OASIS Service Provisioning Markup Language (SPML) v2 – Federated Provisioning

Draft 0.6
2006 March 06

Document identifier: pstc-spml2-fed-prov-use-cases-06

Location: http://www.oasis-open.org/committees/provision/docs/

Send comments to: pstc-comment@lists.oasis-open.org

Editor:

Jeff Bohren, BMC (Jeff_Bohren@bmc.com)

Contributors:

Richard Sand, Tripod Technology Group, Inc. (rsand@ttg.cc)

Abstract:

This specification defines usage of SPML v2 for federated provisioning.

Status:

This is a candidate Committee Specification that is undergoing a vote of the OASIS membership in pursuit of OASIS Standard status.

If you are on the provision list for committee members, send comments there. If you are not on that list, subscribe to the provision-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to provision-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Table of contents

1. Introduction

1.1. Concepts

This document describes user cases for the use of SPML V2 for federated provisioning.

2. High-level Use Cases

2.1. Pre-federation Provisioning of Accounts

In some federation environments an account needs to exist prior to the SSO event (which may never occur). Examples of this include vendor services such as 401K, paycheck services, and outsourced HR apps.

Participants	• IdP (acting as RA) • SP (acting as PSP) • End user (User) • Delegated Administrator (DA)

Preconditions	• The User has an account with the IdP • There is a trust relationship between IdP and the SP such that the SP trusts the IdP as an authoritative source of identity
Postconditions	• The User has an account with the SP
Use case flow	1. The DA defines a policy that entitles the User to an account on the SP 2. The IdP makes a provisioning request to the SP to create the account for the user
Alternative flow	

2.2. Modification of Existing Accounts

A change is required in an existing federated account.

Participants	• IdP (acting as RA) • SP (acting as PSP) • End user (User) • Delegated Administrator (DA)
Preconditions	• The User has an account with the IdP which has been modified • The User has an account with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• The user's account on the SP is modified.
Use case flow	1. The IdP makes a modification request to the SP to update the account for the user
Alternative flows	

2.3. Just-in-Time Provisioning of Accounts

In some federation environments an account may be created as needed during the SSO session.

Participants	• IdP (acting as RA) • SP (acting as PSP) • End user (User) • Delegated Administrator (DA)

Preconditions	• The User has an account with the IdP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• The User has an account with the SP
Use case flow	1. The DA defines a policy that allows the User to have an account on the SP 2. The user does SSO from the IdP to the SP (this could either be IdP-first or SP-first SSO) 3. The SP asks the IdP to make a provisioning request to create an account for the user (this could done using either be a back channel or front channel mechanism)
Alternative flow	1. The DA defines a policy that allows the User to have an account on the SP 2. The user does SSO from the IdP to the SP (this could either be IdP-first or SP-first SSO) 3. The SP makes a query to the IdP for account information in order to create the new account (this could done using either be a back channel or front channel mechanism)

2.4. Deprovisioning of Accounts

When a user is no longer entitled to an account on the SP it should be de-provisioned.

Participants	• IdP (acting as RA) • SP (acting as PSP) • End user (User) • Delegated Administrator (DA)
Preconditions	• The User has an account with the IdP • The User has an account with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• The User no longer has an account with the SP
Use case flow	2. The DA changes a policy so the user is no longer entitled to an account on the SP 3. The IdP makes a deprovisioning request to the SP to delete the account for the user
Alternative flows	

2.5. Bulk Provisioning of Accounts

In some federation environments an account needs to exist prior to the SSO event (which may never occur). Examples of this include vendor services such as 401K, paycheck services, and outsourced HR apps. There are cases where multiple users must be granted accounts at the same time, such as when there is a merger or acquisition.

Participants	• IdP (acting as RA) • SP (acting as PSP) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• There is a set of user accounts with the SP
Use case flow	1. The DA defines a policy that entitles the users to an account on the SP 2. The IdP makes a bulk provisioning request to the SP to create the accounts for the users
Alternative flow	

2.6. Bulk Deprovisioning of Accounts

When a user is no longer entitled to an account on the SP it should be de-provisioned. This may be required for a large set of users when there are no longer entitled to an account. For example there could be a lay-off or an ending of contractual relationships between two parties.

