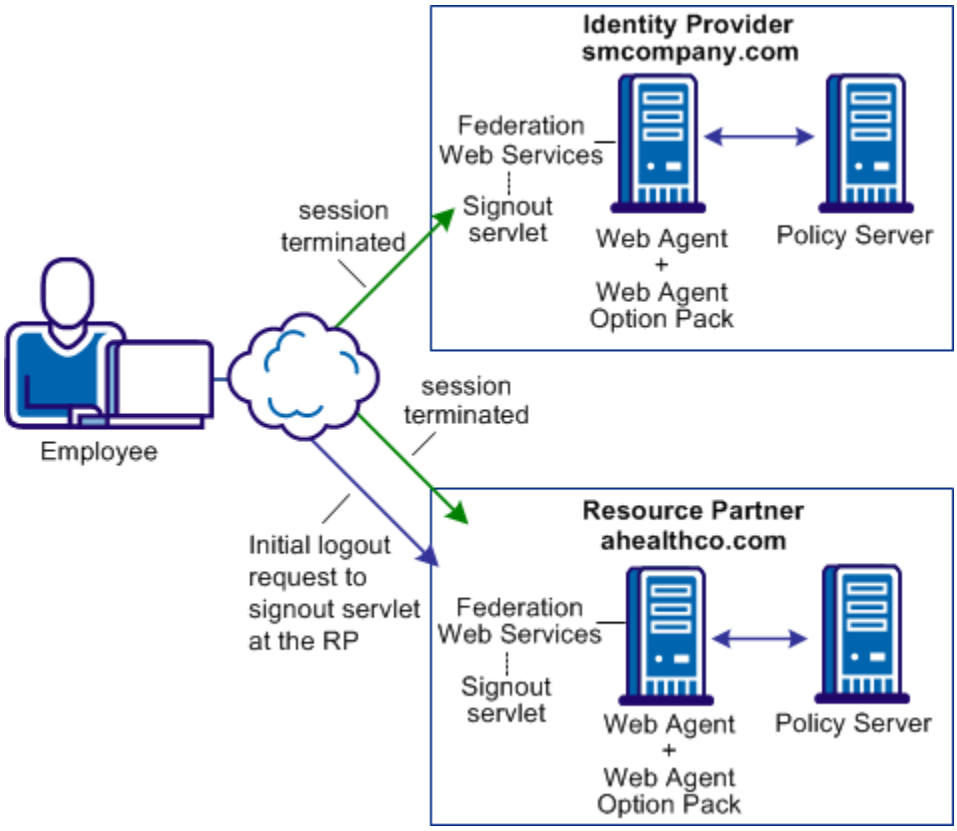
Solution: WS-Federation Signout

The following federated deployment solves Use Case: WS-Federation Sign-out (see page 27).

In this solution:

- smcompany.com is the Identity Provider.
- ahealthco.com is the Resource Partner, and it initiates the sign-out request.
- A WSFED IP-to-RP partnership is configured between smcompany.com and ahealthco.com to enable single sign-on.
- WS-Federation sign-out using the HTTP binding is enabled at the Identity Provider and the Resource Partner.

The following figure illustrates WS-Federation sign-out.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The following sequence of events occurs:

1. An employee authenticates at smcompany.com and federates to ahealthco.com to access their health benefits. During the transaction, smcompany.com places information about ahealthco.com in its session store. Ahealthco.com places information about smcompany.com in its session store.
2. The employee finishes viewing his health benefit information, and clicks a log-out link at ahealthco.com. The sign-out service receives a sign-out request.
3. The sign-out service fetches session information from the SMSESSION cookie and determines the Identity Provider that is associated with the user session.
4. The sign-out service calls the Policy Server to invalidate the session.
5. The sign-out service generates a sign-out request, and forwards the sign-out request to the Signout URL at smcompany.com.
6. The sign-out service at smcompany.com receives the request.

7. The sign-out service fetches session information from the SMSESSION cookie and determines the Resource Partner that is associated with the user session.
8. The sign-out service calls the Policy Server to invalidate the session.
9. The sign-out service generates a sign-out request, and posts a sign-out message and multiple RP-SignoutCleanup locations as post data to the SignoutConfirmURL JSP.

   The SignoutConfirm page generates a frame-based HTML page. Each frame contains a Signout Cleanup URL for each Resource Partner that is associated with the user session.
10. Ahealthco.com forwards the sign-out request to any Resource Partner associated with the user session. At each RP, the session is terminated from the session store.
11. Each RP redirects the browser to the Sign-out Cleanup URL ahealthco.com, as the initiating partner to complete the sign-out.

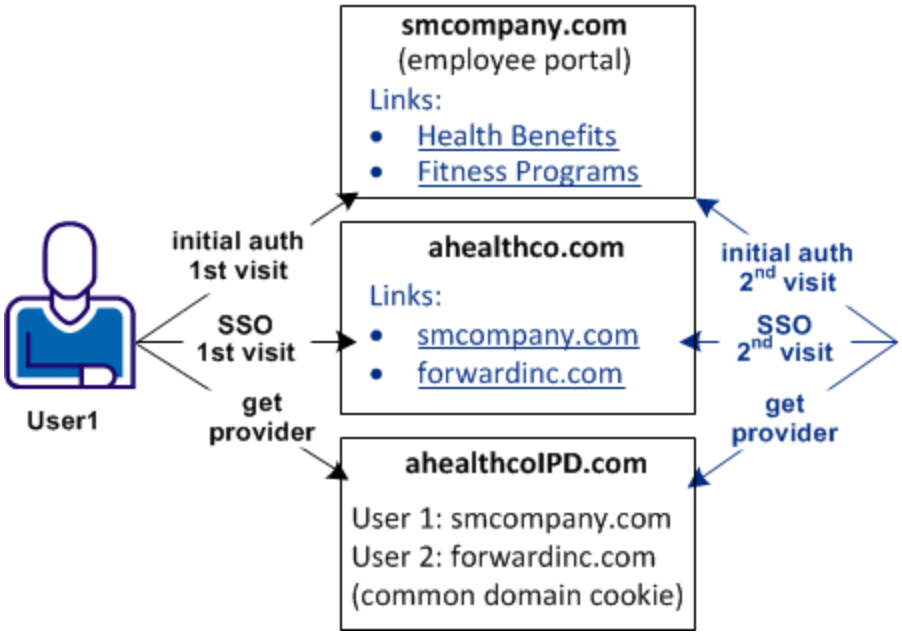
Use Case: Identity Provider Discovery Profile

In this use case, several companies contract health benefits from ahealthco.com. A user logs on to ahealthco.com to view their health benefits. Ahealthco.com must determine which Identity Provider it sends an authentication request to for a particular user.

IdP discovery is useful in federated networks that have more than one partner providing assertions. This profile provides a dynamic way for a Service Provider to determine which Identity Provider has the necessary user record.

The following illustration shows a network with the Identity Provider Discovery profile in use.

Note: A prior business agreement between the sites in this network exists so that all sites interact with the Identity Provider Discovery service.



In this example, User1 arrives at ahealthco.com. Ahealthco.com determines that User1 came from smcompany.com. Aheathco.com sets a cookie for smcompany.com in the common domain cookie at ahealthco.com. Other companies, such as forwardinc.com is another Identity Provider that uses ahealthco.com as a health provider. When a user from forwardinc.com comes to ahealthco.com, a cookie is set in the common domain cookie also.

Solution: Identity Provider Discovery Profile

Federation can solve Use Case: Identity Provider Discovery Profile (see page 30).

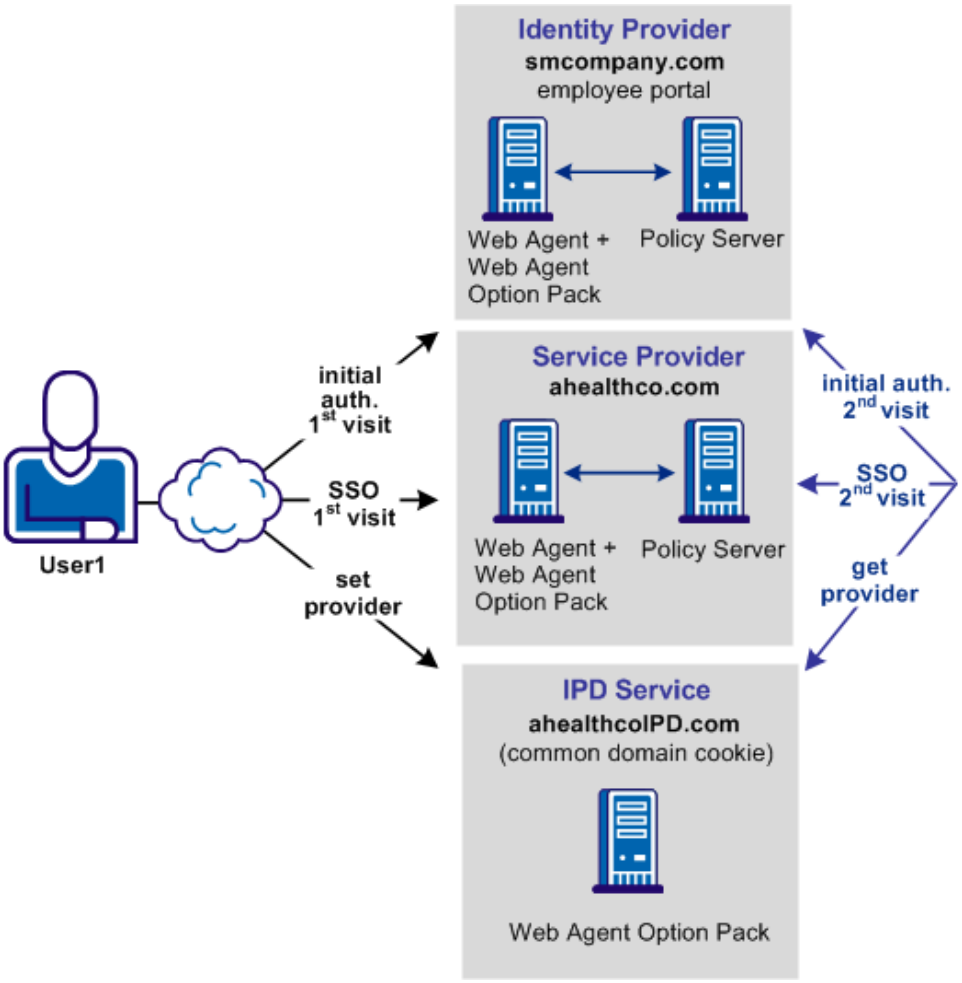
The IdP Discovery profile (SAML 2.0 only) is implemented using a cookie domain that is common to the two federated partners. A cookie in the agreed upon domain contains the list of IdPs that a user has visited.

Note: The user that the Service Provider wants to authenticate must have visited the Identity Provider and authenticated before arriving at the Service Provider.

In this solution:

- smcompany.com issues assertions for User 1 and has ahealthco.com configured as its Service Provider.
- ahealthco.com is the Service Provider for smcompany.com. This site has SAML 2.0 SP-to-IdP partnerships configured with each Identity Provider that uses its services.
- ahealthcoIPD.com is the Identity Provider Discovery Service for ahealthco.com. The Federation Web Services application installed with the Web Agent Option Pack, provides the IPD service. This service reads the common domain cookie, which includes all relevant Identity Providers for ahealthco.com.

The following illustration shows the federated network for this solution.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The transaction flow is as follows:

1. User 1 initially logs in and authenticates at smcompany.com. The user then federates to ahealthco.com without having to reauthenticate.

   An agreement exists between smcompany.com and ahealthco.com to use ahealthcoIPD.com as the IPD service.

2. The FWS application at smcompany.com requests the Identity Provider Discovery Profile (IPD) configuration from the Policy Server, passing its Identity Provider ID.

3. The Policy Server returns with the IPD configuration, such as IPD Service URL, common domain cookie, and persistence information of the common domain cookie.

4. The FWS application at smcompany.com redirects the user to the IPD Service URL to set the common domain cookie. The Identity Provider ID of smcompany.com is written to the common domain cookie at the IPD service.

5. The IPD service redirects the user back to the single sign-on service at smcompany.com. The redirect contains an authentication request.

6. The FWS application at smcompany.com requests an assertion from the Policy Server. The Policy Server generates the assertion that is based on the partnership configuration it has for ahealthco.com.

7. The FWS application returns the assertion response back to ahealthco.com.

8. User 1 is now successfully logged on to ahealthco.com and can look at the health benefits. The user finishes reviewing health benefits and logs out.

9. In a separate transaction on a different day, User 1 logs in to ahealthco.com directly. User 1 clicks a link to view healthy benefits again. AhealthIPD.com has cookies for all Identity Providers where User 1 has visited. ahealthco.com calls the IPD discovery service to obtain the Identity Provider IDs.

10. Ahealthco.com presents a site selection page to User 1 to select the company where they can authenticate. User 1 selects smcompany.com.

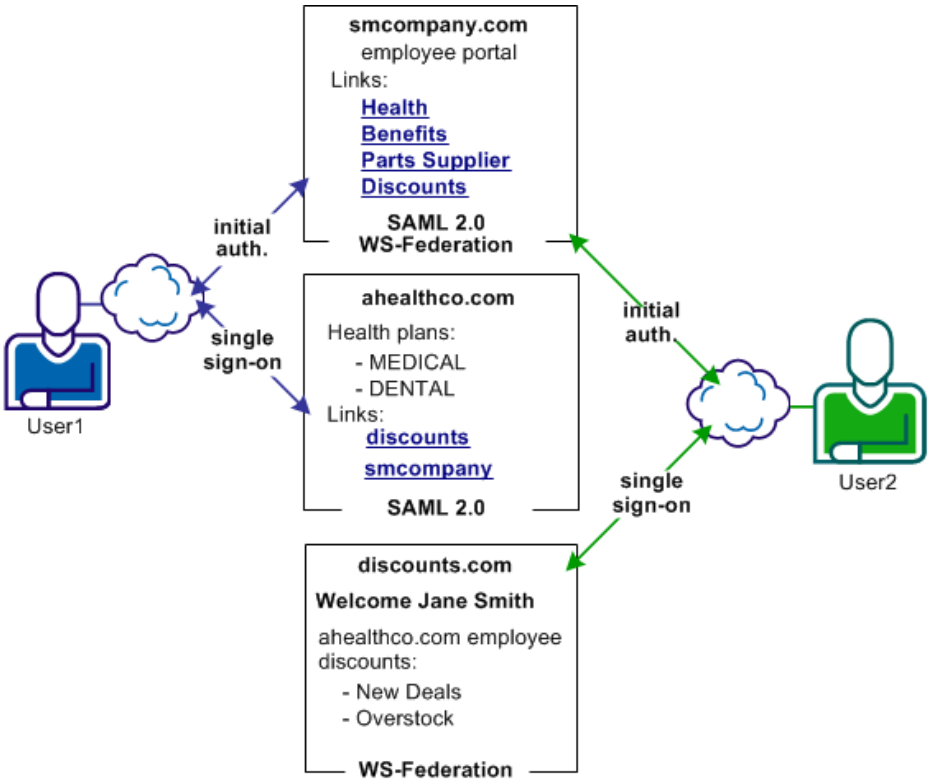
11. Ahealthco.com sends an authentication request to smcompany.com. The Policy Server at smcompany.com generates an assertion and sends it back to the Assertion Consumer Service at ahealthco.com.

12. The user successfully logs in and is redirected to the requested resource.

Use Case: Federation with Multiple SSO Profiles

In this use case, smcompany.com issues assertions for ahealthco.com and discounts.com. Ahealthco.com uses the SAML 2.0 profile. Discounts.com uses the WS-Federation profile. The assertions issued must be generated according to the appropriate profile so that the relying party can consume the assertion.

The following illustration shows the multiprotocol use case.



Solution: Federation with Multiple SSO Profiles

The following federation deployment solves the Use Case: Federation with Multiple SSO Profiles (see page 33).

Note: The single sign-on transactions in this solution are similar to the using account linking transactions.

In this solution:

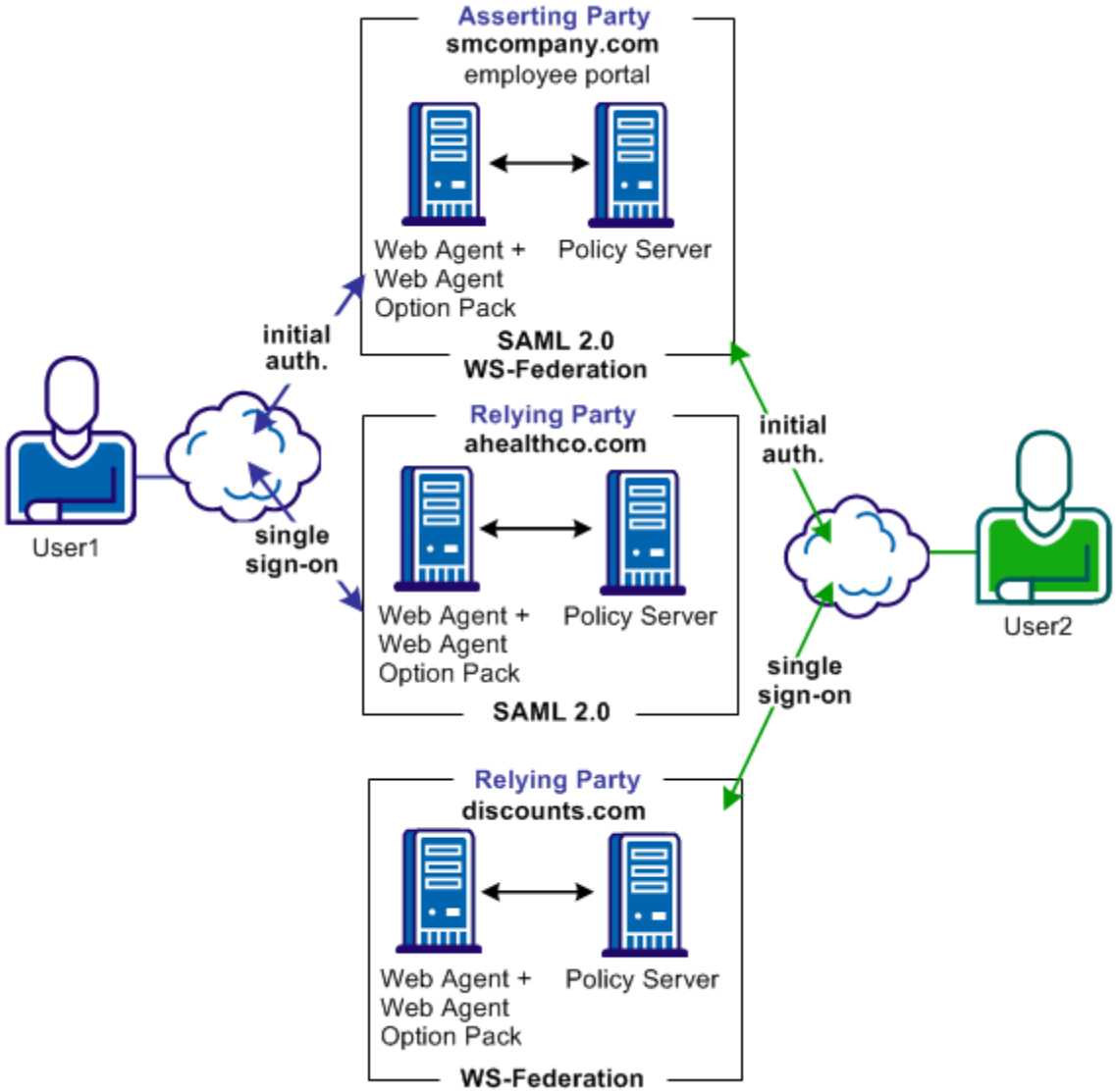
User 1

- smcompany.com is the SAML 2.0 Identity Provider for ahealthco.com.
- ahealthco.com is the SAML 2.0 Service Provider.

User 2

- smcompany.com is the WS-Federation Identity Provider for discounts.com
- discounts.com is the WS-Federation Resource Partner.

The following illustration shows a federated network that implements multiprotocol support.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

In this multiprotocol solution, the flow of the single sign-on transactions for the various SSO profiles is similar to the account linking SSO transactions.

- smcompany.com can issue a SAML 2.0 assertion for User 1 to access resources at ahealthco.com.
- Smcompany.com can also issue a token response that includes a SAML 1.1 assertion for User 2 to authenticate at discounts.com. The SSO profile for the assertion is determined by the partnership configuration, and based on the session cookie that is set during initial authentication.

For this solution, the following partnerships are configured at smcompany.com

- An IdP-to-SP partnership, where smcompany.com is the local IdP and ahealthco.com is the remote SP.
- An IP-to-RP partnership, where smcompany.com is the local IP and discounts.com is the remote RP.

The following partnership is configured at ahealthco.com:

- An SP-to-IdP partnership, where ahealthco.com is the local SP and smcompany.com is the remote IdP.

The following partnership is configured at discounts.com:

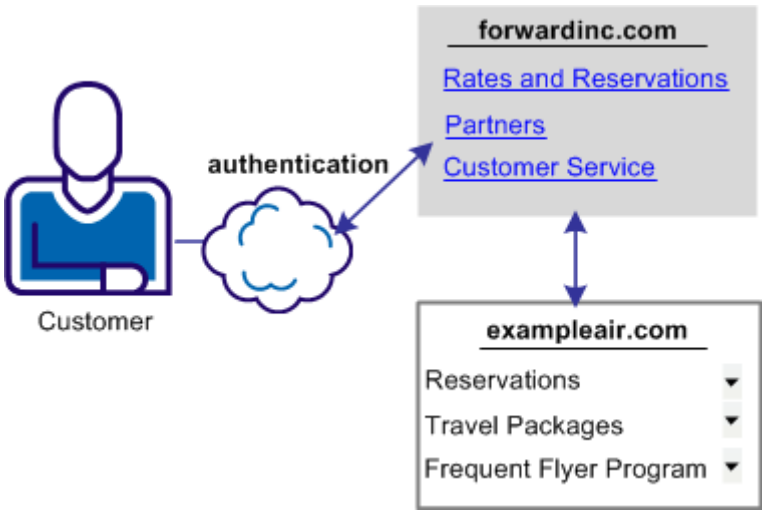
- An RP-to-IP partnership, where discounts.com is the local RP and smcompany.com is the remote IP.

Use Case: SAML 2.0 User Authorization Based on a User Attribute

In this use case, forwardinc.com is a car rental service and exampleair.com is a travel agency.

