WHITE PAPER

Federated Provisioning: The Synergy of Identity Federation and User Provisioning

Ping Identity™

Ping Identity™

Overview

This paper provides a background of federation and provisioning concepts, functional operations, and standards. A complex provisioning use case is presented to demonstrate the potential combination of SPML and SAML in a federated environment. While many technical options exist to solve a federated provisioning use case, this paper discusses several factors that present just-in-time SAML-based provisioning as a feasible starting point.

This paper is targeted towards human resource, application, identity, or operations teams responsible for user provisioning or identity federation.

Background

Secure Internet single sign-on via identity federation and provisioning are two important models for identity management *within* and *across* enterprises—both becoming more and more relevant to enterprises as business processes extend beyond the enterprise boundary.

We will use the following definitions for these two models of identity management:

Federation - *Federated identity refers to the standards, agreements and processes by which identity management responsibilities can be shared between various policy domains to enable user convenience, cost-saving, and regulatory compliance.*

Provisioning - *Provisioning is the automation of all the lifecycle steps required to manage (setup, amend, and revoke) user or system access entitlements or data for a set of information systems.*

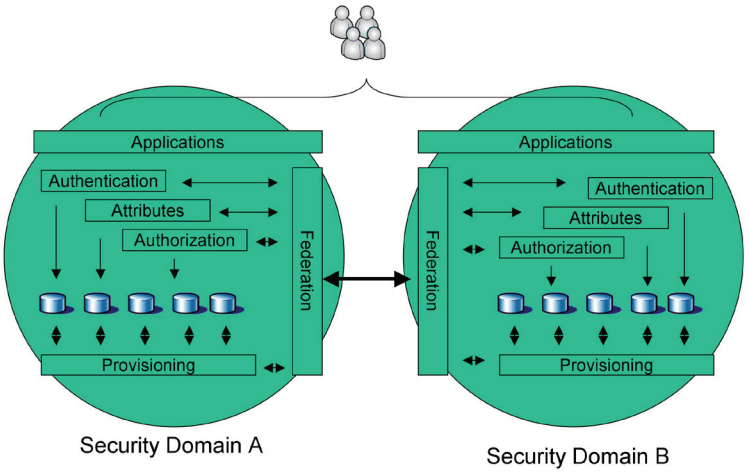
Over generalizing, the provisioning style of identity management (at least in its tradtional guise) is typically found *within* enterprises. Provisioning mechanisms used to ensure enterprise employees, contractors, etc., are able to access the systems and resources they need in order to perform their roles throughout the full lifecycle of their employment. At its most basic level, provisioning technologies automate the previously-manual responsibilities of the HR and IT departments to ensure that enterprise employees (and contractors, partners, customers, etc.) have appropriate access (but no more) to the relevant system and application resources in order to fulfill their job duties. Once provisioned with the appropriate account objects for a particular user, a system or application assumes responsibility for subsequent authentication and authorization steps. One implication is that the various systems into which a user is provisioned must maintain and manage accounts for that user going forward. As a single enterprise is more likely to consist of a single policy domain, the provisioning model is likely to presume a master/slave relationship between policy administration and enforcement.

Federation, on the other hand, more typically refers to a model for identity management *between* different enterprises. As such, it makes different assumptions about what is an equitable and appropriate distribution of policy rights and responsibilities. For example, Enterprise A may not wish to bear the burden of maintaining (and supporting, e.g., password resets, etc.) separate accounts for the multiple employees of a business partner Enterprise B for whom their job duties might have them only occasionally attempting to access some Enterprise A resources. The more seldom an Enterprise B employee visits

Enterprise A, the more certain will be the cost of password resets for Enterprise A. Federation shifts the management burden for Enterprise B employees away from Enterprise A and back to Enterprise B, where it more appropriately belongs—to *its* users.