Participants	• IdP (acting as RA) • SP (acting as PSP) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a set of user accounts with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• There is no longer a set of user accounts with the SP
Use case flow	1. The DA changes a policy so that the users are no longer entitled to the accounts on the SP 2. The IdP makes a bulk deprovisioning request to the SP to delete

	the accounts for the users
Alternative flows	

2.7. Provisioning Federated Relationships

A user may already have unrelated accounts on both the IdP and SP. A federated relationship may be established between those two accounts for future SSO.

Participants	• IdP (acting as PSP) • SP (acting as RA) • End user (User)
Preconditions	• The User has an account with the IdP • The User has an account with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• There is a federated relationship between the users account on the IdP and SP
Use case flow	1. The user does SSO from the IdP to the SP (this could either be IdP-first or SP-first SSO) 2. The user indicates to the SP that a federation relationship should be made between the accounts 3. The SP makes a provisioning request to the IdP to create a federated relationship between the accounts
Alternative flow	

2.8. De-provisioning Federated Relationships

A user may already have unrelated accounts on both the IdP and SP. A federated relationship may be established between those two accounts for future SSO.

Participants	• IdP (acting as PSP) • SP (acting as RA) • End user (User)
Preconditions	• The User has an account with the IdP • The User has an account with the SP

	• There is a federated relationship between the users account on the IdP and SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	• There is no longer a federated relationship between the users account on the IdP and SP
Use case flow	1. The SP makes a provisioning request to the IdP to remove a federated relationship between the accounts
Alternative flow	

2.9. Provisioning Related Accounts

A third party may need to provision accounts to multiple service providers that should initially have a federated realationship, and that relationship may have important privacy aspects. For instance when a new hire joins a company, he may need to be provisioned with a new 401K account and a new medical insurance account. The 401K provider should not have any knowledge of the medical insurance provider and vice-versa, but a federated relationship should be established in order for a SSO session to be possible.

Participants	• IdP (acting as RA) • SP1 (acting as PSP) • SP2 (acting as PSP)
Preconditions	• The User has an account with the IdP • There is a trust relationship between IdP and the SP1 ans SP2 such that the SP trust the IdP as an authoritative source of identity
Postconditions	2. The User has an account with SP1 3. The User has an account with SP2 4. The user’s account on SP1 has a federated relationship to SP2 5. The user’s account on SP2 has a federated relationship to SP1
Use case flow	1. The IdP makes a provisioning request to the SP1 to create the user’s account 2. The IdP makes a provisioning request to the SP2 to create the user’s account 3. The IdP makes a provisioning request to the SP1 to create the federated relationship to SP2 4. The IdP makes a provisioning request to the SP2 to create the federated relationship to SP1
Alternative flow	1. The IdP makes a provisioning request to the SP1 to create the

	user's account and create the federated relationship to SP2 2. The IdP makes a provisioning request to the SP2 to create the user's account create the federated relationship to SP1

TBD add modify and de-federate

2.10. Reconciliation via Search

An IdP and SP have already established a federation relationship and 1 or more user accounts have already been provisioned from the IdP to the SP. The following use cases are all to ensure that the SP has the most up-to-date information from the IdP by use of searches.

Note: the use case flows given below do **not** detail the sequence of *iteration* requests/responses that may result from a given search.

2.10.1. Reconcile Modifications via Search from SP

The SP does a search to determine which user accounts have been modified since the last reconciliation. To do this, the IdP must either:

1. maintain timestamps of the last modification to each user account that the requesting SP's *updateSince* attribute can be compared to

   or

2. provide a *token* in its responses to the SP's that the SP's can then supply in subsequent requests; the IdP uses this token to internally keep track of all updates since the token was issued, and can then respond to any request that provides such a token with exactly the updates necessary

It is entirely up to the IdP to implement one or both of these features however it sees fit.

Participants	• IdP (acting as PSP) • SP (acting as RA) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a set of user accounts with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	6. All user accounts that exist at the IdP and should be federated with the SP (by policy) exist at the SP 7. All updates to user accounts at the IdP are sychnronized at the SP
Use case flow	1. The DA defines a policy that entitles the users to an account on the SP

	2. The SP makes an update request to the IdP to create a list of all of the user accounts (by some primary key such as User ID or e-mail address) that have been modified since the *updatedSince* timestamp provided by the requestor 3. The IdP responds with any accounts which have been modified at the IdP since they were last reconciled at the SP (by comparing the timestamp attributes)
Alternative flow	2. The SP makes an update request to the IdP to create a list of all of the user accounts (by some primary key such as User ID or e-mail address) that have been modified since the last response from the IdP identified by *token* a. The *token* is a value provided by the IdP to the SP in a previous response to a similar query b. The SP supplies the token in the current request to the IdP 3. The IdP responds with any accounts which have been modified at the IdP since they were last reconciled at the SP

2.10.2. Alternate Use Cases

The alternate use cases show reconcile via search flows that can occur between an IdP and SP without using the *token* or *updateSince* mechanisms.