A customer of forwardinc.com logs in and authenticates at forwardinc.com, then clicks a link at the site to get a quote for a car rental. The customer profile at forwardinc.com includes the customer frequent flyer number for exampleair.com. The frequent flyer account determines a certain status level at forwardinc.com. The status level determines which discount offers the customer receives for car rentals.

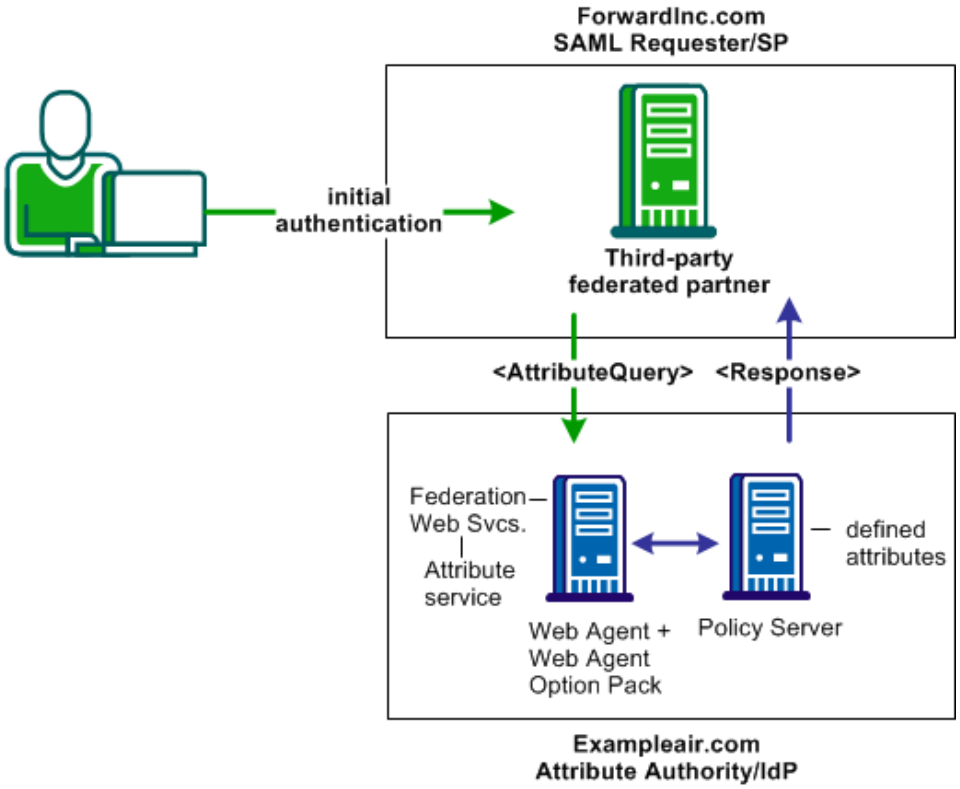
The following illustration shows this use case.



Forwardinc.com wants to present the appropriate discount information to its customers. However, it does not want customers to log in and authenticate at exampleair.com first then having to log in again at its site.

Solution: SAML 2.0 User Authorization Based on a User Attribute

The SAML 2.0 Attribute Query/Response profile can solve Use Case: SAML 2.0 User Authorization Based on a User Attribute (see page 36).



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

In this deployment,

- CA SiteMinder® is deployed only at Example.air, the IdP/Attribute Authority. The Web Agent with the Web Agent Option Pack is installed on one system and the Policy Server on another system.

  Note: CA SiteMinder® must be serving as an IdP to implement the attribute query profile. This means that CA SiteMinder® can only be an Attribute Authority and respond to attribute queries. CA SiteMinder® cannot serve as an SP and cannot send attribute queries.

- Forwardinc.com is a third-party Service Provider that is configured to use the Attribute Query/Response profile.

Forwardinc.com is acting as a SAML Requester. When a customer logs in at this site, the following sequence of events occurs:

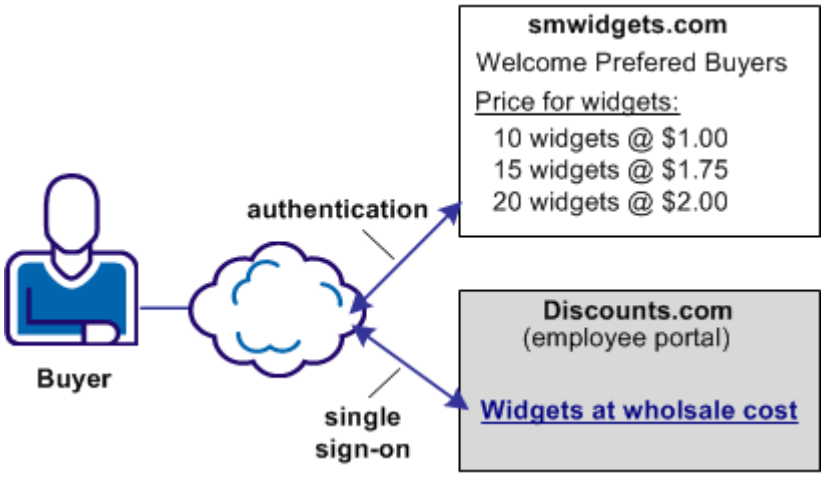
1. The user logs in to forwardinc.com and authenticates the user.
2. The user clicks a link to rent a car. Forwardinc.com identifies an unresolved frequent flyer attribute.
3. Forwardinc.com tries to resolve the attribute by looking up the user in the local user directory, but it cannot resolve the user attribute variable.
4. Forwardinc.com sends an attribute query as a SOAP request to the IdP/Attribute Authority, exampleair.com. The query request contains the frequent flyer attribute.
5. Exampleair.com looks at its own user directory record for the user and resolves the frequent flyer attribute. Exampleair.com returns an assertion as a SOAP response to forwardinc.com. The assertion contains the requested attribute.
6. The SAML Requester resolves the attribute and authorizes the user for the requested resource.
7. The user is redirected to the target resource.

Use Case: Single Sign-on with No Name ID at the IdP

In this use case, discounts.com purchases widgets from smwidgets.com.

A buyer for discounts.com clicks on a link to access the latest widget price list at smwidgets.com. The buyer is taken to the smwidgets.com website and presented with the price list without having to log in to the discounts.com website.

The following illustration shows this use case.



There are no buyer identities stored locally at discounts.com, so discounts.com wants to obtain an identity for buyers at smwidgets.com. Discounts.com sends an authentication request to smwidgets.com. Smwidgets.com receives the request, but it cannot find a value for the NameID attribute. It generates a unique persistent identity for the buyer and adds this identity to the assertion. Discounts.com uses this unique identifier to allow the buyer access to the requested resource.

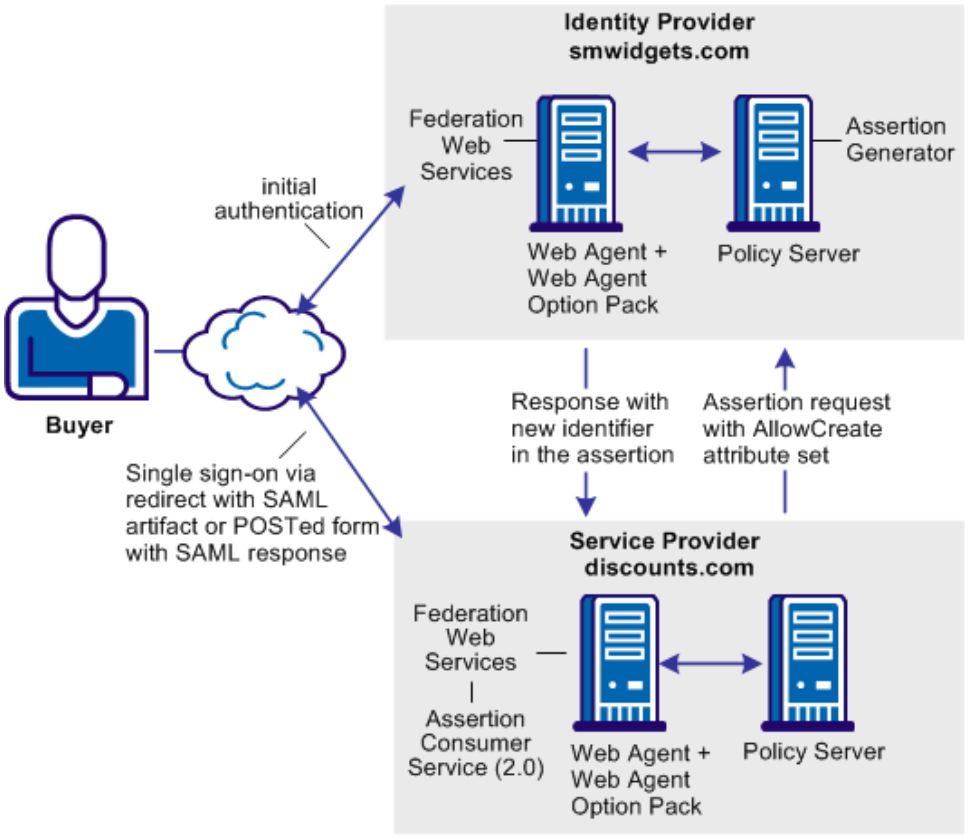
Solution: Single Sign-on with No Name ID at the IdP

Use of the Allow/Create attribute solves Use Case: Single Sign-on with No Name ID at the IdP (see page 39).

NOTE: This solution requires the SAML 2.0 profile.

Federation is deployed at discounts.com and smwidgets.com. A Web Agent and Web Agent Option pack is installed on one system, and the Policy Server is installed on another system.

In the following illustration, smwidgets.com is the Identity Provider and discounts.com is the Service Provider.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

To enable single sign-on between the two sites with no user identity at the Identity Provider, the following sequence of events occurs.

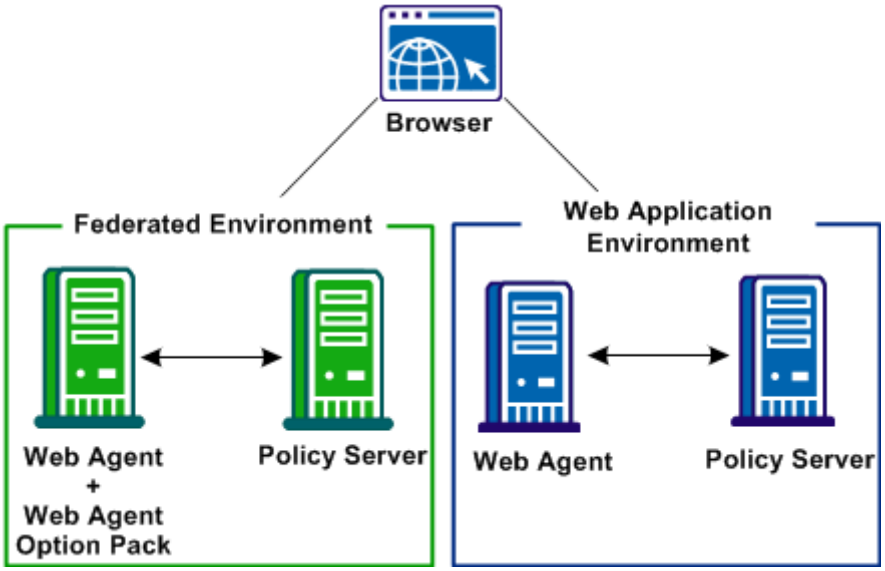
1. The user, in this case the buyer, authenticates at discounts.com. This link initiates an authentication request.
2. The Policy Server at discounts.com checks for presence of the Allow/Create option in the configuration. The option is enabled in the SP-to-IdP partnership at the discounts.com.
3. An attribute named AllowCreate is included in the authentication request. The Federation Web Services application at the local Web Agent redirects the authentication request to smwidgets.com.

4. The Policy Server at smwidgets.com generates an assertion. During the assertion generation, the Policy Server searches for the NameID attribute in the user record of the user requesting the resource. There is no value for the NameID in the user record.
5. The Policy Server verifies its configuration for the AllowCreate option. The Policy Server also checks the authentication request for the AllowCreate attribute, which it finds.
6. The Policy Server generates a unique persistent identifier because of the presence of the AllowCreate attribute in the authentication request and in its own configuration. The Policy Server places the identifier in its user store.
7. The Policy Server returns the assertion to the Federation Web Services at the application at the discounts.com. The Federation Web Services application sends an auto-post form containing the assertion response to the Assertion Consumer Service at discounts.com.
8. The Service Provider at discounts.com uses the response message to log in to the Policy Server, using the response as credentials.
9. The Policy Server validates the response by looking for the NameID in its user store. The Policy Server locates the user and logs in the user.
10. The Web Agent generates an SMSESSION cookie for the discounts.com domain.
11. The Web Agent places the cookie in the browser and redirects the user to the target destination.

Use Case: SSO Using Security Zones

In this use case, CompanyA protects non-federated web applications and supports federated single sign-on. In a CA SiteMinder® deployment, you cannot have two sessions for a single user that travels from the web application environment to the federated environment. As the user navigates between each environment, the session cookies would overwrite one another.

The following illustration shows a site that combines a federated environment and a web application environment.

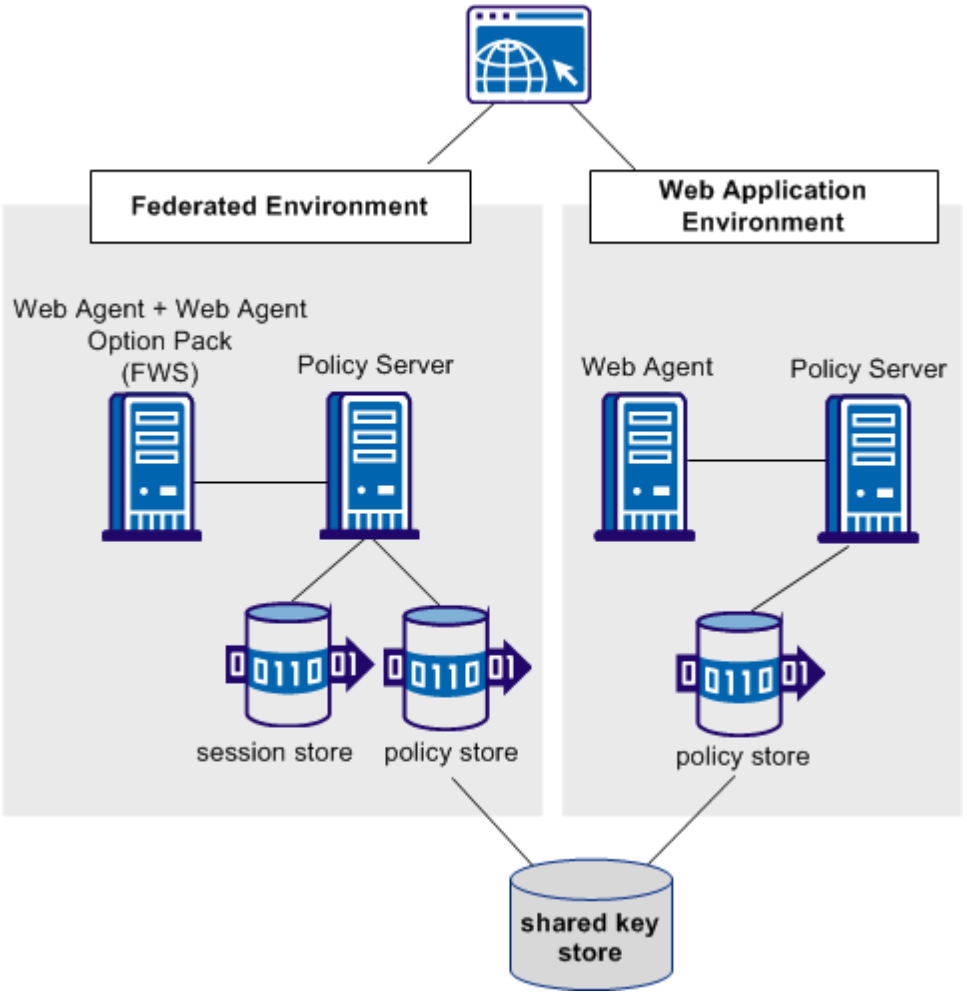


Solution: SSO Using Security Zones

This solution illustrates how security zones lets you set up a parallel web application and a federation environment to solve the Use Case: SSO Using Security Zones (see page 42).

A security zone is a segment of a single cookie domain, which is used to partition applications. You can assign different security requirements to each zone. Security zones lets the Policy Server generate different session cookies for a single user in each environment. Though two uniquely named session cookies are generated, each cookie represents the same session across the web application and federated environments. Web agents at the asserting party enforce security zones.

The following figure illustrates a deployment with two different environments at a single asserting party. One environment is for federation functionality and the other is for web application protection.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The figure reflects the following setup:

Web Application Environment

Agent Configuration Object or Local Configuration file

DefaultAgent

Trusted Security Zone

SM (default zone)

Cookies the Web Agent reads for the zone

The DefaultAgent configuration enables the Web Agent to read and write the default session cookie, SMSESSION.

Federation Environment

Agent Configuration Object or Local Configuration file

FedWA

Trusted Security Zones

- Fed (default zone)
- SM

Cookies the Web Agent reads for the zone

The FedWA configuration enables the Web Agent to read and write SMSESSION cookies.

Note: For this solution to work, each environment must have its own Agent Configuration Object.

The following sequence of events follows:

1. The user logs in to the federation environment.
2. The Web Agent in the federated environment directs a request to the Authentication URL to establish a user session.

   The user already has an SMSESSION cookie from a prior authentication in the web application environment.
3. The Web Agent in the federated environment reads the SMSESSION cookie. The Policy Server generates a new federation session cookie and the Web Agent writes this new session cookie to the browser. The new federation session cookie is based on the SMSESSION cookie.

   Federation requires a persistent session, which is stored in the session store. The SMSESSION cookie that is read from the web application environment is not persistent. When the Policy Server generates a federation cookie, it modifies the cookie and upgrades the session to be a persistent session.

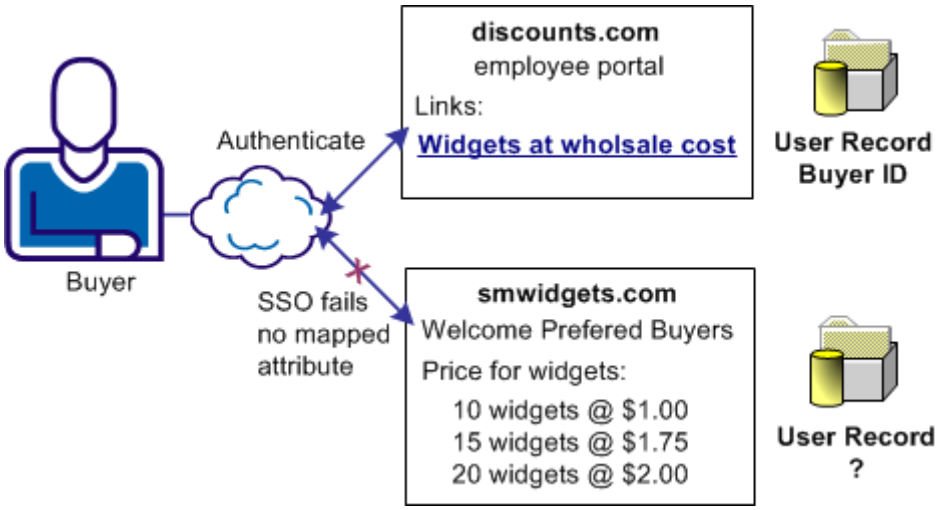
4. The FWS application in the federated environment reads the federation cookie and successfully processes the request for the resource.

Use Case: SSO with Dynamic Account Linking at the SP

In the use case, the IdP, discounts.com, includes an attribute named buyer ID that identifies a particular user. The buyer ID value is entered as the NameID in the assertion. However, there is no mapped identity at smwidgets.com for the buyer ID. Smwidgets.com must create an attribute in the appropriate user record so that the buyer can authenticate and gain access to the protected resource.

The administrator at the smwidgets.com establishes a mapping using dynamic account linking. The mapping lets smwidgets can authenticate the buyer and can allow access to resources. When a buyer at discounts.com selects a link to access the latest price list on widgets at smwidgets.com, the buyer is logged in without having to reauthenticate.

The following illustration shows this use case.



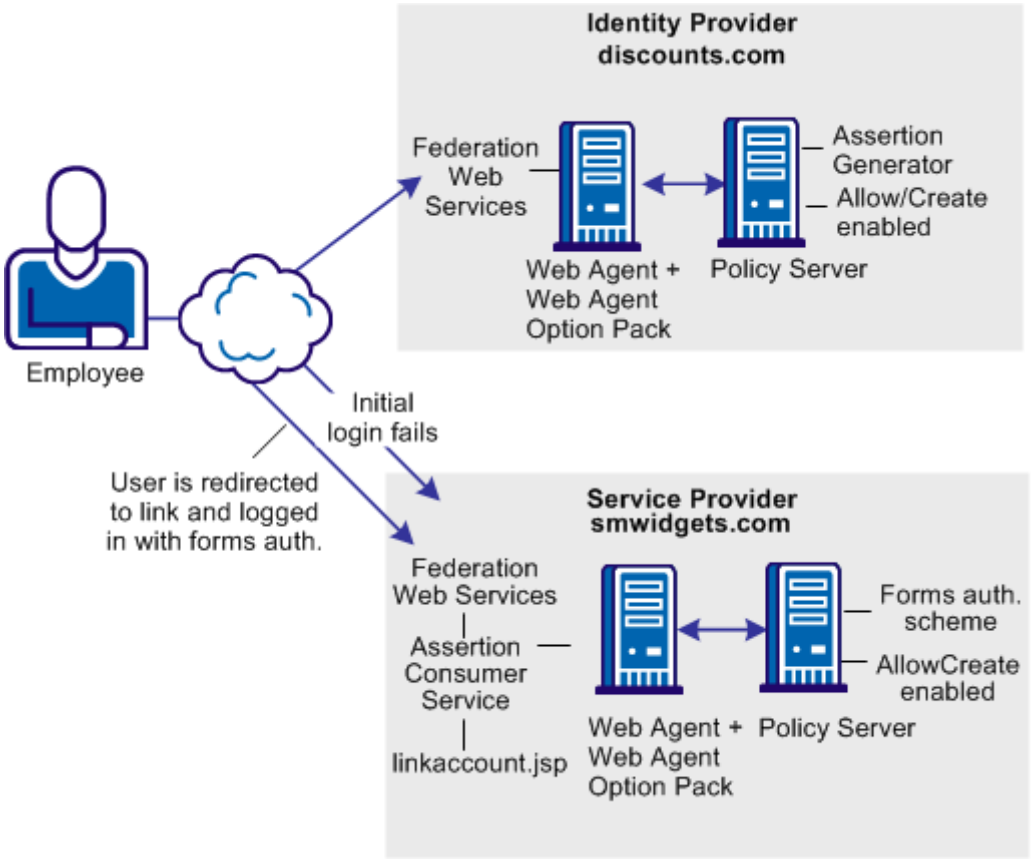
Solution: SSO with Dynamic Account Linking at the SP

Federation can be deployed at IdPA.com and SPB.com to solve Use Case: SSO with Dynamic Account Linking at the SP (see page 46).

Note: Dynamic account linking is only supported with SAML 2.0.

CA SiteMinder® is deployed at both sites. Each site has a Web Agent and Web Agent Option Pack installed on one system, and the Policy Server on another system.

The following illustration shows single sign-on with dynamic account linking at the Service Provider.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The following sequence of events occurs:

1. The employee initially logs in and authenticates at discounts.com. Discounts.com creates an assertion for the employee. Discounts.com posts the assertion (POST binding) or redirects the user with an artifact (artifact binding) to the Assertion Consumer Service at smwidgets.com. This assertion includes an attribute named buyerID.
2. The Assertion Consumer Service at smwidgets.com tries to authenticate the user, but the buyerID attribute does not map to a local user record. The authentication fails.

3. As part of the partnership configuration at smwidgets.com, a redirect URL is defined, which points to the directory *web_agent_home*/affwebservices/linkaccount.jsp. The employee is redirected to this URL.

   Note: The linkaccount.jsp file must be part of a protected realm. The default location for this file is http://*sp_home*/affwebservices/public/. Copy the file from this location to a protected realm.

4. A Web Agent that authenticates the local user with the forms authentication scheme protects this linkaccount.jsp URL. After a successful authentication, a session at smwidgets.com is established and an SMSESSION cookie is placed in the employee's browser.

5. The linkacount.jsp gets loaded in the browser and the user sees a message to link to the Service Provider account. Click on the button to permit the account linking.

6. The user is redirected to the Assertion Consumer Service, where the browser of the employee presents the SMSESSION cookie with the assertion.

7. The Assertion Consumer Service extracts the NameID from the assertion and inserts the NameID value into a newly created buyerID attribute. The buyerID attribute is in the existing user record of the employee. The Assertion Consumer Service knows which user record to map because the UserDN in the SMSESSION cookie identifies the user.

   The search specification configured in the SAML 2.0 partnership indicates which attribute is mapped to the NameID. In this case, the search specification is buyerID=%s.

8. After the attribute is mapped, the user is authenticated based on the assertion. A new user session is established.

   The next time that the same user presents an assertion with the buyer ID, the user successfully gains access to the requested resource.

Configure Dynamic Account Linking at the SP

Configure the following components at the Service Provider to enable dynamic account linking.

Note: Dynamic account linking is only supported with SAML 2.0.

- AllowCreate feature

  Enables the creation of attributes in an existing user store.

- Redirect URL

  Sends the user to the linkaccount.jsp file when authentication fails. An authentication scheme protects the redirect URL. The scheme requests the user to log in to create a session.

- Post Preservation at the Web Agent

  Must be enabled at the Service Provider Web Agent.

- Search Specification

  Indicates which attribute the NameID from the assertion replaces

Enable dynamic account linking for SAML 2.0 POST or Artifact single sign-on at the Service Provider

Follow these steps:

1. For the linkaccount.jsp file, do the following:
   - (Optional) Customize the linkaccount.jsp file to provide a custom user experience when the user is redirected after a failed authentication attempt. This file must POST the **accountlinking** and **samlresponse** parameters back to the Assertion Consumer Service URL.

     Note: The accountlinking must be set to yes (accountlinking=yes).

     The default location for this file is http://*sp_home*/affwebservices/public/.

   - Protect the linkaccount.jsp file with a CA SiteMinder® forms authentication scheme, which supports POST-Preservation. The SAML response that contains the assertion is posted to the Assertion Consumer Service after the user has logged in locally at the Service Provider. Preserve the SAML response POST data during the entire local authentication process.

     To protect resources with an authentication scheme, refer to information about authentication schemes in the *Policy Server Configuration Guide*.
2. Enable the Allow/Create feature at the Service Provider.
3. For the Web Agent at the Service Provider, set the POST Preservation parameter to yes. This setting enables the POST data from the SAML response to be preserved.
4. Configure a redirect URL that sends the user to the linkaccount.jsp file if authentication fails. Direct the user only to this file.

   The redirect URL is part of the SAML 2.0 authentication scheme setup at the Service Provider.

Complete the following fields with the values shown:

Redirect URL for the User Not Found Status

http://*sp_home*/*protected_realm*/linkaccount.jsp

Example: http://smwidgets.com/partner_resources/linkaccount.jsp

The default location of the linkaccount.jsp file is http://*sp_home*/affwebservices/public/. Copy the file from this directory to a directory that is configured as a protected realm.

Mode

HTTP POST

5. Configure a search specification for the SAML authentication scheme. For example, if the Name ID from the assertion replaces buyerID, the search specification would be buyerID=%s.

Chapter 3: Federation Deployment Considerations

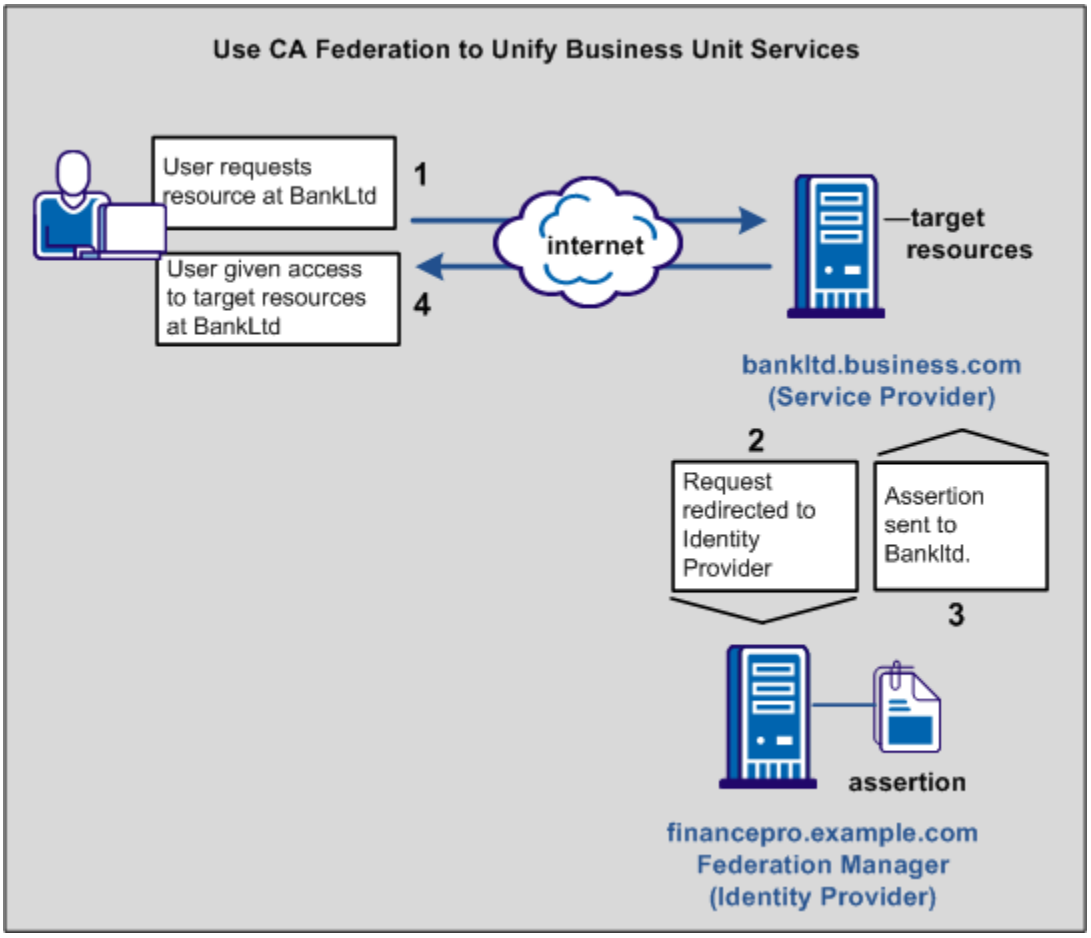
Federation Business Case

A sample business case best illustrates how CA SiteMinder® Federation can solve a common business problem.

In this business case, Financepro is a financial planning firm that recently bought the banking firm BankLtd to provide private banking to its clients. These two companies have different information infrastructures, but they want to appear as one company to their customers. To solve this problem, they set up a federated partnership.

By establishing a federated relationship, the two companies can provide a seamless customer experience using single sign-on. Customers can travel between Financepro and BankLtd without constantly being challenged to authenticate. Additionally, the sharing of customer identities and customer information can further customize the user experience and cross-promote the financial products of each partner.

The following illustration shows the federated partnership between Financepro and BankLtd. The flow of communication is based on a SAML 2.0 Service Provider-initiated single sign-on.



The illustration describes the following information flow:

1. The user tries to access a federated resource at BankLtd.
2. The user is redirected to the Financepro for authentication and the assertion is generated.
3. The assertion is passed back to BankLtd.
4. Single sign-on occurs based on either a SAML HTTP-Artifact or HTTP-POST. The user gets access to the target resource.

For this partnership to work, decide how the partnership functions before implementing the relationship using federation.

The issues to consider include:

- How users are identified across the partnership.
- What attributes get sent in an assertion and for what purpose.
- Which federation binding to use (SAML POST or Artifact, WS-Federation).

Your decisions help structure the business partnership.

User Identification Across the Partnership

Business partners have their own method of defining user identity in their respective user stores. How users are identified determines how one partner can map its users to the other partner.

Consider the following scenarios:

- The User ID is the same at the user store of each site.

  Account linking is the method of user identification.

- The User ID is unique at the user store of each site.

  Identity mapping is the method of user identification. At FinancePro, a customer is identified as JohnDoe, while at BankLtd this same customer is identified as DoeJ. The partners must agree on a user attribute profile to use for identity mapping.

- The User ID does not exist at the relying party.

  Account provisioning is the method for user identification. Provisioning an account can require creating an account for a user or simply populating an existing user account with information in the SAML assertion.

The user identification decision determines what information is sent as the user identity in the assertion.

User Mapping

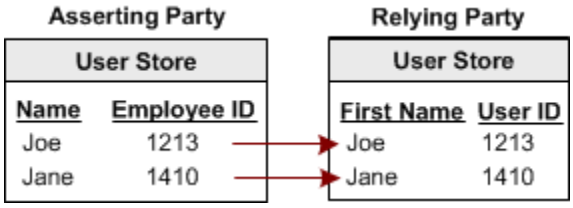
User mapping is the ability to establish a relationship between a user identity at one business and a user identity at another business. Map remote users at the asserting party to local users at the relying party.

The types of mapping are as follows:

- One-to-one mapping maps a unique remote user directory entry at the producing authority to a unique user entry at the consuming authority.

  One-to-one mapping, or account linking, links an account at the asserting party to an account at the relying party.

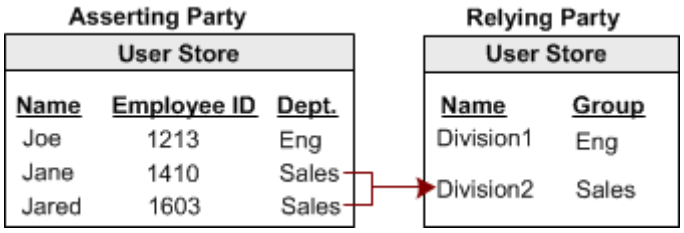
  The following illustration shows one-to-one mapping.



- N-to-one mapping maps a group of remote user directory entries to a single local profile entry.

  N-to-one mapping allows several user records at a producing authority to be mapped to one user record or profile at a consuming authority. An administrator at the relying party can use n-to-one mapping for a group of remote users without maintaining a record for each remote user.

  The following illustration shows n-to-one mapping:



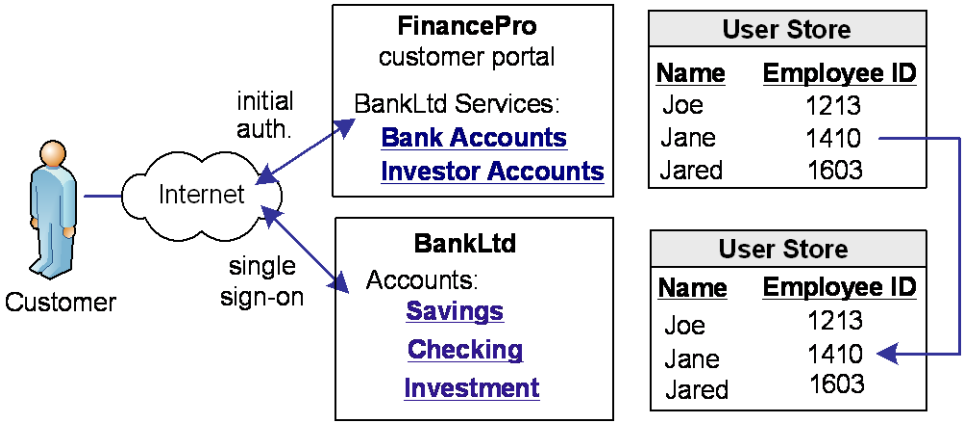
For legacy federation, user mapping is configured as part of a federation authentication scheme. For partnership federation, user mapping is configured as part of the name ID and attribute settings.

Account Linking to Establish a Federated Identity

When a customer at FinancePro accesses a resource at BankLtd, the NameID is always in the assertion. This identifier allows BankLtd to determine who the customer is and the level of access to allow for that customer.

The NameID can establish a federated identity when the user store at each partner identifies the users in the same way with the same ID.

The following figure shows the user store at each site with the same employee IDs.



CA SiteMinder® Federation lets you configure account linking as part of the partnership configuration process. You specify a NameID format and Name ID type, which determines the type of value that defines the Name. You associate the specific Name ID type, with a static, user, or DN attribute from a user directory. The NameID that CA SiteMinder® Federation includes in the assertion conforms to the configuration you define.

When the relying party receives the assertion, the user disambiguation process at BankLtd occurs. The process links the NameID value in the assertion to a record in its user store.

Identity Mapping to Establish a Federated Identity

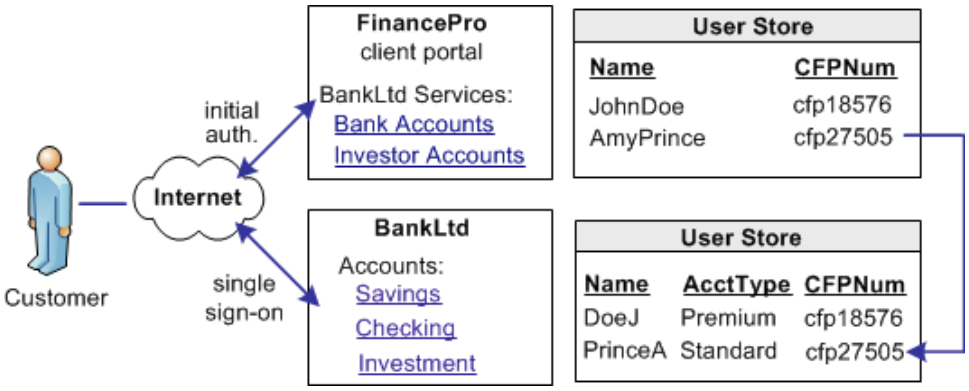
An investor at Financepro authenticates and selects a link to access information at BankLtd. The investor is taken directly to the accounts area of the BankLtd website without having to sign on.

BankLtd maintains user identities for all customers at Financepro, but the identities differ from the identities at FinancePro. For example, at FinancePro, JohnDoe is a customer. At BankLtd, this same customer is identified as DoeJ. Regardless, BankLtd must control access to sensitive portions of the company website. To establish the federated identity, the partners agree on an attribute that maps to the appropriate identity for a single customer at either site.

The partners agree on which attribute to use during an out-of-band exchange of information, meaning that the agreement is not part of any communication in any message over a channel. For this example, the attribute that the partners agree upon is a certified financial planner license number, referred to as the CFPNum in each user store.

When a customer tries accessing the federated resource at BankLtd, the request triggers the single sign-on process. The assertion that is generated at FinancePro contains the CFPNum attribute. When BankLtd receives the assertion, an application at its site has to perform the user disambiguation process. The process relies on the attribute to determine which profile identity is used for the request.

The following illustration shows how the same users are identified differently at each partner.



CA SiteMinder® Federation lets you configure identity mapping as part of the partnership configuration process. For the NameID and attribute configuration, you define an attribute called CFPID. Associate this attribute with the user attribute CFPNum, which is the name of the attribute in the user store at each partner.

CA SiteMinder® Federation includes the attribute in the assertion. When BankLtd receives the assertion, the user disambiguation process links the attribute in the assertion to the appropriate record in its user store.

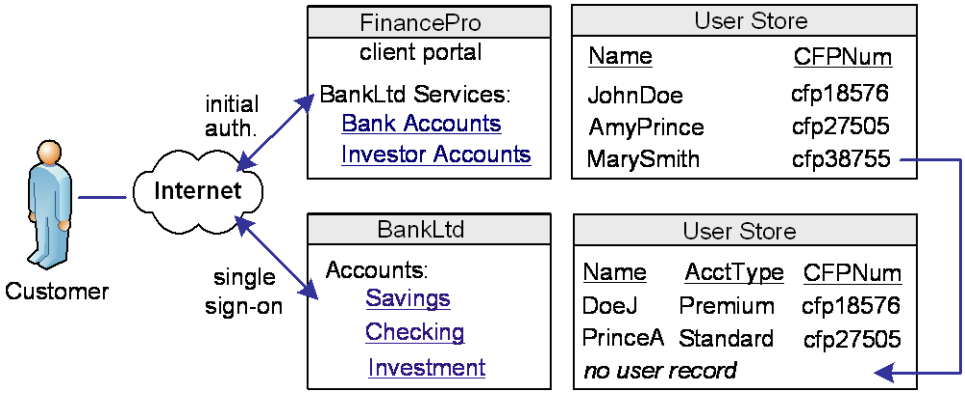
User Provisioning to Establish a Federated Identity (partnership federation only)

Partnership federation can work with provisioning applications at the relying party to establish an identity.

A client at Financepro, Mary Smith, authenticates and clicks a link to access information at BankLtd. Initially, BankLtd cannot find a user account for Mary Smith. BankLtd wants to protect sensitive portions of its website while allowing new customers.

BankLtd has configured federation to implement provisioning to establish the new federated identity for Mary Smith. CA SiteMinder® redirects Mary Smith to the provisioning server at BankLtd. The provisioning application, using the federated identity information, creates a user account in the user store.

The following illustration shows the user stores at FinancePro and BankLtd.



Federation lets you configure provisioning as part of the partnership configuration at the relying party. In this example, you select remote provisioning and determine how assertion data is delivered to the BankLtd provisioning server. This configuration enables the dynamic creation of a user entry in the user store.

Attributes for Customizing an Application

CA SiteMinder® Federation offers two ways of using attributes to customize target applications.

Attributes Added to Assertions at the Asserting Party

You can include attributes from a user store record in an assertion to identify a user for the purpose of customizing an application.

Servlets, web applications, and other custom applications can use attributes to display customized content or enable and disable other custom features. When used with web applications, attributes can implement fine-grained access control by limiting user activity at the target site. For example, you send an attribute variable named Account Balance and set it to reflect the account holdings of the user at BankLtd.

Attributes take the form of name/value pairs. When the relying party receives the assertion, it makes the attribute values available to applications.

Attribute Mapping at the Relying Party

The relying party receives a set of assertion attributes, which can be mapped to a set of application attributes being delivered to the target application.

For example, FinancePro includes an assertion attribute CellNo=5555555555. At BankLtd, this attribute name is transformed to an application attribute Mobile=5555555555. The attribute name is converted but the value remains the same.

Multiple assertion attributes can also be transformed into a single application attribute. For example, FinancePro sends an incoming assertion with the attributes Acct=Savings and Type=Retirement and transformed at BankLtd into FundType= Retirement Savings.

Federation Profile for Single Sign-on

Determining whether to use SAML or WS-Federation for a partnership depends on the binding that each side supports.

For a new federation, there are no legacy requirements for either partner. Therefore, the recommended SAML profile to use for single sign-on is SAML 2.0 POST profile. SAML 2.0 POST profile offers secure transmission of assertion data and the configuration process is simpler than SAML Artifact profile. If, however, the agreement of two partners requires SAML Artifact, this binding can also be implemented.

For deployments use Active Directory Federation Services (ADFS), configure WS-Federation.

Federating with Each CA SiteMinder® Federation Model

The legacy federation or partnership federation model can establish a federated partnership between Financepro and BankLtd. Using federation, users move between each company as if they are one company.

Partnership Federation Model

Configure the partnership model in the Administrative UI, guided by a partnership wizard. The partnership objects focus on creating partnerships and identifying each side of the partnership to accomplish single sign-on.

These steps in the partnership wizard include:

1. Configuring a Partnership

   Names the partnership and identifies the two entities that make up the partnership.

2. Establishing the Federation Users/User Identification

   Identifies the users for which the asserting party generates assertions/tokens and the relying party authenticates.

3. NameID and Attributes

   Determines how a federated identity is established and lets you add attributes to identify and customize the content of the assertion.

   Using the NameID and attributes, you can verify that the appropriate information is available to the application at the relying party. The NameID and Attributes step is where you configure account linking and identity mapping.

4. SSO and SLO or Sign-out

   Defines the Single Sign-on binding, including the location of the service consuming assertions at the relying party. For SAML 2.0, you can configure more features, such as single logout (SLO), authentication context, Enhanced Client or Proxy (ECP) profile, and Identity Provider Discovery profile. For WS-Federation, you can configure sign-out.

5. AuthnContext (SAML 2.0 only)

   Enables the Service Provider to obtain information about the authentication process to establish a level of confidence. This feature also enables the Identity Provider to include the authentication context in an assertion.