The first two alternate use cases use a combination of search and provisioning requests to ensure that accounts exist and are federated on each side of the federation between IdP and SP. This will handle added & deleted accounts, and federating / defederating accounts, but does not handle other modifications to accounts (such as enabling / disabling or modification to other account attributes).

The third case uses update requests to do the above and also maintain modification synchronization between both parties.

Alternate A - Reconcile via Search from IdP

The IdP (acting as the RA) sends a search request to the SP to determine which user records it is missing

Participants	• IdP (acting as RA) • SP (acting as PSP) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a set of user accounts with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	8. All user accounts that exist at the IdP and should be federated with

	the SP (by policy) exist at the SP
Use case flow	1. The DA defines a policy that entitles the users to an account on the SP 2. The IdP makes a search request to the SP to create a list of all of the user accounts (by some primary key such as User ID or e-mail address) that should be federated to the SP 3. The IdP compares this list to its local user repository, to a. determine which users accounts are missing, or b. which exist but have not been federated 4. The IdP makes a [batch] provisioning request to the SP to a. create accounts for any missing users i. individual requests as per 2.1, *Pre-federation Provisioning of Accounts* ii. bulk requests as per 2.4, *Bulk Provisioning of Accounts* b. create a federated relationship between the user accounts where missing as per 2.6, *Provision Federated Relationships*
Alternative flow	

Alternate B - Reconcile via Search from SP

The SP (acting as the RA) sends a search request to the IdP to determine which user records it is missing. The SP then follows up with appropriate provisioning requests (as per 2.1, 2.4, and/or 2.6) to complete the reconciliation.

Participants	• IdP (acting as PSP) • SP (acting as RA) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a set of user accounts with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	9. All user accounts that exist at the IdP and should be federated with the SP (by policy) exist at the SP
Use case flow	1. The DA defines a policy that entitles the users to an account on the SP 2. The SP makes a search request to the IdP to create a list of all of the user accounts (by some primary key such as User ID or e-mail

	address) that should be federated to the SP 3. The SP compares this list to its local user repository to determine which users accounts are missing or have not been federated 4. The SP makes a followup search request to the IdP for all attributes of any missing users 5. The SP uses the results of the above search to add any missing users, or create federated relationships between the user accounts where missing
Alternative flow	If the IdP is also capable of acting as the RA and the SP as the PSP, then steps 4 and 5 above can be replaced with: 4. The SP sends a query to the IdP asking it to send provisioning requests with the user accounts missing as per 2.1, 2.4 and/or 2.6. 5. The IdP complies as per step 4 in Alternate A above

Alternate C - Reconcile Modifications via Search

The IdP does a search to determine which user accounts have been modified locally since the last reconciliation. To do this, the IdP must provide a user attribute that holds the timestamp of the last modification to that user account that has been provisioned at the SP. The IdP has provisioned this timestamp in any previous provisioning transactions with the SP, so the IdP now will query this timestamp so it can determine what changes have happened locally and need to be reconciled to the SP.

Participants	• IdP (acting as RA) • SP (acting as PSP) • Delegated Administrator (DA)
Preconditions	• There is a set of user accounts with the IdP • There is a set of user accounts with the SP • There is a trust relationship between IdP and the SP such that the SP trust the IdP as an authoritative source of identity
Postconditions	10. All user accounts that exist at the IdP and should be federated with the SP (by policy) exist at the SP 11. All updates to user accounts at the IdP are sychnronized at the SP
Use case flow	1. The DA defines a policy that entitles the users to an account on the SP 2. The IdP makes a search request to the SP to create a list of all of the user accounts (by some primary key such as User ID or e-mail address) and their last-modified timestamp that should be federated to the SP

	3. The IdP compares this list to its local user repository, to a. determine which users accounts are missing (these are handled as per 2.8.1) b. which exist but have not been federated (these are handled as per 2.8.1) c. find any federated accounts which have been modified at the IdP since they were last reconciled at the SP (by comparing the timestamp attributes) 4. The IdP makes a [batch] provisioning request to the SP to modify the user accounts that were found to have been updated at the IdP since they were last reconciled as per [… *use case for modify/update account*]
Alternative flow	