6. Signature and Encryption

   Defines the signature and encryption options for secure exchange of data, including:

   - assertions
   - authentication requests
   - SAML 2.0 single logout requests and responses
   - WS-Federation sign-out responses.

7. Application Integration

   Enables you to configure redirection to the target application, lets you set up provisioning of user records, and define relying-party side attribute mapping. You can also set up redirects for failed user authentication.

Legacy Federation Model

The legacy federation model focuses on the domain, realm, rule, authentication schemes, and policy objects.

If CA SiteMinder® is the asserting party, the configuration steps include:

1. Configuring an entity in an affiliate domain

   Names the partner for which the asserting party generates assertions.

2. Establishing federation users

   Specifies the user directories for which the asserting party generates assertions and the relying party authenticates.

3. Selecting profiles (SAML or WS-Federation) for transactions

   Determines how a federated identity is established. In the profiles configuration, you add attributes to identify and customize the content of the assertion.

   Using NameID and attributes, you can verify that the appropriate information is available to the application at the relying party. The profiles configuration is where you specify account linking and identity mapping.

   As part of the profiles, configure single sign-on. For SAML 2.0, you can configure more features, such as single logout (SLO), Enhanced Client or Proxy (ECP) profile, and Identity Provider Discovery profile. For WS-Federation, you can configure sign-out.

4. Signature processing and encryption (SAML 2.0)

   Defines the signature options for secure exchange of assertions, authentication requests, and single logout requests and responses.

If CA SiteMinder® is the relying party, the configuration steps include:

1. Setting up SAML and WS-Federation authentication schemes

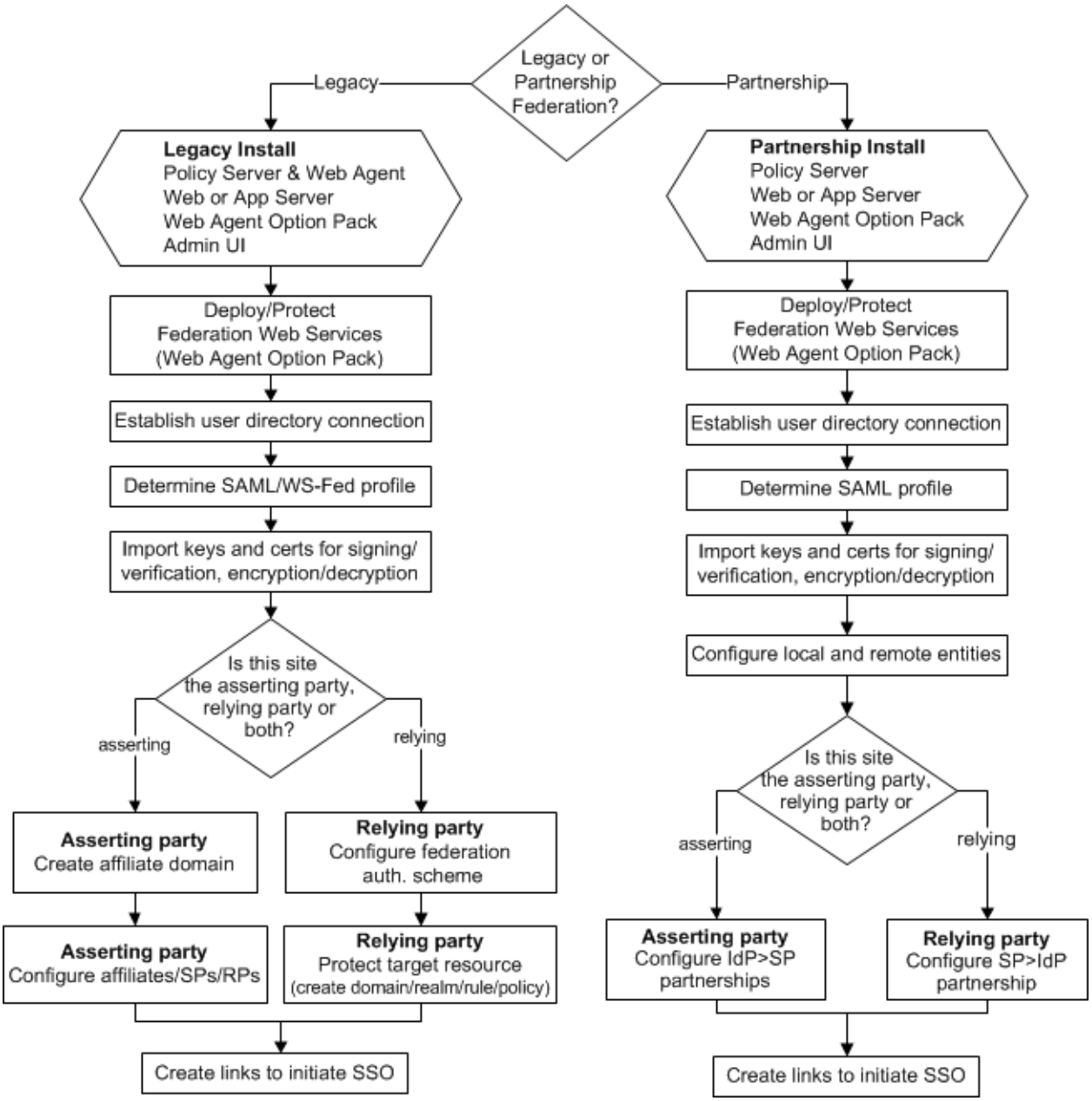
   Enables you to configure redirection to the target application, lets you set up provisioning of user records, and define relying-party side attribute mapping.

2. Configuring federation-specific settings included with the authentication scheme, such as single sign-on, single logout, sign-out, encryption, and decryption.

Federation Flow Diagram

Configure the components to establish successful federated partnerships. Most of these components are configurable using the Administrative UI.

The following flow chart highlights the general process for legacy federation and partnership federation.



See the following guides for detailed instructions on required components and configuration procedures:

Partnership federation

Partnership Federation Guide

Partnership Federation refers to partnership model of federation.

Legacy federation

Legacy Federation Guide

Legacy federation refers to the product known as Federation Security Services

Chapter 4: Comparing Federation and Web Access Management for Single Sign-on

Advantages of Federation and Web Access Management

Federation and web access management (WAM) offer different benefits for single sign-on. Determining when to use federation or WAM single sign-on is dependent on your deployment.

Federation allows you to expand on your WAM capabilities; it does not replace those capabilities.

Federation has the following advantages:

- Many applications can handle federation directly out-of-the-box, such as SAP, SharePoint, WebLogic. These applications accept assertions.
- A direct connection to a centralized server is unnecessary. A federation request always goes through the asserting party to get the generated assertion. After a user gains access to content on one server, the user returns to the federation hub and gets redirected to the next server. Only if the user session times out at the hub does the user have to reauthenticate.
- Two models of CA SiteMinder® federation. Partnership federation is business-centric, emphasizing relationships with partners. Legacy federation is protocol-centric and more customizable to the protocol specification.

These advantages make federated partnerships better for an environment where sites are remote, inaccessible, or under third-party control.

CA SiteMinder® WAM single sign-on has the following advantages:

- Transactions are faster because there are fewer browser redirects.
- CA SiteMinder® provides centralized authorization and auditing.
- Direct links can exist from one web server to another in a network without the user going through a centralized hub for assertion generation.
- CA SiteMinder® offers timeout management.
- Applications are independent of a remotely initiated transaction.

These advantages make WAM single sign-on better suited to an environment with sites that are under your control, such as internal data centers.

Deployments that Favor Federation

Federation is advantageous in networks where your company does not control the server. For example, a third party owns the web server and does not allow you to install a web agent on the server. Also, when a remote server is in a location where there is a high network latency between the web agent and Policy Server. When you have no control over the target server, a SAML assertion is an ideal way to pass identity information.

The partners in a federated network follow the specific standards for the protocol used in communications. The common standards make the generating and consuming of assertions universal. The result is that the vendor at the asserting or relying party is not important nor is the remote location of each vendor.

Finally, federation is a good solution when timeouts are not a major concern, and obtaining identity information is the goal. External authorization checking is not a focus of federation.

Deployments that Favor Web Access Management

WAM single sign-on works best in an environment where you have control over each website. Having CA SiteMinder® in the same data center as the website or other internal single sign-on environments are good deployments for web access management. Controlling over each website is also important for auditing your network performance and monitoring timeout issues.

WAM single sign-on lets you integrate with an application by way of a WAM session. WAM implementations also reduce some of the performance issues inherent with federation. For example, a transaction that is initiated by an asserting party can require several redirects after a user selects a link to make a request.

Chapter 5: Federation Web Services

Federation Web Services Overview

The Federation Web Services (FWS) application is installed with the Web Agent Option Pack on a server that has a connection to a Policy Server. The Federation Web Services and the Web Agent support the following web browser single sign-on profiles. These profiles convey information from one site to another through a standard browser.

The supported profiles are:

- SAML artifact profile 1.0 (legacy federation only)
- SAML artifact profile 1.1 and 2.0 (legacy federation and partnership federation)
- SAML POST profile 1.x and 2.0 (legacy federation and partnership federation)
- WS-Federation Passive Requestor profile (legacy federation and partnership federation)

SAML 1.x Artifact and POST Profiles

For the SAML 1.x artifact and POST profiles, the Federation Web Services application uses the following services:

Assertion Retrieval Service (SAML 1.x Artifact only)

A producer-side component. This service handles a SAML request for the assertion that corresponds to a SAML artifact by retrieving the assertion from the CA SiteMinder® session store. The SAML specification defines the assertion retrieval request and response behavior.

Note: Only the SAML artifact profile uses the assertion retrieval service..

SAML Credential Collector (SAML 1.x)

A consumer-side component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The credential collector issues CA SiteMinder® cookies to a browser of the user.

Intersite Transfer Service (SAML 1.x)

A producer-side component for the SAML POST profile. The intersite transfer service transfers a user from the producer site to a consumer site. For the SAML artifact profile, the Web Agent performs the same function as the intersite transfer service.

SAML 2.0 Artifact and POST Profiles

For SAML 2.0 artifact and POST profiles, the Federation Web Services application uses the following services:

Artifact Resolution Service (SAML 2.0 Artifact only)

An Identity Provider-side service that corresponds to the SAML 2.0 authentication using the HTTP-artifact binding. This service retrieves the assertion stored in the CA SiteMinder® session store at the Identity Provider.

Note: Only the HTTP-artifact binding uses the artifact resolution service.

Assertion Consumer Service (SAML 2.0)

A Service Provider component that receives a SAML artifact or an HTTP form with an embedded SAML response and obtains the corresponding SAML assertion. The Assertion Consumer Service issues CA SiteMinder® cookies to a browser.

Note: The Assertion Consumer Service accepts an AuthnRequest with an AssertionConsumerServiceIndex value of 0. All other values for this setting are denied.

AuthnRequest Service (SAML 2.0)

This service is deployed for use by SAML 2.0. A Service Provider can generate an <AuthnRequest> message to authenticate a user for cross-domain single sign-on. This message contains information that enables the Federation Web Services application to redirect the browser to the single sign-on service at the Identity Provider. The AuthnRequest service is used for POST and Artifact single sign-on.

Single Sign-on Service (SAML 2.0)

The single sign-on service enables an Identity Provider to process AuthnRequest messages. The service also invokes the assertion generator to create an assertion that is sent to the Service Provider.

Single Logout Service (SAML 2.0)

This service implements processing of single logout functionality, which an Identity Provider or a Service Provider can initiate.

Identity Provider Discovery Service (SAML 2.0)

Implements SAML 2.0 Identity Provider Discovery Profile and sets and retrieves the common domain cookie. An IdP requests to set the common domain cookie after authenticating a principal. An SP requests to obtain the common domain cookie to discover which Identity Provider a principal is using.

WS-Federation Profile

For the WS-Federation profile, the Federation Web Services application uses the following services:

Security Token Consumer Service

A Resource Partner component that receives a security token and extracts the corresponding SAML assertion. The Security Token Consumer Service issues cookies to a browser.

Single Sign-on Service

Enables an Identity Provider to process a sign-on message and gather the necessary Resource Partner information to authenticate the user. This service also invokes the assertion generator to create an assertion that is sent to the Resource Partner.

Sign-out Service

Implements processing of a single sign-out transaction by way of a sign-out servlet. An Identity Provider or a Resource Partner can initiate sign-out.

Chapter 6: Federated Transaction Process Flows

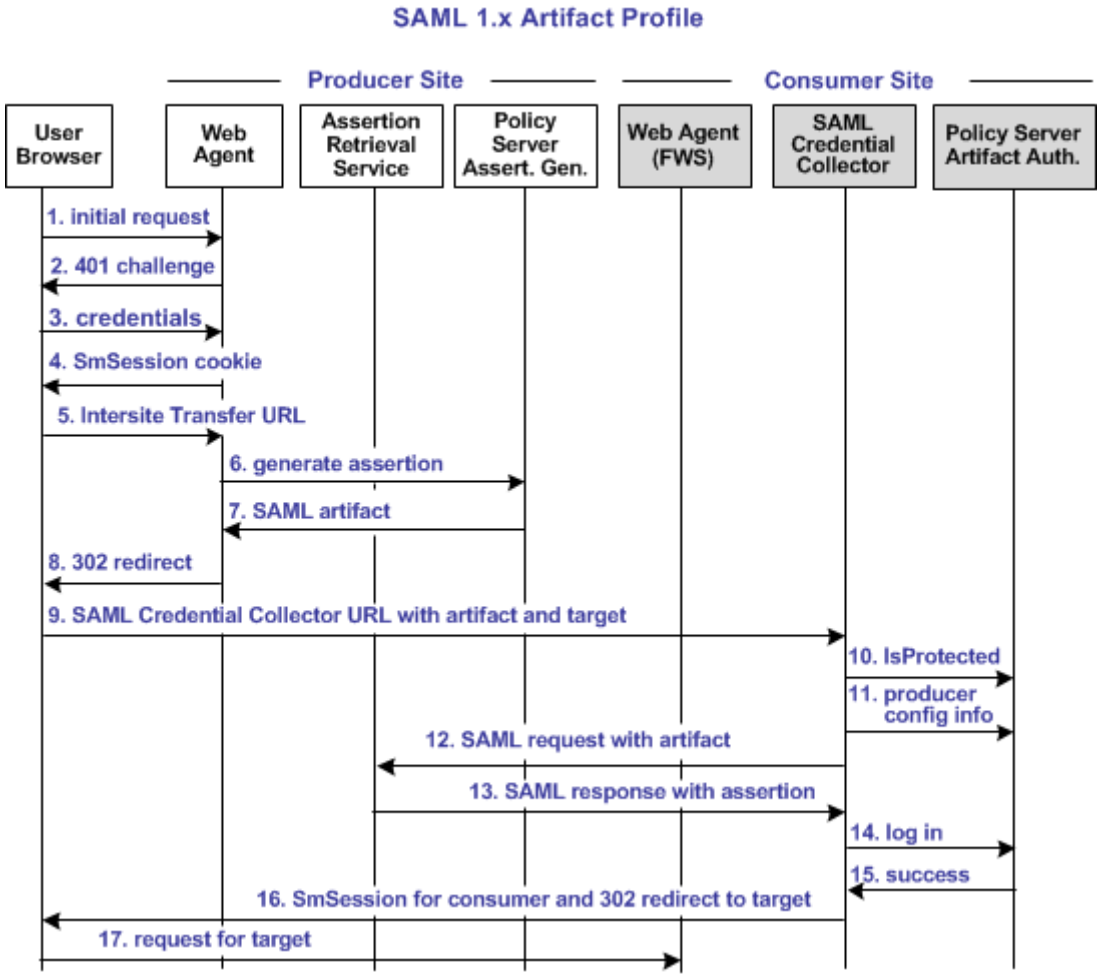
SAML 1.x Artifact SSO Transaction Flow (Producer-initiated)

The following illustration shows the flow between a user and the federation components at the producer and consumer sites. The flow shows single sign-on between the sites using the SAML 1.x artifact profile as the method for processing the SAML assertion.

The flow diagram assumes the following information:

- The Producer initiates the transaction.
- Successful authentication and authorization at each site.
- CA SiteMinder® is shown as the producer and consumer only so you can see the processes at each partner. If CA SiteMinder® is the Producer in your environment, review the producer activities in the table. If CA SiteMinder® is the consumer, review the consumer activities in the table.

The following diagram shows the SAML 1.x artifact SSO transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
User Agent (browser)	1. The user makes an initial request to a protected page at the producer site.
CA SiteMinder® as the Producer	2. The Web Agent at the producer site responds with a 401 challenge to the user for credentials.

Actor	Transaction Process
	3. The user submits credentials, such as the user name and password to the Web Agent.
	4. The Web Agent issues a SMSESSION cookie to the browser for the producer site domain, and allows access to the local page. **Log message:** Session cookie does not exist, redirecting to authentication url. **Checkpoint code:** [SSOSAML11_AUTHENTICATIONURL_REDIRECT]
	5. The user clicks a link on the local page to visit the consumer site. This link is the intersite transfer URL. The intersite transfer URL makes a request to the Web Agent at the producer. The Web Agent forwards an IsProtected call to the Policy Server. This URL contains the location of the SAML credential collector and the target URL at the consumer site. **Log messages:** SAML11 Consumer Configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML11_CONSUMERCONFFROMPS_REQ]
	6. The Web Agent makes a request to the Policy Server to generate the assertion. **Log messages:** Request to policy server for generating saml11 assertion/artifact based on selected profile. **Checkpoint code:** [SSOSAML11_GENERATEASSERTIONORARTIFACT_REQ]
	7. The Policy Server generates an assertion, places it in the session store, and returns the SAML artifact for the assertion. **Log message:** Policy server generates the saml11 assertion. **Checkpoint code:** [SSOSAML11_PSGENERATEASSERTION_RSP] **Log message:** Policy server stores the assertion in session store **Checkpoint code**: [SSOSAML11_PSSTOREASSERTIONINSSTORE_REQ] **Log message:** Policy server returns the wrappedassertion/artifact(based on profile selected) in response message **Checkpoint code:** [SSO_PSWRAPPEDASSERTION_RSP]
	8. The Web Agent responds with a 302 redirect to the SAML credential collector at the consumer. The redirect contains the artifact and the target URL as query parameters. **Log message:** Sending artifact to credential collector service url. **Checkpoint code:** [SSOSAML11_SENDARTIFACTTOCONSUMERURL_RSP]
User Agent (browser)	9. The browser makes a request to the URL for the SAML credential collector at the consumer site.
CA SiteMinder® as the Consumer	10. The SAML credential collector makes an isProtected call to the Policy Server for information about the producer. **Log message:** IsProtected call to policy server for producer configuration **Checkpoint code:** SSOSAML11_ISPROTECTEDCALLTOGETPRODUCERCONF_REQ

Actor	Transaction Process
	11. The Policy Server returns the producer configuration information.
	12. The SAML credential collector uses the producer configuration to make a SAML request to the assertion retrieval service at the producer. **Log message:** Reading producer configuration from property. **Checkpoint code:** SSOSAML11_GETPRODUCERCONFFROMPROPERTY_REQ **Log message:** Backchannel call to resolve the artifact **Checkpoint code:** [SSOSAML11_RESOLVEARTIFACT_REQ]
CA SiteMinder® as the Producer	13. The assertion retrieval service at the producer retrieves the SAML assertion from the session store. The service responds with a SAML response that contains the SAML assertion. The assertion is sent to the consumer. **Log message:** Retrieving assertion from session store **Checkpoint code:** [SSOSAML11_RETREIVEASSERTIIONFROMSSTORE_REQ] **Log message:** Received the assertion from session store. **Checkpoint code:** [SSOSAML11_RECEIVEDASSERTIONFROMSSTORE_RSP] **Log message:** Sending assertion as artifact response. **Checkpoint code:** SSOSAML11_SENDARTIFACTRESPONSE_RSP
CA SiteMinder® as the Consumer	14. The SAML credential collector makes a login call to the Policy Server, passing the SAML assertion as credentials. **Log message:** Obtained the SAML11 assertion as response from artifact resolve call **Checkpoint code:** [SSOSAML11_GOTARTIFACTRESPONSE_RSP] **Log message:** Passing response message through login call **Checkpoint code:** [SSO_RESPONSEMESSAGEINLOGIN_REQ]
	15. The consumer validates the assertion. The user is looked up in the user record. The Policy Server returns a success reply. **Log message:** Login successful **Checkpoint code:** [SSO_LOGINSUCEESS_RSP] If the SAML assertion is not valid or a user record cannot be located, a failure reply is returned. **Log message:** Login failure. **Checkpoint code:** [SSO_LOGINFAILURE_RSP]

Actor	Transaction Process
CA SiteMinder® as the Consumer (continued)	16. If the scheme returns a success reply, the SAML credential collector issues a SMSESSION cookie for the consumer domain to the browser. The SAML credential collector also issues a 302 redirect to the target URL. **Log message:** Creating the smsession cookie for SP domain. **Checkpoint code:** [SSO_SMSESSIONFORSPDOMAIN_REQ] **Log message:** Placing smsession in browser. **Checkpoint code:** [SSO_PLACESMSSESSIONTOBROWSER_REQ] If the scheme returns a failure reply, the SAML credential collector issues a 302 redirect to a no access URL.
User Agent (Browser)	17. The browser makes a request to the target URL at the consumer, which the Web Agent protects.

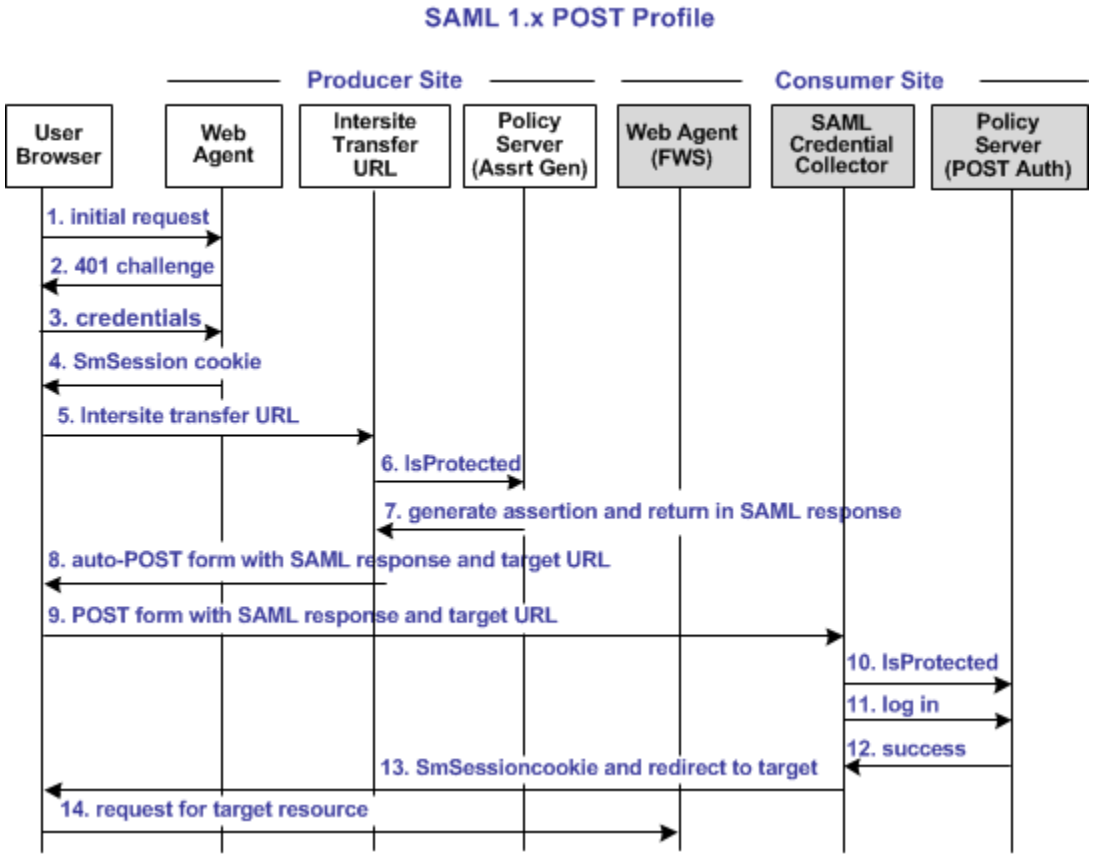
SAML 1.x POST SSO Transaction Flow (Producer-initiated)

The following illustration shows the flow between a user and the federation components at the producer and consumer sites. The flow shows single sign-on between the sites using the SAML 1.x artifact profile as the method for processing the SAML assertion.

The flow diagram assumes the following information:

- The Producer initiates the transaction.
- Authentication and authorization is successful at each site.
- CA SiteMinder® is shown as the producer and consumer only so you can see the processes at each partner. If CA SiteMinder® is the Producer in your environment, review the producer activities in the table. If CA SiteMinder® is the consumer, review the consumer activities in the table.

The process flow diagram for SAML 1.x POST profile follows.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
User Agent (browser)	1. The user makes an initial request to a protected page at the producer site.
CA SiteMinder® as the Producer	2. The Web Agent at the producer site responds with a 401 challenge to the user for credentials. **Log message:** SMSESSION cookie does not exist, redirecting to Authentication URL. **Checkpoint code:** [REDIRECT_AUTH_URL]
	3. The user submits credentials, such as the user name and password to the Web Agent.

Actor	Transaction Process
	4. The Web Agent issues a SMSESSION cookie to the browser for the producer site domain, and allows access to the local page.
	5. The user clicks a link on the local page to visit the consumer site. This link is the intersite transfer URL, which transfers the user to another site. The intersite transfer URL makes a request to the Web Agent at the producer. This URL contains query parameters for the name of the consumer, the location of the SAML credential collector, and the target URL at the consumer site. **Log messages:** SAML11 Consumer Configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML11_CONSUMERCONFFROMPS_REQ]
	6. The Intersite Transfer Service makes an IsProtected call to the Policy Server for the resource. The URL contains the name query parameter that uniquely identifies the consumer. **Log messages:** Request to policy server for generating saml11 assertion/artifact based on selected profile. **Checkpoint code:** [SSOSAML11_GENERATEASSERTIONORARTIFACT_REQ]
	7. The Policy Server generates the assertion and returns it in a digitally signed SAML response. The Policy Server then returns the response to the intersite transfer URL. **Log message:** Policy server generates the saml11 assertion. **Checkpoint code:** [SSOSAML11_PSGENERATEASSERTION_RSP]
	8. The intersite transfer URL service generates an auto-POST form containing the encoded SAML response and the target URL as form variables. The service sends the form to the browser. **Log messages:** Adding response in form for HTTP post. **Checkpoint code:** [FWSBASE_POSTDATAFORM_ADD]
User Agent (browser)	9. The browser posts the HTML form to the SAML Credential Collector at the consumer site. This URL is read from the SAML response that the intersite transfer URL service sends.
CA SiteMinder® as the Consumer	10. The SAML credential collector makes an isProtected call to the Policy Server. **Log message:** IsProtected call to policy server for producer configuration. **Checkpoint code:** SSOSAML11_ISPROTECTEDCALLTOGETPRODUCERCONF_REQ
	11. The SAML credential collector makes a login call to the Policy Server for the requested target resource, passing the assertion as credentials. **Log message:** Reading the configuration to get the target url. **Checkpoint code:** [SSOSAML11_READTARGETURL_REQ]

Actor	Transaction Process
	12. If the login succeeds, the SAML Credential Collector generates an SMSESSION cookie for the consumer site domain. **Log message:** Login successful **Checkpoint code:** [SSO_LOGINSUCEESS_RSP] **Log message:** Creating the smsession cookie for SP domain. **Checkpoint code:** [SSO_SMSESSIONFORSPDOMAIN_REQ]
	13. The SMSESSION cookie is placed in the browser and redirects the user to the target resource. **Log message:** Placing smsession in browser. **Checkpoint code:** [SSO_PLACESMSSESSIONTOBROWSER_REQ]
	14. The browser requests the target resource, which is protected by the consumer-side Web Agent. The browser has an SMSESSION cookie for the consumer domain so the Web Agent does not challenge the user.

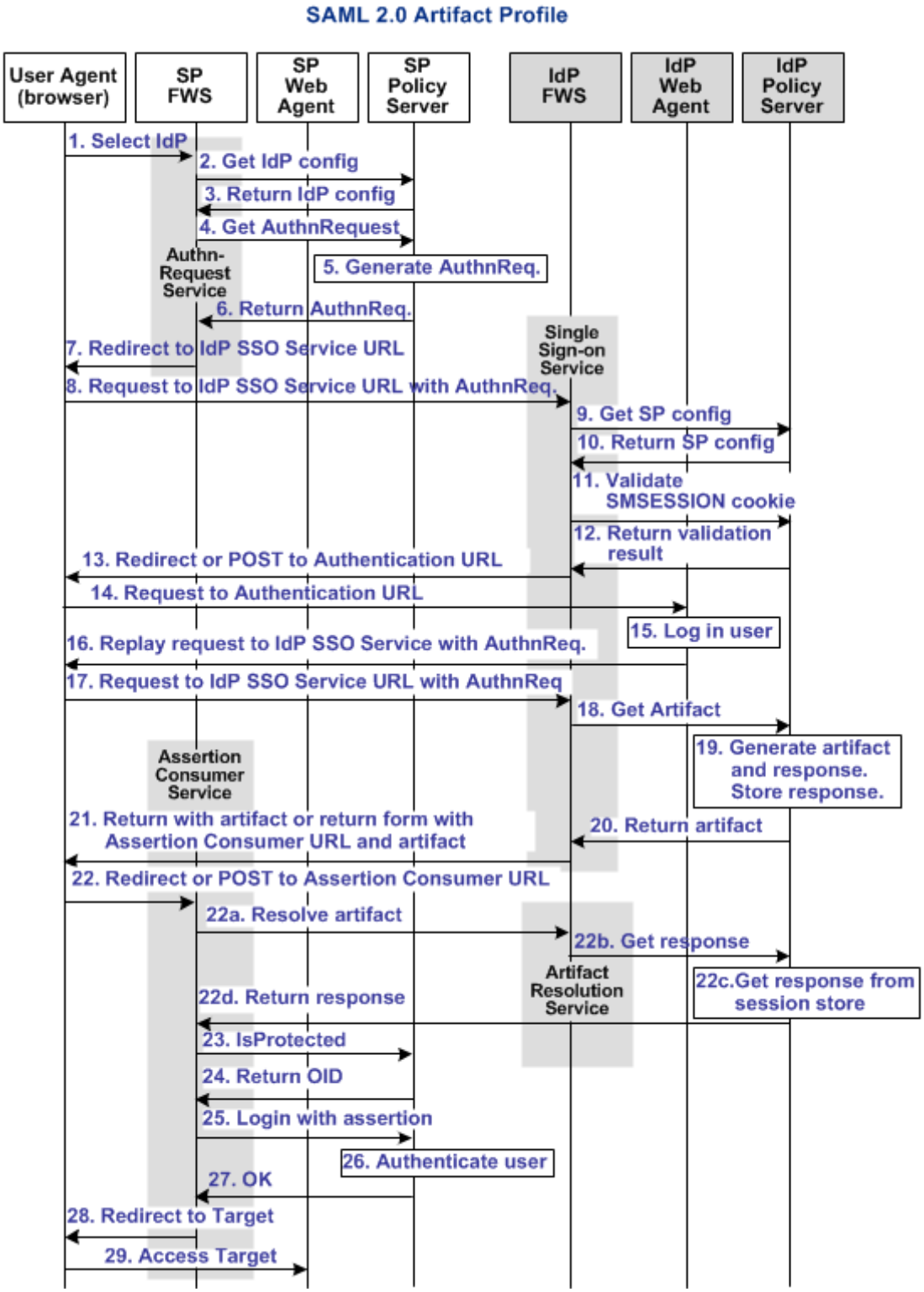
SAML 2.0 Artifact SSO Transaction Flow (SP-initiated)

The following illustration shows the detailed flow between a user and the components at the Identity Provider and Service Provider. The flow shows single sign-on between the sites using SAML 2.0 artifact profile as the method for processing the SAML assertion.

The flow diagram assumes the following information:

- The SP initiates the request for a resource.
- Authentication and authorization at the IdP and SP sites is successful.
- CA SiteMinder® is shown as the IdP and SP so you can see the processes at each partner. If CA SiteMinder® is the SP in your environment, review the SP activities in the table. If CA SiteMinder® is the IdP, review the IdP activities in the table.

The following diagram shows the SAML 2.0 artifact SSO transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the SP	1. The user chooses a link at the SP to authenticate at a specific IdP. This link must include a Provider ID representing the chosen IdP
	2. SP FWS requests the IdP configuration information from the local Policy Server. **Log message:** SAML2.0 IDP Configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML2_IDPCONFFROMPS_REQ]
	3. The local Policy Server returns the IdP configuration to the SP FWS application. FWS caches this information. **Log message:** Policy server returns SAML2.0 IDP Configuration **Checkpoint code:** [SSOSAML2_IDPCONFFROMPS_RSP]
	4. SP FWS requests an AuthnRequest message from the local Policy Server through a tunnel call, passing the Provider ID. The request contains the artifact profile in the ProtocolBinding element value. **Log message**: Get authentication request from policy server **Checkpoint code**: [SSOSAML2_GETAUTHENTICATIONREQFROMPS_REQ]
	5. The SP Policy Server generates the AuthnRequest message and returns it to the SP FWS application.
	6. The local Policy Server returns the AuthnRequest message to the SP FWS in an HTTP redirect binding. **Log message**: Policy server returns authentication request. **Checkpoint code**: [SSOSAML2_GETAUTHENTICATIONREQFROMPS_RSP]
	7. The SP FWS application redirects the user to the IdP Single Sign-on Service URL, which is obtained from the configuration information with the AuthnRequest message. **Log message**: Service redirecting to SSO URL **Checkpoint code**: [SSOSAML2_SSOURL_REDIRECT]
User Agent (browser)	8. The browser requests the IdP Single Sign-on Service URL.
CA SiteMinder® as the IdP	9. IdP FWS requests the SP configuration information from the local IdP Policy Server. **Log message:** SAML2.0 SP configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML2_SPCONFFROMPS_REQ]

Actor	Transaction Process
	10. The local Policy Server returns the configuration, which the FWS application caches. **Log message:** Policy server returns SAML2.0 SP Configuration **Checkpoint code**: [SSOSAML2_SPCONFFROMPS_RSP]
	11. The IdP FWS application gets an SMSESSION cookie for this IdP domain. FWS then calls the Policy Server to validate it. If there is no SMSESSION cookie, the FWS application redirects or posts to the Authentication URL. **Log message**: Session cookie does not exists. Redirecting to authentication URL **Checkpoint code:** [SSOSAML2_AUTHENTICATIONURL_REDIRECT]
	12. The Policy Server validates the SMSESSION cookie and returns the result. **Log message:** Request to validate the session. **Checkpoint message:** [SSOSAML2_SESSIONCOOKIEVALIDATE_REQ]
	13. If the SMSESSION cookie is valid, the IDP FWS requests a SAML 2.0 artifact from the local Policy Server (see step 18). If the SMSESSION cookie is not valid, does not exist or is not valid, the IDP FWS redirects or posts to the Authentication URL. **Log message:** Session cookie does not exists, redirecting to authentication url. **Checkpoint code:** [SSOSAML2_AUTHENTICATIONURL_REDIRECT]
User Agent (browser)	14. If the SMSESSION cookie is not valid, the browser requests the Authentication URL, which the IdP Web Agent protects.
CA SiteMinder® as the IdP	15. The IdP Web Agent logs the user in, sets the SMSESSION cookie, and lets the request pass to the Authentication URL. **Log message:** Service redirecting to SSO URL **Checkpoint code:** [SSOSAML2_SSOURL_REDIRECT]
	16. The Authentication URL is the redirect.jsp file, which replays the request to the IdP single sign-on service with the AuthnRequest message
User Agent (browser)	17. The browser requests the IdP single sign-on service URL. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.
CA SiteMinder® as the IdP	18. The IdP FWS requests a SAML 2.0 artifact from the local Policy Server. FWS passes the AuthnRequest through an authorization call to the realm obtained from the configuration information. **Log message:** Request to policy server for generating saml2 assertion/artifact based on selected profile. **Checkpoint code:** [SSOSAML2_GENERATEASSERTIONORARTIFACT_REQ]

Actor	Transaction Process
	19. The Policy Server generates the artifact and the corresponding response message. The message is formed from the Service Provider configuration. The Policy Server stores the response in the session store. The message is stored as a session variable, and is named using the string representation of the artifact message handle. **Log message:** Policy server generates the artifact for the assertion. **Checkpoint code:** [SSOSAML2_PSGENERATEARTIFACT_REQ] **Log message:** Policy server stores the assertion in session store. **Checkpoint code:** [SSOSAML2_PSSTOREASSERTIONINSSTORE_REQ]
	20. The Policy Server returns the artifact to IdP FWS. **Log message:** Policy server returning the wrappedassertion/artifact based on profile selected in response message. **Checkpoint code:** [SSO_PSWRAPPEDASSERTION_RSP]
	21. The Policy Server returns the SP configuration information. **Log message:** Policy server returns SAML2.0 SP Configuration. **Checkpoint code:** [SSOSAML2_SPCONFFROMPS_RSP] Based on the information, the IdP FWS takes one of the following actions: ■ Redirects the browser to the Assertion Consumer URL at the SP. The URL-encoded artifact is a URL parameter. **Log message**: Sending artifact to assertion consumer as url parameter. **Checkpoint code**: [SSOSAML2_SENDINGARTIFACTASURLPARAM_RSP] ■ Returns a form to the user. The form contains the response message, the Assertion Consumer URL, and the JavaScript to auto-POST the form in the browser. **Log messages:** Adding response in form for HTTP post. **Checkpoint code:** [FWSBASE_POSTDATAFORM_ADD] **Note**: The assertion generator can indicate that the authentication level for the current session is too low. If the level is too low, the IdP FWS redirects to the authentication URL to facilitate step-up authentication.
User Agent (browser)	22. The browser posts the response message to the Assertion Consumer URL at the SP.
CA SiteMinder® as the IdP	23. If the artifact is sent as part of a URL, the browser redirects the user to the Assertion Consumer URL with the artifact. If the artifact is returned in a form, then the browser POSTs the artifact to the Assertion Consumer URL. **Log messages**: Browser posting the response to assertion consumer url. **Checkpoint code**: [SSOSAML2_POSTASSERTIONTOCONSUMERURL_RSP] The SP FWS Assertion Consumer service makes a back channel call to the IdP FWS Artifact Resolution Service to obtain the artifact. Steps 23a-23d reflect the back-channel call.

Actor	Transaction Process
	23a. The SP FWS obtains the artifact from the GET or POST data, depending on how the IdP FWS is configured to redirect the browser. FWS then obtains the SOAP endpoint of the Artifact Resolution Service from the IdP configuration. The source ID is part of the artifact. After the SOAP endpoint is obtained, the SP FWS makes a back-channel call to the IdP FWS Artifact Resolution service to resolve the artifact into a response message. **Log message:** Backchannel call to resolve the artifact. **Checkpoint code:** [SSOSAML2_RESOLVEARTIFACT_REQ] **Log message:** Obtained response message from post data for artifact binding. **Checkpoint code:** SSOSAML2_READRESPONSEARTIFACTDATA_RSP
	23b. The IdP FWS requests the response message from the local Policy Server. The message that is stored as a session variable is requested using the Java Agent API. The session ID is extracted from the artifact. The session variable name is the string representation of the artifact message handle. **Log message:** Extracting session id from artifact. **Checkpoint code:** [SSOSAML2_EXTRACTSESSIONIDFROMARTIFACT_REQ]
	23c. The local Policy Server retrieves the assertion response message from the session store. The Policy Server deletes it after the artifact retrieval. **Log message:** Retrieving assertion from session store. **Checkpoint code:** [SSOSAML2_RETREIVEASSERTIIONFROMSSTORE_REQ]
	23d. The local Policy Server obtains the assertion and returns the artifact response to the IdP FWS. The IdP FWS returns the artifact response to the SP FWS Assertion Consumer Service. **Log message:** Obtained the SAML2 asserion as response from artifact resolve call. **Checkpoint code:** [SSOSAML2_GOTARTIFACTRESPONSE_RSP] **Log message:** Sending assertion as artifact response. **Checkpoint code:** [SSOSAML2_SENDARTIFACTRESPONSE_RSP] The back-channel call is now complete.
CA SiteMinder® as the SP	24. The SP FWS obtains the response message from the post data. The service then determines the target resource from the configuration and makes an isProtected call to the Policy Server for the target resource. **Log message:** Reading the configuration to get the target URL. **Checkpoint code:** [SSOSAML2_READTARGETURL_REQ] **Log message:** IsProtected call to policy server for target resource realm **Checkpoint code:** [SSOSAML2_ISPROTECTEDCALLTOPS_REQ] If the assertion is encrypted, the FWS makes a tunnel call. This call takes the encrypted assertion and returns the assertion in the clear. **Log message:** Tunnel call to decrypt the assertion. **Checkpoint code:** [SSOSAML2_DECRYPTASSERTION_REQ]

Actor	Transaction Process
	25. The Policy Server returns the realm OID for the target resource. **Log message:** Policy server returns the realm OID for target resource. **Checkpoint code:** [SSOSAML2_REALMOIDFORTARGETFROMPS_RSP]
	26. The SP FWS passes the response message to the local Policy Server through a login call. The response message acts as the credentials and the realm OID is obtained from the isProtected call. **Log message:** Passing response message through login call. **Checkpoint code:** [SSO_RESPONSEMESSAGEINLOGIN_REQ] The SAML 2.0 authentication scheme logs the user in using the response message as credentials. **Log message:** Policy server logs in the user using SAML 2 auth scheme. **Checkpoint code:** [SAML2_AUTH_COMPLETE]
	27. The local Policy Server returns OK to the SP FWS.
	28. If a success reply is returned, SP FWS creates an SMSESSION cookie for the SP domain. The service places the cookie in the browser. **Log message:** Login successful **Checkpoint code:** [SSO_LOGINSUCEESS_RSP] **Log message:** Creating the smsession cookie for SP domain **Checkpoint code:** [SSO_SMSESSIONFORSPDOMAIN_REQ] The browser redirects the user to the target URL, which is obtained from the configuration information. **Log message:** Redirecting user to target url. **Checkpoint code**: [SSOSAML2_REDIRECTUSERTARGETURL_REQ] If the login fails, the SP FWS redirects the user to a No Access URL. **Log message:** Login failure **Checkpoint code:** [SSO_LOGINFAILURE_RSP]
User Agent (browser)	29. The browser sends a request to the target URL, which is protected by the SP-side Web Agent. The browser has an SMSESSION cookie so the Web Agent does not challenge the user. The user gains access to the resource.