Appendix A. References

[AES] National Institute of Standards and Technology (NIST), FIPS-197: Advanced Encryption Standard, **http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf**, National Institute of Standards and Technology (NIST)

[ARCHIVE-1] OASIS Provisioning Services Technical Committee, email archive, **http://www.oasis-open.org/apps/org/workgroup/provision/email/archives/index.html**, OASIS PS-TC

[DS] IETF/W3C, *W3C XML Signatures*, **http://www.w3.org/Signature/**, W3C/IETF

[DSML] OASIS Directory Services Markup Standard, *DSML V2.0 Specification*, **http://www.oasis-open.org/specs/index.php#dsmlv2**, OASIS DSML Standard

[GLOSSARY] OASIS Provisioning Services TC, *Glossary of Terms*, **http://www.oasis-open.org/apps/org/workgroup/provision/download.php**, OASIS PS-TC

[RFC 2119] S. Bradner., *Key words for use in RFCs to Indicate Requirement Levels*, **http://www.ietf.org/rfc/rfc2119.txt**, IETF

[RFC 2246] T. Dierks and C. Allen, *The TLS Protocol*, **http://www.ietf.org/rfc/rfc2246.txt**, IETF

[SAML] OASIS Security Services TC, **http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security**, OASIS SS-TC

[SOAP] W3C XML Protocol Working Group, **http://www.w3.org/2000/xp/Group/**

[SPML-Bind] OASIS Provisioning Services TC, SPML V1.0 Protocol Bindings, **http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc**, OASIS PS-TC

[SPML-REQ] OASIS Provisioning Services Technical Committee, *Requirements*, **http://www.oasis-open.org/apps/org/workgroup/provision/download.php/2277/draft-pstc-requirements-01.doc**, OASIS PS-TC

[SPML-UC] OASIS Provisioning Services Technical Committee, *SPML V1.0 Use Cases*, **http://www.oasis-open.org/apps/org/workgroup/provision/download.php/988/drfat-spml-use-cases-05.doc**, OASIS PS-TC

[SPMLv2-Profile-DSML] OASIS Provisioning Services Technical Committee, SPMLv2 DSMLv2 Profile, OASIS PS-TC

[SPMLv2-Profile-XSD] OASIS Provisioning Services Technical Committee, SPML V2 XSD Profile, OASIS PS-TC

[SPMLv2-REQ] OASIS Provisioning Services Technical Committee, Requirements, OASIS PS-TC

[SPMLv2-ASYNC] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Async Capability of SPMLv2, OASIS PS-TC

[SPMLv2-BATCH] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Batch Capability of SPMLv2, OASIS PS-TC

[SPMLv2-BULK] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Bulk Capability of SPMLv2, OASIS PS-TC

[SPMLv2-CORE] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Core Operations of SPMLv2, OASIS PS-TC

[SPMLv2-PASS] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Password Capability of SPMLv2, OASIS PS-TC

[SPMLv2-REF] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Reference Capability of SPMLv2, OASIS PS-TC

[SPMLv2-SEARCH] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Search Capability of SPMLv2, OASIS PS-TC

[SPMLv2-SUSPEND] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Suspend Capability of SPMLv2, OASIS PS-TC

[SPMLv2-UPDATES] OASIS Provisioning Services Technical Committee, XML Schema Definitions for Updates Capability of SPMLv2, OASIS PS-TC

[SPMLv2-UC] OASIS Provisioning Services Technical Committee., SPML V2.0 Use Cases, OASIS PS-TC

[WSS] OASIS Web Services Security (WSS) TC, **http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss**, OASIS SS-TC

[X509] RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile, **http://www.ietf.org/rfc/rfc2459.txt**

[XSD] W3C Schema WG ., *W3C XML Schema*, **http://www.w3.org/TR/xmlschema-1/** W3C

Appendix B. Acknowledgments

The following individuals were voting members of the Provisioning Services committee at the time that this version of the specification was issued:

Jeff Bohren, BMC
Robert Boucher, CA
Gary Cole, Sun Microsystems
Rami Elron, BMC
Marco Fanti, Thor Technologies
James Hu, HP
Martin Raepple, SAP
Gavenraj Sodhi, CA
Kent Spaulding, Sun Microsystems
Richard Sand, Tripod Technology Group

Appendix C. Notices