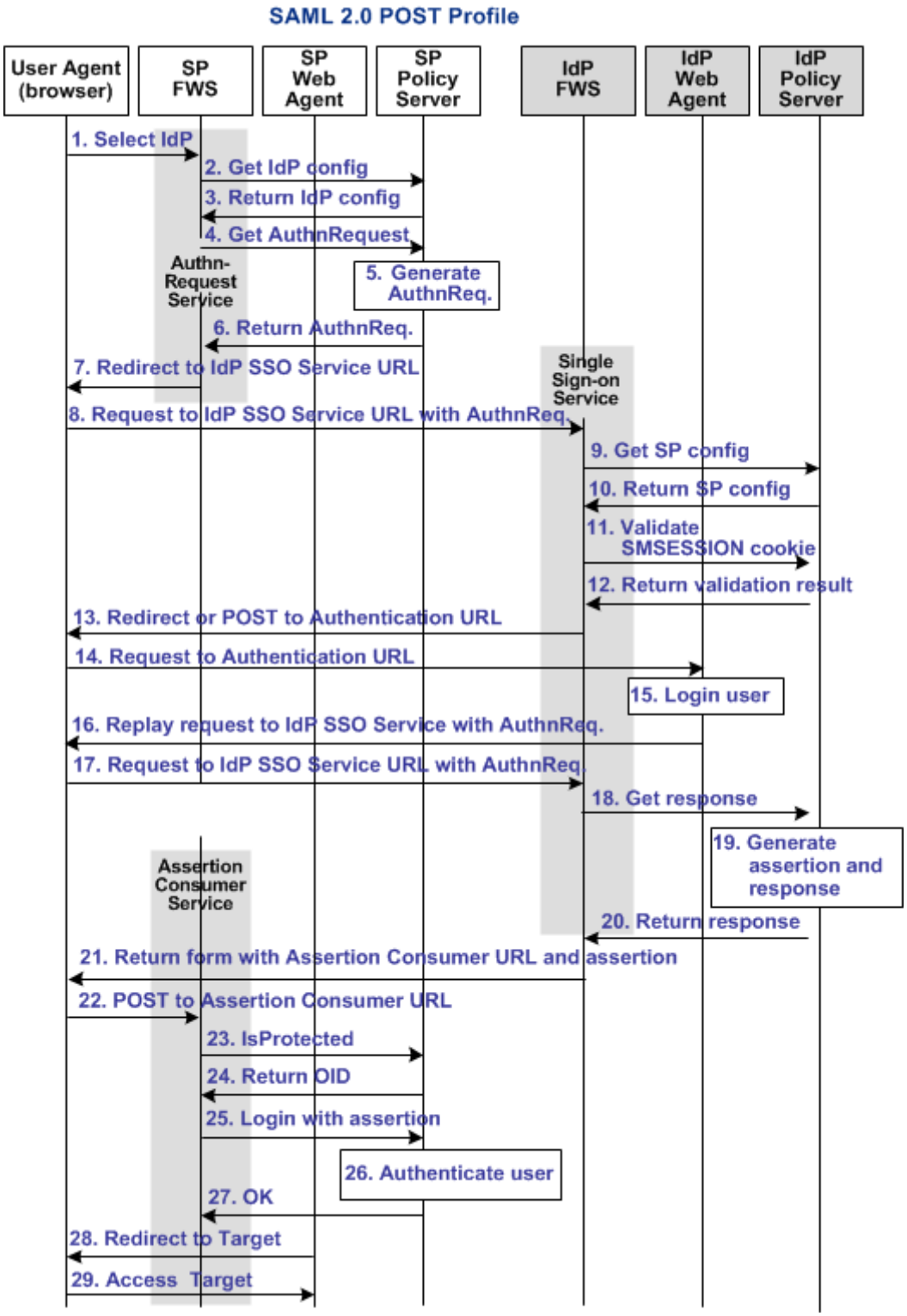
SAML 2.0 POST SSO Transaction Flow (SP-initiated)

The following diagram shows the detailed flow between a user and the components that are deployed at a CA SiteMinder® Identity Provider (IdP) and Service Provider (SP) sites. The flow shows single sign-on between the sites using SAML 2.0 POST profile as the method for processing the SAML assertion.

The flow diagram assumes the following information:

- An SP-initiated request for an assertion.
- Authentication and authorization at the IdP and SP sites is successful.
- CA SiteMinder® is shown as the IdP and SP only so you can see the processes at each partner. If CA SiteMinder® is the SP in your environment, review the SP activities in the table. If CA SiteMinder® is the IDP, review the IdP activities in the table.

The following diagram shows the SAML 2.0 POST SSO transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the SP	1. The user chooses a link at the SP to authenticate at a specific IdP. This link must include a Provider ID representing the chosen IdP.
	2. SP FWS requests the IdP configuration from the local Policy Server. **Log message:** SAML2.0 IDP Configuration is not in cache. Requesting to get from policy server **Checkpoint code:** [SSOSAML2_IDPCONFFROMPS_REQ]
	3. The Policy Server returns the IdP configuration to the SP FWS application. The FWS application caches this information. **Log message:** Policy server returns SAML2.0 IDP Configuration **Checkpoint code:** [SSOSAML2_IDPCONFFROMPS_RSP]
	4. The SP FWS application requests an AuthnRequest message from the local SP Policy Server through a tunnel call, passing the Provider ID. **Log message**: Get authentication request from policy server **Checkpoint code**: [SSOSAML2_GETAUTHENTICATIONREQFROMPS_REQ]
	5. The SP Policy Server generates the AuthnRequest message and returns it to the SP FWS application.
	6. The SP FWS application gets the AuthnRequest response in an HTTP redirect binding. **Log message**: Policy server returns authentication request. **Checkpoint code**: [SSOSAML2_GETAUTHENTICATIONREQFROMPS_RSP]
	7. The SP FWS application redirects the user to the IdP Single Sign-on Service URL. **Log message**: Service redirecting to SSO URL **Checkpoint code**: [SSOSAML2_SSOURL_REDIRECT]
User Agent (browser)	8. The browser requests the IdP Single Sign-on Service URL.
CA SiteMinder® as the IdP	9. IdP FWS requests the SP configuration from the local IdP Policy Server. **Log message:** SAML2.0 SP configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML2_SPCONFFROMPS_REQ]

Actor	Transaction Process
	10. The local Policy Server returns the configuration, which the FWS application caches. **Log message:** Policy server returns SAML2.0 SP Configuration **Checkpoint code**: [SSOSAML2_SPCONFFROMPS_RSP]
	11. The IdP FWS application gets an SMSESSION cookie for this IdP domain. The FWS application then calls the Policy Server to validate it. If there is no SMSESSION cookie, it redirects or posts to the Authentication URL. **Log message**: Session cookie does not exists. Redirecting to authentication URL **Checkpoint code:** [SSOSAML2_AUTHENTICATIONURL_REDIRECT]
	12. The Policy Server validates the SMSESSION cookie and returns the result. **Log message:** Request to validate the session. **Checkpoint message:** [SSOSAML2_SESSIONCOOKIEVALIDATE_REQ]
	13. If the SMSESSION cookie is valid, the IdP FWS skips to step 18. If the SMSESSION cookie is not valid, does not exist or is not valid, the IdP FWS redirects or posts to the Authentication URL. **Log message:** Session cookie does not exists, redirecting to authentication url. **Checkpoint code:** [SSOSAML2_AUTHENTICATIONURL_REDIRECT]
User Agent (browser)	14. If the SMSESSION cookie is not valid, the browser requests the Authentication URL, which the IdP Web Agent protects.
CA SiteMinder® as the IdP	15. The IdP Web Agent logs the user in, sets the SMSESSION cookie, and lets the request pass to the Authentication URL. **Log message:** Service redirecting to SSO URL **Checkpoint code:** [SSOSAML2_SSOURL_REDIRECT]
	16. The Authentication URL is the redirect.jsp file, which replays the request to the IdP single sign-on service with the AuthnRequest message.
User Agent (browser)	17. The browser requests the IdP single sign-on service URL. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.
CA SiteMinder® as the IdP	18. The IdP FWS requests a SAML 2.0 assertion from the Policy Server. The AuthnRequest goes through an authorize call to the realm obtained from the configuration. **Log message:** Request to policy server for generating saml2 assertion/artifact based on selected profile. **Checkpoint code:** [SSOSAML2_GENERATEASSERTIONORARTIFACT_REQ]

Actor	Transaction Process
	19. The Policy Server generates an assertion that is based on the configuration for the SP, signs it, and returns the assertion wrapped in a response message. **Log message:** Policy server generates the saml2 assertion. **Checkpoint code:** [SSOSAML2_PSGENERATEASSERTION_RSP]
	20. The response message is returned to IdP FWS. **Log message:** Policy server returns the wrappedassertion/artifact(based on profile selected) in response message. **Checkpoint code**: [SSO_PSWRAPPEDASSERTION_RSP]
	21. The IdP FWS returns a form to the user. The form contains the response message, the Assertion Consumer URL, and the JavaScript to submit the form. **Log messages:** Adding response in form for HTTP post. **Checkpoint code:** [FWSBASE_POSTDATAFORM_ADD] **Note:** If the Policy Server indicates that the current session authentication level is too low, the IdP FWS redirects to the authentication URL as in Step 13 to facilitate step-up authentication.
User Agent (browser)	22. The browser posts the response to the Assertion Consumer URL at the SP.
CA SiteMinder® as the SP	23. The SP FWS obtains the response message from the POST data. FWS then determines the target resource from the configuration and makes an isProtected call to the Policy Server for the target resource. If the assertion is encrypted, the FWS makes a tunnel call. The call takes the encrypted assertion and returns the assertion in the clear. **Log messages:** Reading the configuration to get the target url. **Checkpoint code:** [SSOSAML2_READTARGETURL_REQ] **Log message:** Get realm oid for target resource from property **Checkpoint code:** [SSOSAML2_REALMOIDFORTARGETFROMPROPERTY_RSP] **Log message:** Tunnel call to decrypt the assertion. **Checkpoint code:** [SSOSAML2_DECRYPTASSERTION_REQ]
	24. The Policy Server returns the realm OID for the target resource. **Log message:** Policy server returns the realm oid for target resource. **Checkpoint code**: [SSOSAML2_REALMOIDFORTARGETFROMPS_RSP]
	25. The SP FWS passes the response message to the local Policy Server through a login call. FWS uses the response message as credentials and the realm OID obtained from the isProtected call. **Log message**: Passing response message through login call. **Checkpoint code**: [SSO_RESPONSEMESSAGEINLOGIN_REQ]

Actor	Transaction Process
	26. The Policy Server logs the user in using the response message as credentials. **Log message:** Policy server logs in the user using SAML 2 auth scheme. **Checkpoint code:** [SAML2_AUTH_COMPLETE]
	27. The local Policy Server returns OK to the SP FWS. **Log message:** Login successful **Checkpoint code**: [SSO_LOGINSUCEESS_RSP]
	28. If a success reply is returned, the SP FWS creates an SMSESSION cookie for the SP domain. The FWS application places the cookie in the browser and redirects the user to the target URL. **Log message:** Redirecting user to target url **Checkpoint code:** [SSOSAML2_REDIRECTUSERTARGETURL_REQ] If the login fails, the SP FWS redirects the user to a No Access URL. **Log message:** Login failure **Checkpoint code:** [SSO_LOGINFAILURE_RSP]
User Agent (browser)	29. The browser sends a request to the target URL, which is protected by the SP-side Web Agent. The browser has an SMSESSION cookie so the Web Agent does not challenge the user. The user gains access to the resource.

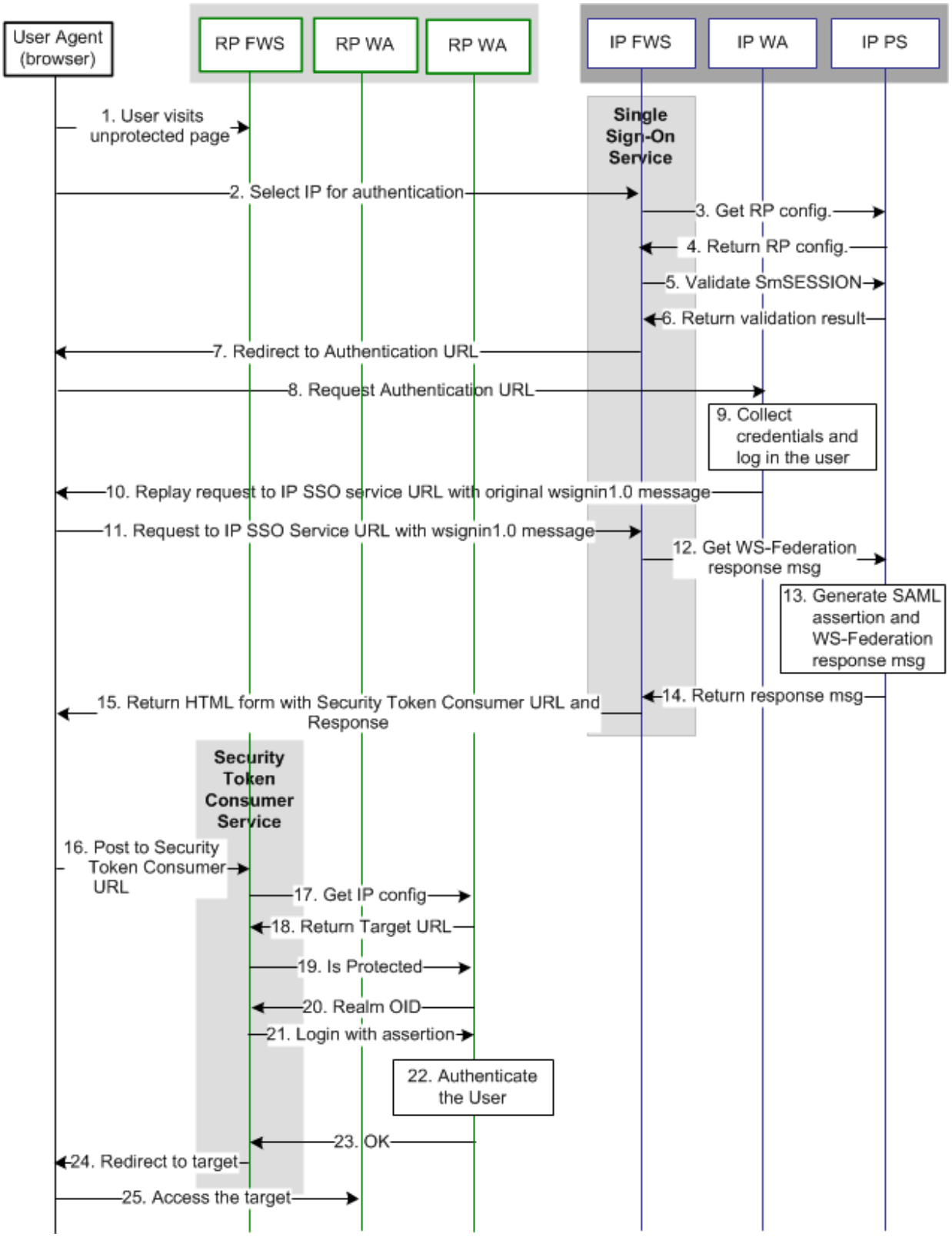
WS-Federation SSO Transaction Flow (RP-initiated)

The following illustration shows the flow between a user and the federation components at the Identity Partner (IP) and Resource Partner (RP) sites. The flow shows single sign-on between the sites using the WS-Federation Passive Requestor profile as the method for processing the SAML assertion.

The flow diagram assumes the following information:

- The Resource Partner initiates the request for a resource.
- Authentication and authorization at each site is successful.
- CA SiteMinder® is shown as the IP and RP only so you can see the processes at each partner. If CA SiteMinder® is the IP in your environment, review the IP activities in the table. If CA SiteMinder® is the RP, review the RP activities in the table.

The following diagram shows the WS-Federation SSO transaction flow.


User Agent (browser)
RP FWS
RP WA
RP WA
IP FWS
IP WA
IP PS
Single Sign-On Service
1. User visits unprotected page
2. Select IP for authentication
3. Get RP config.
4. Return RP config.
5. Validate SmSESSION
6. Return validation result
7. Redirect to Authentication URL
8. Request Authentication URL
9. Collect credentials and log in the user
10. Replay request to IP SSO service URL with original wsignin1.0 message
11. Request to IP SSO Service URL with wsignin1.0 message
12. Get WS-Federation response msg
13. Generate SAML assertion and WS-Federation response msg
14. Return response msg
15. Return HTML form with Security Token Consumer URL and Response
Security Token Consumer Service
16. Post to Security Token Consumer URL
17. Get IP config
18. Return Target URL
19. Is Protected
20. Realm OID
21. Login with assertion
22. Authenticate the User
23. OK
24. Redirect to target
25. Access the target

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the RP	1. The user visits an unprotected site selection page at the Resource Partner.
	2. The user clicks on a link to authenticate at the IP. This link points to the Single Sign-on Service at the IP. The link must contain the RP Provider ID and can include optional parameters, such as the wctx parameter.
	3. The IP FWS requests the RP configuration from the local Policy Server. **Log message:** Trying to fetch Wsfed Resource Partner Configuration from cache **Checkpoint code:** [SSOWSFED_RESOURCEPARTNERCONFFROMCACHE_REQ] **Log message:** Wsfed Resource Partner Configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOWSFED_RESOURCEPARTNERCONFFROMPS_REQ]
	4. The local Policy Server returns the configuration. **Log message**: Policy server returns Wsfed Resource Partner Configuration **Checkpoint code**: [SSOWSFED_RESOURCEPARTNERCONFFROMPS_RSP]
	5. The IP FWS gets the SMSESSION cookie for the IP domain and calls the Policy Server to validate it. If there is no SMSESSION cookie, the IP FWS skips to step 7. **Log message**: Request to validate the session **Checkpoint code**: [SSOWSFED_SESSIONCOOKIEVALIDATE_REQ]
	6. The Policy Server validates the SMSESSION cookie and returns the result to the FWS application.
	7. If the SMSESSION cookie does not exist or is not valid, the IP FWS redirects the user to the Authentication URL obtained from the RP configuration. If the SMSESSION cookie is valid, the IP FWS skips to step 12. **Log message:** Session cookie does not exists, redirecting to authentication url **Checkpoint code:** [SSOWSFED_AUTHENTICATIONURL_REDIRECT]
User Agent (browser)	8. The browser requests the Authentication URL, which the IP Web Agent protects.
CA SiteMinder® as the IP	9. The IP WA authenticates the user and sets the SMSESSION cookie. The IP WA lets the request pass to the Authentication URL.
	10. The Authentication URL replays the request to the IP SSO service with the original wsignin message.

Actor	Transaction Process
User Agent (browser)	11. The browser requests the IP SSO Service URL. This request is equivalent to the request from step 2, but now the user has a valid SMSESSION cookie.
CA SiteMinder® as the IP	12. The IP FWS requests a WS-Federation <RequestSecurityTokenResponse> from the Policy Server through an authorize call to the realm obtained from the configuration. **Log message:** Request to policy server for generating Wsfed assertion. **Checkpoint message:** [SSOWSFED_GENERATEASSERTION_REQ]
	13. The Policy Server generates a SAML1.1 assertion that is based on the RP configuration. **Log message:** Policy server generates the samlxx assertion for wsfed12. **Checkpoint code**: [SSOWSFED12_PSGENERATESAML11ASSERTION_RSP]
	14. The Policy Server signs the assertion and returns it to the IP FWS application in an <RequestSecurityTokenResponse> message. **Log message:** Policy server signs the Assertion element of the RequestSecurityTokenResponse **Checkpoint code:** Differs for different SAML protocols. [SSOWSFED10_PSGENERATELEGACYASSERTION_RSP] [SSOWSFED12_PSGENERATESAML12ASSERTION_RSP] [SSOWSFED_PSSIGNASSERTION_RSP]
	15. The IP FWS returns a form to the user containing the URL encoded <RequestSecurityTokenResponse> message, the Security Token Consumer Service URL, the Optional wctx in the wsignin message, and the JavaScript to auto submit the form. If the original wsignin request contains the wreply parameter, its value becomes the Security Token Consumer URL. The wreply value becomes the URL only if the Security Token Consumer URL setting is not in the RP configuration. The Security Token Consumer URL in the RP configuration takes precedence over the wreply parameter. **Note:** The Policy Server can indicate that the authentication level of the current session is too low. If the level is too low, the IP FWS application redirects the browser to the authentication URL as in step 7 to facilitate step-up authentication. **Log message:** Received the assertion response. **Checkpoint code:** [SSOWSFED_RECEIVEDASSERTION_RSP]
User Agent (browser)	16. The browser posts the <RequestSecurityTokenResponse> message and wctx to the Security Token Consumer URL at the RP.

Actor	Transaction Process
CA SiteMinder® as the RP	17. The RP FWS application obtains the <RequestSecurityTokenResponse> message and wctx from the POST data. The RP FWS application requests the IP configuration from the local Policy Server. **Log message:** Browser posting the response to security token consumer service url. **Checkpoint code:** [SSOWSFED_POSTASSERTIONTOSECURITYTOKENCONSUMER_RSP] **Log message:** Extracting the assertion from security token consumer response **Checkpoint code:** [SSOWSFED_EXTARCTASSERTIONFROMSECURITYTOKENRESPONSE_REQ]
	18. RP FWS determines the target resource from the IP configuration from local Policy Server. If the target resource is not part of the IP configuration, and the wctx parameter is found in the POST data, the wctx value becomes the target resource. **Log message:** Request to get the target url realm. **Checkpoint code:** [SSOWSFED_GETTARGETURLREALM_REQ]
	19. FWS makes an isProtected call to the Policy Server for the target resource.
	20. The Policy Server returns the realm OID for the target resource.
CA SiteMinder® as the RP (continued)	21. The RP FWS application passes the <RequestSecurityTokenResponse> message to the local Policy Server through a login call. The <RequestSecurityTokenResponse> message and the realm OID obtained from the isProtected call service as credentials.
	22. The RP FWS application logs in the user with the <RequestSecurityTokenResponse> message as credentials.
	23. The local Policy Server returns an OK status message to the RP FWS application.
	24. The RP FWS application generates the SMSESSION cookie for the RP domain. FWS puts the cookie in the browser and redirects the user to the Target URL or to the wctx POST data. If the login fails, the FWS application redirects the user to a No Access URL. **Log message:** Redirecting user to target url. **Checkpoint code**: [SSOWSFED_REDIRECTUSERTARGETURL_REQ]
User Agent (browser)	25. The user agent requests the Target URL that the RP-side Web Agent protects. The browser has the SMSESSION cookie for the RP domain, so the Web Agent does not have to challenge the user.

WS-Federation SSO Transaction Flow (IP-initiated)

IP-initiated single sign-on is similar to an RP-initiated transaction. The main difference is the few actions that take place at the IP before the user is sent to the RP.

At the IP, the following actions occur:

1. The user selects a link for a specific partner site. This link is part of the HTML content at the IP, which contains intersite transfer links to different RP sites.
2. The link directs the web browser to the IP SSO Service URL.
3. The SSO service redirects the browser to the RP, where the rest of the processing is same as specified in the RP-initiated SSO transaction flow (see page 89).

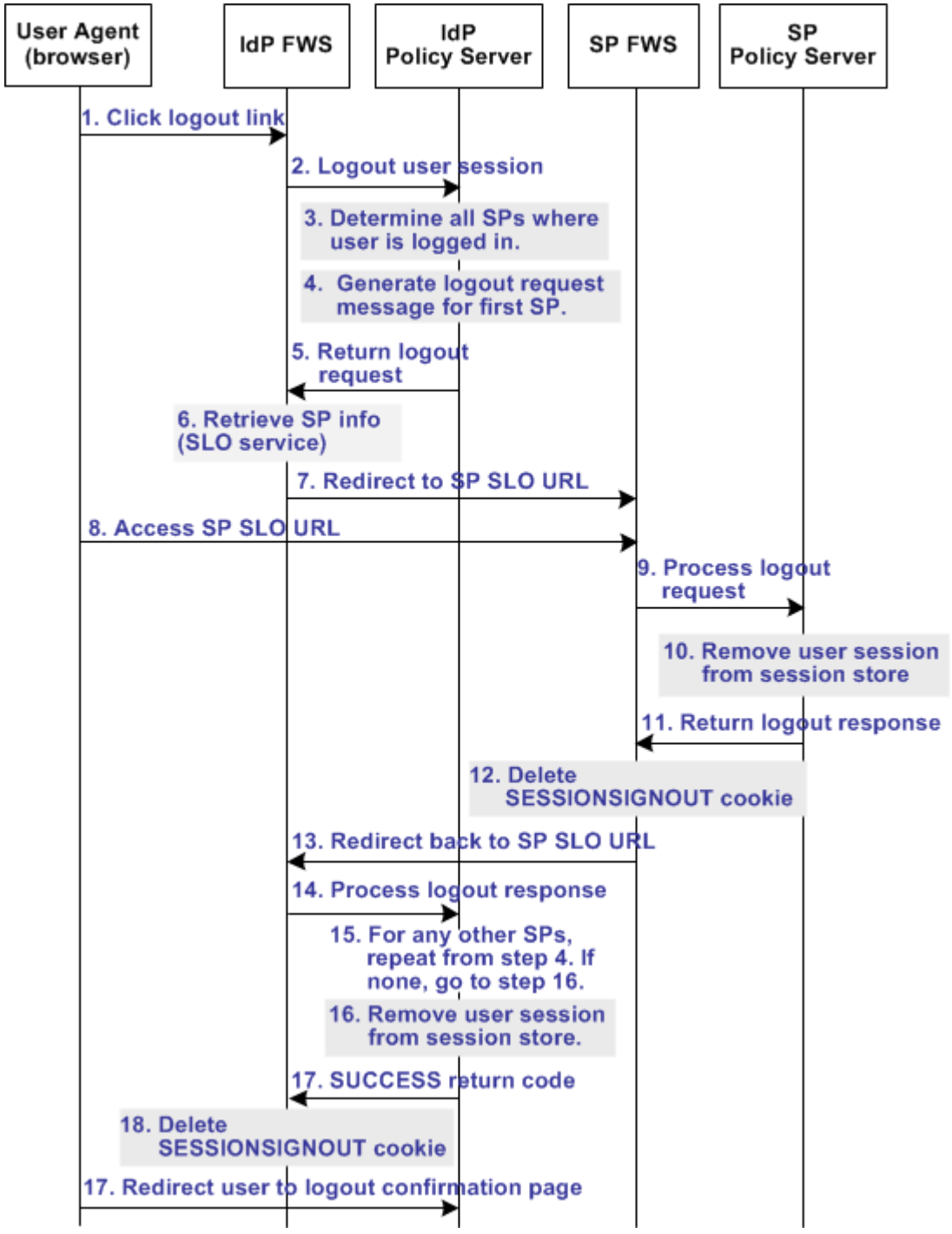
SAML 2.0 Single Logout Transaction Flow (IdP-initiated)

The following illustration shows the detailed flow for a single logout (SLO) request between a user and the components that are deployed at a CA SiteMinder® Identity Provider (IdP) and Service Provider (SP). This flow shows single logout for all entities that have a session with a particular user.

The flow diagram assumes the following information:

- An IdP initiates the log out request.
- The HTTP-Redirect binding is in use.
- CA SiteMinder® is shown as the IdP and SP only so you can see the processes at each partner. If CA SiteMinder® is the SP in your environment, review the SP activities in the table. If CA SiteMinder® is the IDP, review the IdP activities in the table.

The following illustration shows the SLO transaction flow. When the IdP initiates SLO, several SPs can receive the SLO request.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the IdP	1. The user clicks a logout link at the IdP. The browser accesses the Single Logout servlet at the IdP. The IdP FWS application renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the current user session. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	2. The IdP FWS application reads the SessionID value from the SESSIONSIGNOUT cookie and sends a request to the IdP Policy Server to terminate the user session. **Log message:** Fetching session details from cookie. **Checkpoint code:** [SLO_SESSION_FETCH] Depending on the request type (GET or POST), one of the corresponding checkpoint messages are logged: **Log message:** Receiving request at SAML2 SLO Logout URL through GET method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEGET_RECEIVE] or **Log message:** Receiving request at SAML2 SLO Logout URL through POST method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEPOST_RECEIVE]
	3. The IdP Policy Server determines all of the SPs where the user was logged in.
	4. Based on the session store information, the user session status for the first SP in the list is changed to a LogoutInProgress state. The Policy Server generates a LogoutRequest request to invalidate the user session at the SP. **Log message**: Generating SAML LogoutRequest. **Checkpoint code:** [SLO_LOGOUTREQUEST_GEN]
	5. The Policy Server returns a LogoutRequest request to IdP FWS. The Policy Server also returns the Provider ID of the SP and provider type. **Log message**: Generating SAML LogoutRequest. **Checkpoint code:** [SLO_LOGOUTREQUEST_GEN]
	6. The IdP FWS application retrieves the provider configuration data of the SP from the Policy Server. The data includes the SLO service URL at the SP. **Log message**: Fetching provider information. **Checkpoint code**: [SLOSAML2_PROVIDERINFO_FETCH]
	7. The IdP FWS application redirects the user to the SP SLO service with the LogoutRequest message added as a query parameter. **Log message**: Redirecting to service providers single logout service url. **Checkpoint code**: [SLOSAML2_SPSLOSERVICEURL_FORWARD]

Actor	Transaction Process
User Agent (browser)	8. The browser accesses SLO service at the SP.
CA SiteMinder® as the SP	9. The SP FWS application receives and processes the LogoutRequest message. **Log message:** Receiving request at SAML2 SLO Logout URL through GET method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEGET_RECEIVE] or **Log message:** Receiving request at SAML2 SLO Logout URL through POST method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEPOST_RECEIVE] The SP renames the SMSESSION cookie to SESSIONSIGNOUT. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	10. The SP removes the user session from the SP session store. **Log message:** Logging out session cookie. **Checkpoint code**: [SLO_SESSIONCOOKIE_LOGOUT] **Log message:** Terminating user session from session store. **Checkpoint code**: [SLO_USERSESSION_TERMINATE]
	11. The SP Policy Server returns a signed LogoutResponse message to the SP FWS application. This response contains the provider ID of the IdP and the provider type. The Policy Server also informs the application that the user session is no longer in the session store. **Log message:** Generating SAML LogoutResponse. **Checkpoint code**: [SLO_LOGOUTRESPONSE_GEN]
	12. After learning that the user session is removed from the session store, the SP FWS application deletes the SESSIONSIGNOUT cookie. **Log message:** Terminating user session from session store. **Checkpoint code**: [SLO_USERSESSION_TERMINATE]
	13. The SP FWS application redirects the user to the IdP SLO service with the LogoutResponse message added as a query parameter. The browser accesses the SLO service at the IdP. The service processes the signed LogoutResponse message. **Note:** If the LogoutResponse message contains non-SUCCESS return code, the SP issues a SIGNOUTFAILURE cookie. A base 64-encoded Partner ID is appended to the cookie value. If there are multiple IDs in the cookie, a space character separates them. **Log message:** Redirecting to identity provider single logout service url. **Checkpoint code**: [SLOSAML2_IDPSLOSERVICEURL_FORWARD]
CA SiteMinder® as the IdP	14. The IdP Policy Server receives the LogoutResponse message and processes it.

Actor	Transaction Process
	15. The SP Policy Server removes the user session from the session store. **Log message:** Terminating user session from session store. **Checkpoint code:** [SLO_USERSESSION_TERMINATE]
	16. The IdP Policy Server checks for any more SPs. If there are more, the flow repeats, beginning at step 4. Otherwise, the process continues to the next step.
	17. After the session is removed from the session store, the IdP Policy Server sends a SUCCESS return code to the FWS application. The Policy Server includes the SP ID in the final LogoutResponse message.
	18. If there are no more LogoutRequest or LogoutResponse messages to process, the IdP FWS application deletes the SESSIONSIGNOUT cookie.
	19. The browser redirects the user to the Logout Confirmation page at the SP. **Log message:** Redirecting to SLO confirmation URL. **Checkpoint code**: [SLOSAML2_LOGOUTCONFIRMURL_REDIRECT] **Log message:** Displaying local logout message / URL. **Checkpoint code:** [SLO_LOCALLOGOUT_DISPLAY]

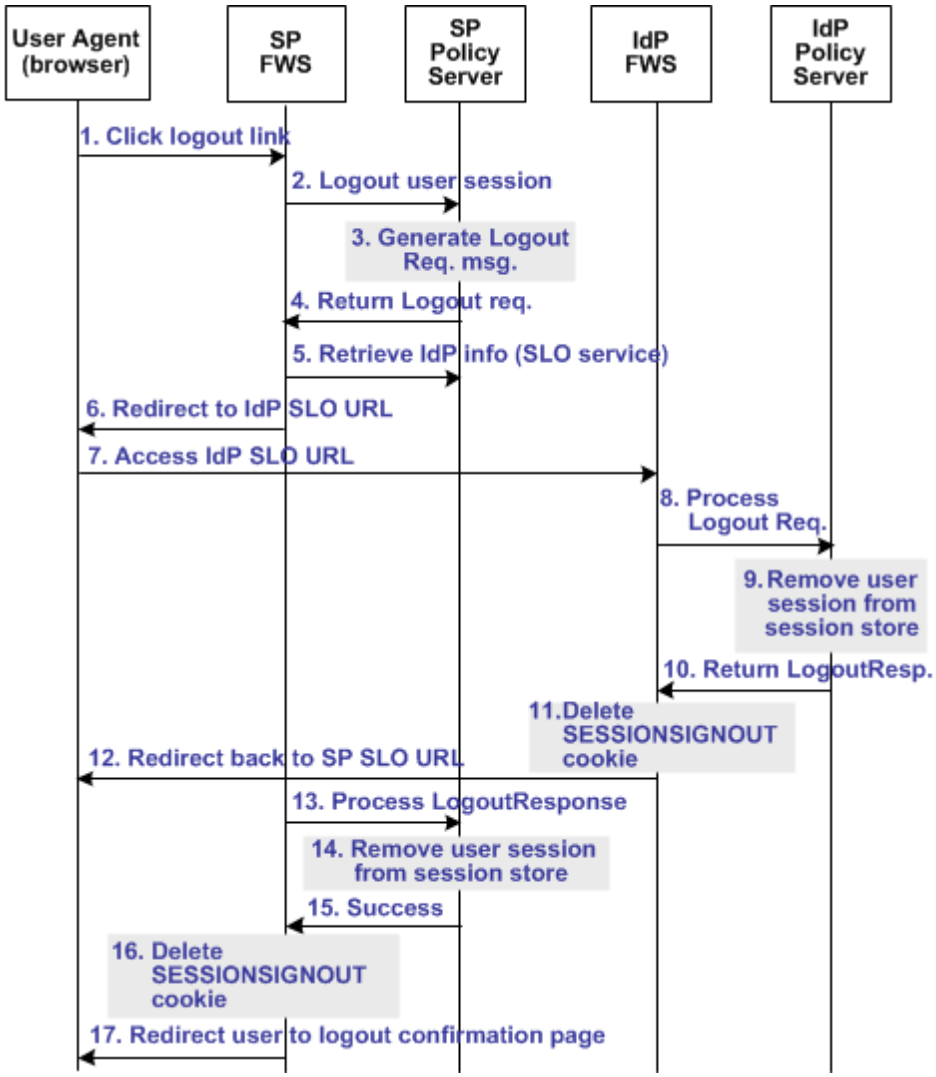
SAML 2.0 Single Logout Transaction Flow (SP-initiated)

The following illustration shows the detailed flow for a single logout (SLO) request between a user and the components deployed at a CA SiteMinder® Identity Provider (IdP) and Service Provider (SP). This flow shows single logout for all entities that have a session with a particular user.

The flow diagram assumes the following information:

- An SP initiates the log out request.
- The HTTP-Redirect binding is in use.
- CA SiteMinder® is shown as the IdP and SP only so you can see the processes at each partner. If CA SiteMinder® is the SP in your environment, review the SP activities in the table. If CA SiteMinder® is the IDP, review the IdP activities in the table.

The following illustration shows the SLO transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

The sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the SP	1. The user clicks a logout link at SP. The browser accesses the Single Logout servlet at the SP. The SP FWS application renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the current user session. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	2. The FWS application reads the SessionId value from the SESSIONSIGNOUT cookie and sends a request to the Policy Server to terminate the user session. **Log message:** Fetching session details from cookie. **Checkpoint code:** [SLO_SESSION_FETCH] Depending on the request type (GET or POST), one of the corresponding checkpoint messages are logged: **Log message:** Receiving request at SAML2 SLO Logout URL through GET method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEGET_RECEIVE] or **Log message:** Receiving request at SAML2 SLO Logout URL through POST method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEPOST_RECEIVE]
	3. Based on the session store information, the user session status is changed to a LogoutInProgress state. The Policy Server determines that the user session is created based on the assertion received from an IdP. The Policy Server generates a LogoutRequest request to invalidate the user session at the IdP. **Log message**: Generating SAML LogoutRequest. **Checkpoint code:** [SLO_LOGOUTREQUEST_GEN] **Log message**: Identifying providers associated with user session for single logout. **Checkpoint code:** [SLO_PROVIDERFORLOGOUT_IDENTIFY]
	4. The Policy Server returns a LogoutRequest request to SP FWS. The Policy Server also returns the Provider ID of the IdP and provider type. **Log message**: Generating SAML LogoutRequest. **Checkpoint code:** [SLO_LOGOUTREQUEST_GEN]
	5. The SP FWS application retrieves the provider configuration data of the IdP from the Policy Server. The data includes the SLO service URL at the IdP. **Log message**: Fetching provider information. **Checkpoint code**: [SLOSAML2_PROVIDERINFO_FETCH]

Actor	Transaction Process
	6. The SP FWS application redirects the user to the SLO service at the IdP with the SAML LogoutRequest message added as a query parameter. **Log message**: Redirecting to identity provider single logout service url. **Checkpoint code**: [SLOSAML2_IDPSLOSERVICEURL_FORWARD]
User Agent (browser)	The browser accesses SLO service at the IdP.
CA SiteMinder® as the IdP	7. The IdP FWS application receives a LogoutRequest message. Depending on the request type (GET or POST), one of the corresponding checkpoint messages are logged: **Log message:** Receiving request at SAML2 SLO Logout URL through GET method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEGET_RECEIVE] or **Log message:** Receiving request at SAML2 SLO Logout URL through POST method. **Checkpoint code:** [SLOSAML2_LOGOUTSERVICEPOST_RECEIVE] The IdP renames the SMSESSION cookie to SESSIONSIGNOUT. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	8. The IdP processes the signed LogoutRequest message. The IdP then tries to invalidate the user session at all SPs specified in the session store for that session. The only SP that is not invalidated is the SP that sent the original LogoutRequest. **Note:** The process for logging the user out at each SP is the same from Step 2 through Step 7. **Log message:** Logging out session cookie. **Checkpoint code**: [SLO_SESSIONCOOKIE_LOGOUT]
	9. After terminating the user session from all relevant SPs, the IdP removes the user session from the session store. **Log message:** Terminating user session from session store. **Checkpoint code**: [SLO_USERSESSION_TERMINATE] 10. The IdP Policy Server returns a signed LogoutResponse message to the IdP FWS application. This response contains the provider ID of the SP and the provider type. The IdP Policy Server also informs the application that the user session is no longer in the session store. **Log message:** Generating SAML LogoutResponse. **Checkpoint code**: [SLO_LOGOUTRESPONSE_GEN]
	11. After learning that the user session is removed from the session store, the IdP FWS application deletes the SESSIONSIGNOUT cookie.

Actor	Transaction Process
CA SiteMinder® as the IdP (continued)	12. The IdP redirects the user to the single logout service at the SP with the LogoutResponse message added as a query parameter. The browser accesses the SLO service at the SP. The service processes the signed LogoutResponse message. **Note:** If the LogoutResponse message contains non-SUCCESS return code, the SP issues a SIGNOUTFAILURE cookie, and a base 64-encoded Partner ID is appended to the cookie value. If there are multiple IDs in the cookie, a space character separates them. **Log message:** Redirecting to service providers single logout service url. **Checkpoint code**: [SLOSAML2_SPSLOSERVICEURL_FORWARD]
CA SiteMinder® as the SP	13. The SP Policy Server receives the LogoutResponse message from the FWS application and processes it.
CA SiteMinder® as the SP (continued)	14. The SP Policy Server removes the user session from the session store. **Log message:** Terminating user session from session store. **Checkpoint code:** [SLO_USERSESSION_TERMINATE]
	15. After the session is removed from the session store, the Policy Server sends a SUCCESS return code to the FWS application. The Policy Server includes the SP ID in the final LogoutResponse message.
	16. If there are no more LogoutRequest or LogoutResponse messages to process, the SP FWS application deletes the SESSIONSIGNOUT cookie.
	17. FWS redirects the user to the Logout Confirmation page at the SP. **Log message:** Redirecting to SLO confirmation URL. **Checkpoint code**: [SLOSAML2_LOGOUTCONFIRMURL_REDIRECT] **Log message:** Displaying local logout message / URL. **Checkpoint code:** [SLO_LOCALLOGOUT_DISPLAY]

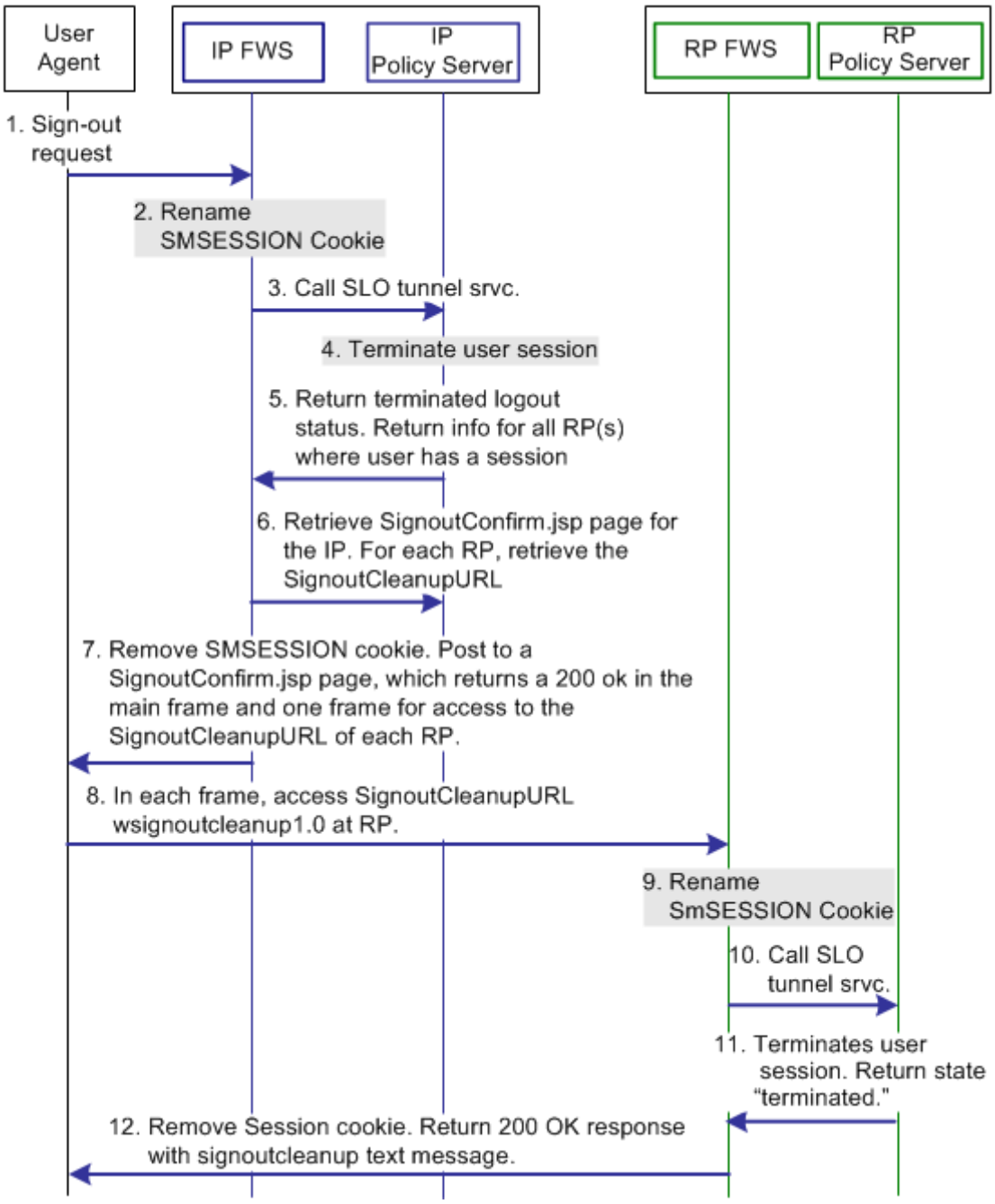
WS-Federation Sign-out Transaction Flow (IP-initiated)

The following illustration shows the flow for a signout request between a user and the components that are deployed at an Identity Provider (IP) and Resource Partner (RP). This flow shows a sign-out transaction for all WS-Federation entities that have a session with a particular user.

The flow diagram assumes the following information:

- The IP initiates the signout transaction.
- CA SiteMinder® is shown as the IP and RP so you can see the processes at each partner. If CA SiteMinder® is the IP in your environment, review the IP activities in the table. If CA SiteMinder® is the RP, review the RP activities in the table.

The following illustration shows the WS-Federation sign-out transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

When sign-out is initiated at the Identity Provider, the sequence of events is as follows:

Actor	Transaction Process
CA SiteMinder® as the IP	1. The user clicks a link at the IP to end the global session. The browser sends an HTTP-based wsignout request to the signout servlet at the IP.
	2. The IP FWS application renames the SMSESSION cookie to SESSIONSIGNOUT to invalidate the current user session. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	3. The IP FWS reads the SessionId value from the SESSIONSIGNOUT cookie and calls the SLO Tunnel Service API to terminate the user session. **Log message:** Fetching session details from cookie. **Checkpoint message:** SLO_SESSION_FETCH **Log message:** Performing tunnel call for WSFED signout. **Checkpoint code:** [SLOWSFED_TUNNEL_REQUEST]
	4. The SLO Tunnel Service API sets the user session status to "Terminated" in the session store. The service also removes all the RP references from the session store that are associated with that user session. **Log message**: Setting session to inactive assuming a cleanup state. **Checkpoint code:** [SLOWSFED_INACTIVESTATE_SET]
	5. The SLO Tunnel Service API returns the logout status "Terminated" to the FWS Signout Servlet. The Tunnel library also returns the RP providerID and providerType for all the RPs associated with the user session. **Log message**: Terminating user session from session store. **Checkpoint code**: [SLO_USERSESSION_TERMINATE]
	6. The IP FWS retrieves the provider configuration data of the RP from the cache of the provider that FWS maintains. The information includes the signout cleanup URL. **Log message**: Validate GET request for necessary parameters. **Checkpoint code**: [SLOWSFED_GETREQUEST_VALIDATE]

Actor	Transaction Process
	7. The IP FWS removes the SESSIONSIGNOUT cookie and posts an IP Signout message and multiple RP-SignoutCleanup locations as POST data to the SignoutConfirmURL JSP. **Log message:** Logging out session cookie. **Checkpoint code:** [SLO_SESSIONCOOKIE_LOGOUT] The SignoutConfirmURL JSP is responsible for parsing various post variables and creating a frame-based HTML page. The main frame of this HTML page displays the IP-SignOut message. Each of the remaining frames accesses the SignoutCleanupURL of individual RPs associated with the user session. **Log message**: Sending signout message to Identity Provider (Account Partner). **Checkpoint code**: [SLOWSFED_IDPSIGNOUTMSG_SEND] **Log message**: Redirecting to signout confirmation URL **Checkpoint code**: [SLOWSFED_LOGOUTCONFIRMURL_REDIRECT]
User Agent (browser)	8. The browser accesses the SignoutCleanup service at the RP.
CA SiteMinder® as the RP	9. When the RP FWS application receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT. FWS then calls the SLO Tunnel Service API to process the wsignoutcleanup request. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME] **Log message:** Receiving signout request at WSEFD through GET method **Checkpoint code:** [SLOWSFED_LOGOUTSERVICEGET_RECEIVE]
	10. The SLO tunnel library processes the wsignoutcleanup request and terminates the user session from the session store. **Log message:** Terminating user session from session store. **Checkpoint code:** [SLO_USERSESSION_TERMINATE]
	11. The SLO tunnel library returns FWS with a "Terminated" status message indicating that the user session no longer exists in the session store. **Log message:** Logging out session cookie. **Checkpoint code:** [SLO_SESSIONCOOKIE_LOGOUT]
	12. The FWS Signout Servlet removes the SESSIONSIGNOUT cookie and returns a 200 OK response in the frame. **Log message:** Displaying local logout message / URL. **Checkpoint message:** [SLO_LOCALLOGOUT_DISPLAY]

Note: Steps 8-12 are repeated for individual RPs simultaneously in different frames of the same HTML page.

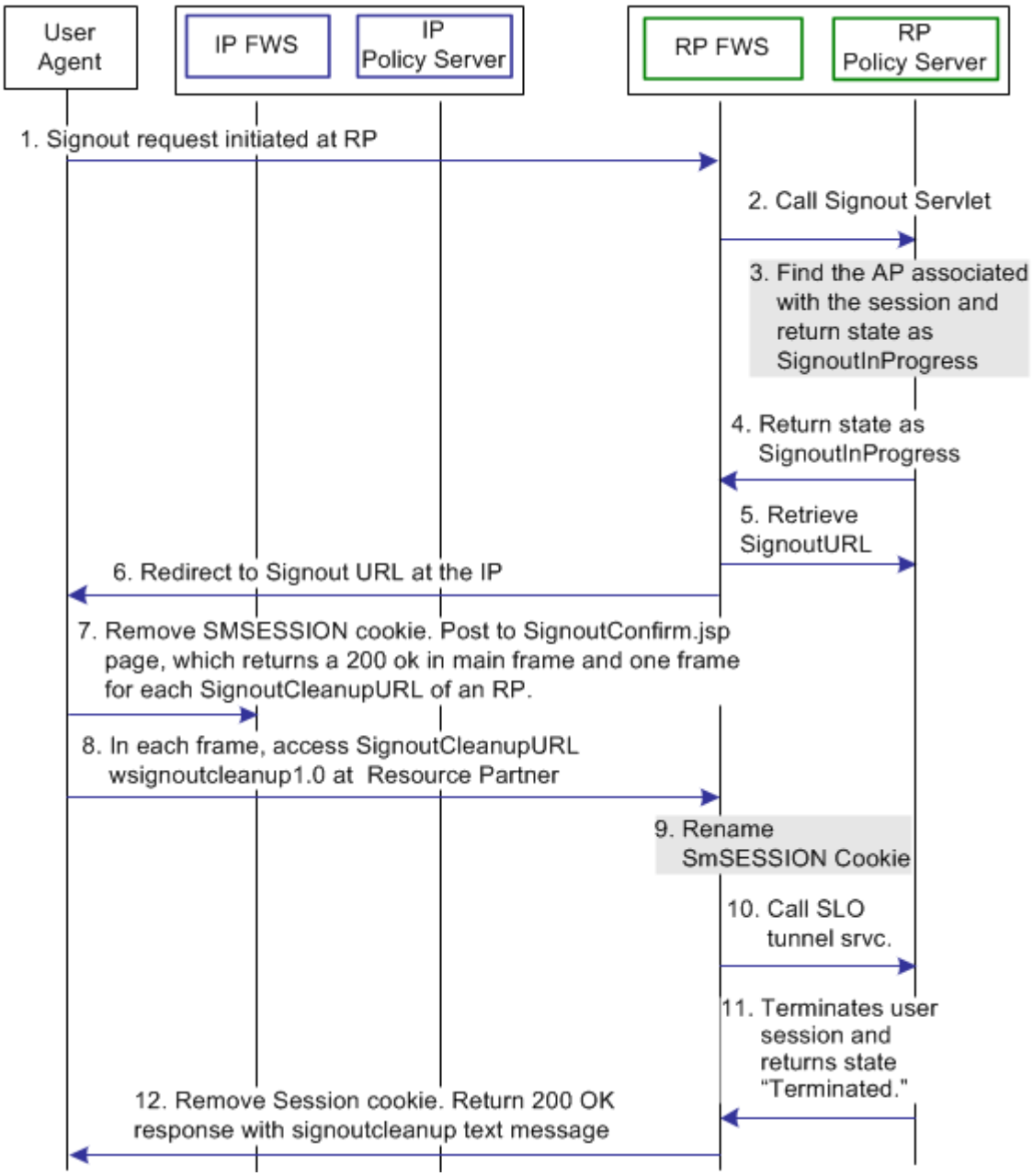
WS-Federation Sign-out Transaction Flow (RP-initiated)

The following illustration shows the flow for a signout request between a user and the components that are deployed at an Identity Provider (IP) and Resource Partner (RP). This flow shows a sign-out transaction for all WS-Federation entities that have a session with a particular user.

The flow diagram assumes the following information:

- The RP initiates the signout transaction.
- CA SiteMinder® is shown as the IP and RP so you can see the processes at each partner. If CA SiteMinder® is the IP in your environment, review the IP activities in the table. If CA SiteMinder® is the RP, review the RP activities in the table.

The following illustration shows the sign-out request transaction flow.



Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack, which provide the FWS application functions. For more information about installing and configuring the SPS federation gateway, see the *CA SiteMinder Secure Proxy Server Administration Guide*.

When sign-out is initiated at the Resource Partner, the process flow is as follows:

Actor	Transaction Process
CA SiteMinder® as the RP	1. The user clicks a link at the Resource Partner to end the global session. The browser sends a HTTP-based wsignout request to the Signout servlet at the Resource Partner. **Note:** The RP site is receiving a wsignout message and not a wsignoutcleanup message.
	2. The RP FWS application reads the SessionId value from the SMSESSION cookie. The application renames the SMSESSION cookie to SESSIONSIGNOUT, and calls the SLO tunnel library with the wsignout request. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME] **Log message:** Performing tunnel call for WSFED signout. **Checkpoint code:** [SLOWSFED_TUNNEL_REQUEST]
	3. Based on information in the session store, the tunnel library determines the IP associated with the user session. The SLO tunnel library sets the user session state to SignoutInProgress, but does not terminate it. **Log message:** Sending signout message and awaiting response from ap for cleanup. **Checkpoint code:** [SLOWSFED_AWAITINGRESPONSE_SEND]
	4. The tunnel library returns the SignoutInProgress state message and the IP providerID and providerType. **Log message**: Performing tunnel call for WSFED signout. **Checkpoint code:** [SLOWSFED_TUNNEL_REQUEST]
	5. The RP FWS application retrieves the IP configuration data, which includes the Signout URL from the FWS cache or Policy Server.
	6. The RP FWS application redirects the browser to the Signout URL. When RP FWS (Signout Servlet) receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT. The service then calls the SLO Tunnel Service API to process the wsignoutcleanup request. **Log message**: Redirecting to signout confirmation URL. **Checkpoint code**: [SLOWSFED_LOGOUTCONFIRMURL_REDIRECT]

Actor	Transaction Process
CA SiteMinder® as the IP	7. The IP FWS application removes the SESSIONSIGNOUT cookie then posts an IP signout message and multiple RP-SignoutCleanup locations as post data to the SignoutConfirmURL JSP. The SignoutConfirmURL JSP is responsible for parsing various post variables and creating a frame-based HTML page. The primary frame in this HTML page displays the IP-SignOut message. Each of the remaining frames accesses the SignoutCleanupURL of individual RPs associated with the user session. **Log message:** Sending signout message and awaiting response from ap for cleanup. **Checkpoint code**: [SLOWSFED_AWAITINGRESPONSE_SEND] **Log message:** Sending signout cleanup message. **Checkpoint code:** [SLOWSFED_CLEANUPMESSAGE_SEND]
User Agent (browser)	8. The browser accesses SignoutCleanup service at the Resource Partner site in an individual frame.
CA SiteMinder® as the RP (continued)	9. When RP FWS (Signout Servlet) receives a wsignoutcleanup request, it renames the SMSESSION cookie to SESSIONSIGNOUT. The service then calls the SLO Tunnel Service API to process the wsignoutcleanup request. **Log message:** Renaming session cookie to sessionsignout cookie. **Checkpoint code:** [SLO_SESSION_RENAME]
	10. The SLO tunnel library processes the wsignoutcleanup request and terminates the user session from the session store. **Log message:** Terminating user session from session store. **Checkpoint code**: [SLO_USERSESSION_TERMINATE] 11. Then SLO tunnel library returns FWS with a Terminated status message indicating that the user session no longer exists in the session store. **Log message:** Redirecting to signout confirmation URL. **Checkpoint code**: [SLOWSFED_LOGOUTCONFIRMURL_REDIRECT]
	12. The FWS Signout Servlet removes the SESSIONSIGNOUT cookie and returns a 200 OK response in the frame. **Log message:** Displaying local logout message / URL. **Checkpoint message:** [SLO_LOCALLOGOUT_DISPLAY]

Note: Steps 8-12 are repeated for individual RPs simultaneously in different frames of the same HTML page.

Identity Provider Discovery Transaction Flow

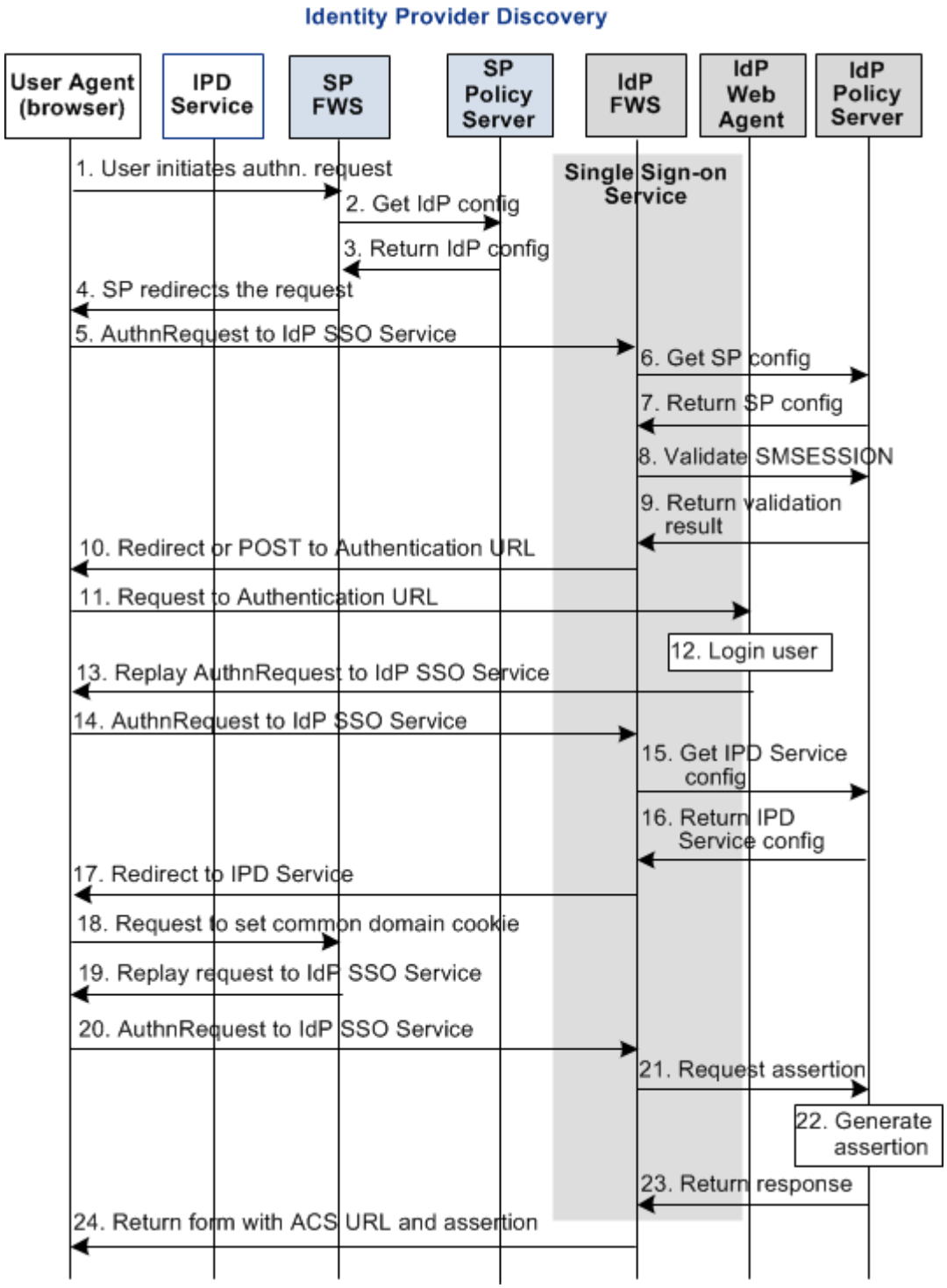
The following illustration shows the flow for a single sign-on transaction using the Identity Provider Discovery profile. The Identity Provider Discovery (IPD) profile provides a common discovery service that enables a Service Provider to select a unique IdP for authentication. A prior business agreement between partners is established so that all sites in the network interact with the Identity Provider Discovery service.

In this diagram, an Identity Provider Discovery service sits between the user and the federation components at a CA SiteMinder® Identity Provider. This flow redirects a request from an Identity Provider to the Identity Provider Discovery service to set the common domain cookie.

The flow diagram assumes the following information:

- The SP FWS redirects the user to the IdP SSO Service URL to initiate the transaction.
- SAML 2.0 HTTP POST is the single sign-on profile.
- CA SiteMinder® is shown as the IdP so you can see the processes at each partner.

The following illustration shows the flow of an Identity Provider Discovery transaction.


Identity Provider Discovery
User Agent (browser)
IPD Service
SP FWS
SP Policy Server
IdP FWS
IdP Web Agent
IdP Policy Server
Single Sign-on Service
1. User initiates authn. request
2. Get IdP config
3. Return IdP config
4. SP redirects the request
5. AuthnRequest to IdP SSO Service
6. Get SP config
7. Return SP config
8. Validate SMSESSION
9. Return validation result
10. Redirect or POST to Authentication URL
11. Request to Authentication URL
12. Login user
13. Replay AuthnRequest to IdP SSO Service
14. AuthnRequest to IdP SSO Service
15. Get IPD Service config
16. Return IPD Service config
17. Redirect to IPD Service
18. Request to set common domain cookie
19. Replay request to IdP SSO Service
20. AuthnRequest to IdP SSO Service
21. Request assertion
22. Generate assertion
23. Return response
24. Return form with ACS URL and assertion

Note: The SPS federation gateway can replace the Web Agent and Web Agent Option Pack to provide the Federation Web Services application functions. In the flow diagram, the Web Agent block would be the embedded Web Agent in the SPS federation gateway. For information about installing and configuring the SPS federation gateway, see the *Secure Proxy Server Administration Guide*.

The Identity Provider Discovery process is as follows.

Actor	Transaction Process
User Agent (browser)	1. The user clicks on a link to initiate an authentication request.
CA SiteMinder® as the SP	2. The SP FWS requests the IdP configuration information from the local Policy Server. **Log message:** Reading SAML 2.0 IDP Configuration **Checkpoint code:** [SSOSAML2_IDPCONFREAD_REQ]
	3. The local Policy Server returns the configuration information. **Note:** The SP FWS application can cache the configuration information. **Log message:** Policy server returns SAML2.0 IDP Configuration **Checkpoint code:** [SSOSAML2_IDPCONFFROMPS_RSP]
	4. The SP FWS redirects the request to the browser.
User Agent (browser)	5. The user agent (browser) requests the IdP SSO Service.
CA SiteMinder® as the IdP	6. The IdP FWS requests the SP configuration information from the local Policy Server. **Log message:** SAML2.0 SP Configuration is not in cache. Requesting to get from policy server. **Checkpoint code:** [SSOSAML2_SPCONFFROMPS_REQ]
	7. The local Policy Server returns the configuration information. **Note:** The IdP FWS application can cache the configuration information. **Log message:** Policy server returns SAML2.0 SP Configuration. **Checkpoint code:** [SSOSAML2_SPCONFFROMPS_RSP]
	8. The IdP FWS gets the SMSESSION cookie for the IdP domain and calls to the Policy Server to validate it. If there is no SMSESSION cookie, the IdP FWS skips to Step 6. **Log message**: Request to validate the session. **Checkpoint code:** [SSOSAML2_SESSIONCOOKIEVALIDATE_REQ]
	9. The Policy Server validates the SMSESSION cookie and returns the result.

Actor	Transaction Process
	10. If the SMSESSION cookie does not exist or is not valid, the IdP FWS redirects or posts to the Authentication URL obtained from the configuration. If the SMSESSION cookie is valid, the IdP FWS skips to Step 18. **Log message**: Session cookie does not exists. redirecting to authentication url **Checkpoint code**: [SSOSAML2_AUTHENTICATIONURL_REDIRECT]
User Agent (browser)	11. The browser requests the Authentication URL. The IdP Web Agent protects the Authentication URL.
CA SiteMinder® as the IdP	12. The IdP Web Agent logs the user in, setting the SMSESSION cookie and lets the request pass to the Authentication URL.
	13. The Authentication URL replays the request to the IdP SSO Service with the AuthnRequest message. **Log message:** Policy server returns the authentication request **Checkpoint code:** [SSOSAML2_GETAUTHENTICATIONREQFROMPS_RSP] **Log message:** Service redirecting to SSO URL **Checkpoint code:** [SSOSAML2_SSOURL_REDIRECT]
User Agent (browser)	14. The browser requests the IdP SSO Service. This request is equivalent to the request from step 8, but now the user has a valid SMSESSION cookie.
CA SiteMinder® as the IdP	15. The IdP FWS requests the Identity Provider Discovery Profile (IPD) configuration from the Policy Server, passing the Identity Provider ID. **Log message:** Request for IPD configuration **Checkpoint code**: [SSOIPD_READIPDCONF_REQ]
	16. The Policy Server returns with the IPD configuration, including IPD Service URL, common domain cookie, and persistence information of the common domain cookie. **Log message:** Reading IPD service URL from configuration **Checkpoint code**: [SSOIPD_READIPDSERVICEURL_REQ] **Log message:** Reading common domain cookie from configuration **Checkpoint code:** [SSOIPD_READCOMMONDOMAINCOOKIE_REQ] **Log message:** Reading persistence information of the common domain cookie **Checkpoint code:** [SSOIPD_READPERSISTENCEINFOFORCOMMONCOOKIE_REQ]
	17. The IdP FWS application redirects the call to the IPD Service URL. **Log message:** Redirecting to IPD service URL **Checkpoint code:** SSOIPD_REDIRECTTOIPDURL_REQ
User Agent (browser)	18. The browser redirects the user to the IPD Service to set the common domain cookie.

Actor	Transaction Process
Identity Provider Discovery Service	19. The IPD Service sets or updates the common domain cookie with the Identity Provider ID. The IPD Service redirects the user agent back to the IdP FWS from which it received the Set Request. **Log message:** IPD service setting common domain cookie with identity provider id **Checkpoint code:** [SSOIPD_SETCOMMONDOMAINCOOKIE_REQ]
User Agent (browser)	20. The browser makes a request to the IdP SSO Service.
CA SiteMinder® as the IdP	21. The IdP FWS requests a SAML 2.0 assertion from the Policy Server, passing the AuthnRequest through an authorize call to the realm obtained from the configuration. **Log message:** Request to policy server for generating saml2 assertion/artifact based on selected profile. **Checkpoint code:** [SSOSAML2_GENERATEASSERTIONORARTIFACT_REQ]
	22. The Policy Server generates an assertion that is based on the configuration information for the Service Provider. The Policy Server signs the assertion and returns the assertion in a response message. **Log message:** Policy server generates the saml2 assertion **Checkpoint code:** [SSOSAML2_PSGENERATEASSERTION_RSP] **Log message:** Policy server signs saml2 assertion **Checkpoint code:** [SSOSAML2_PSSIGNASSERTION_RSP]
	23. The response message is returned to the IdP FWS. **Log message:** Received the assertion/artifact response based on profile selected. **Checkpoint code**: [SSOSAML2_RECEIVEDASSERTION_RSP]
	24. The IdP FWS returns a form to the user containing the response message and the Assertion Consumer URL obtained from the configuration, and the Javascript to submit the form. **Log message:** Browser posting the response to assertion consumer url **Checkpoint code:** SSOSAML2_POSTASSERTIONTOCONSUMERURL_RSP **Note:** The Policy Server can indicate that the authentication level of the current session is too low. If the level is too low, the IdP FWS redirects to the authentication URL to facilitate step-up authentication.

After the final step in the diagram, the user agent posts the response message to the Assertion Consumer URL at the Service Provider.

Index

W