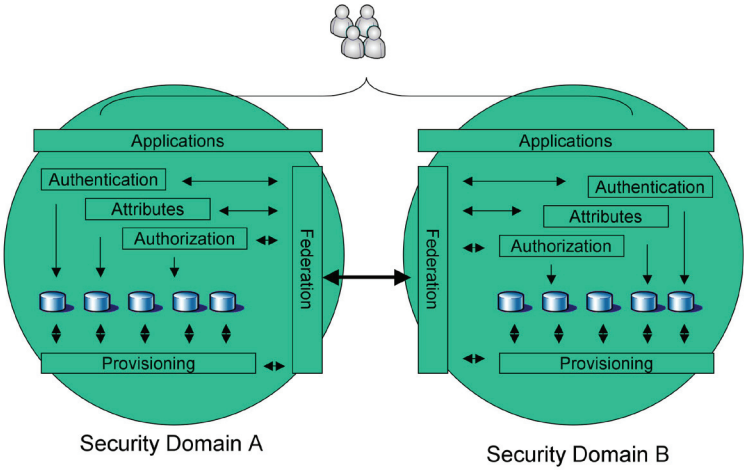
Standards exist for both federation and provisioning models: SAML 2.0 (standardized under OASIS) and WS-Federation (submitted to OASIS at time of writing) for federation and SPML (standardized under OASIS) for provisioning. For the most part (at least in current implementations and deployments), the two models are mutually exclusive, with little interplay or integration between domains and standards. As a result, deployers have been forced to use proprietary mechanisms where, ideally, standardized integration profiles would exist.

The following diagram represents these models as they are currently applied today. Provisioning supports identity transactions within each respective security domain, while federation deals with user transactions between the domains.



Both federation and provisioning models have their strengths and weaknesses, and consequently different value propositions. Specifically, this paper will highlight how the provisioning model (historically focused *within* the enterprise), when applied *between* enterprises, can significantly enable the federated model.

The following diagram is an attempt to describe this fundamental change:



Standards

As mentioned above, a number of open standards exist to address federated identity and user provisioning:

SAML

The Organization for the Advancement of Structured Information Standards (OASIS) developed SAML as an XML-based specification for exchanging security information. Currently at Version 2, SAML defines syntax and exchange mechanisms for three kinds of assertions:

1. Authentication assertions, which are declarations about a user's identity
2. Attribute assertions, which contain particular details about a user
3. Authorization decision assertions, which specify what the user is allowed to do on a particular site

SAML is a flexible and extensible protocol designed to be used—and customized if necessary—by other standards. The Liberty Alliance, the Internet2 Shibboleth project and the OASIS Web Services Security (WSS) committee have all defined how SAML can be profiled for particular usages.

Previous versions of SAML (prior to SAML 2.0) relied on out-of-band agreement on the types of identifiers that would be used to represent a federated identity between partners (e.g., the use of X.509 subject names). While they supported the use of federated identities, they provided no means to directly establish the identifiers for those identities using SAML message exchanges. SAML 2.0 introduced constructs and message formats to support the dynamic establishment and management of federated name identifiers.

The SAML Technical Overview discusses the different models that SAML supports for *connecting* a user's account between two different providers.

WS-Federation

WS-Federation Passive Requestor Profile ("WS-Federation" for short in this paper) is a component of the so-called WS-* suite of Web service specifications spearheaded by Microsoft and IBM. WS-Federation provides comparable functionality to SAML's Web Browser SSO Profiles. At its most basic level, WS-Federation can be considered a profile of WS-Trust for browser-based account linkage and SSO, extending WS-Trust to allow the sharing of tokens containing pseudonyms and attributes.

WS-Federation is most noteworthy because of Microsoft's support for it within its Active Directory Federation Server (ADFS), thereby enabling SSO between mixed Windows and non-Windows environments.

At the time of writing, a proposed charter for a Web Services Federation (WSFED) Technical Committee is under discussion within OASIS. WS-Trust, on which WS-Federation is based, is an OASIS standard.

SPML

OASIS developed the Service Provisioning Markup language (SPML) as an open XML-based standard protocol for the communication of provisioning operations between actors.

Currently at Version 2, the general model adopted by SPML is one of clients performing protocol operations against servers. In this model, a client issues a SPML request describing the operation to be performed at a given service point. The service point is then responsible for performing the necessary operation(s) to constitute the implementation of the requested service. Upon completion of the

operation(s), the service point returns to the client an SPML response detailing any results or errors pertinent to that request.

The operational components of an SPML model system are shown below. An SPML Requesting Authority (RA) constructs an SPML document containing a Provisioning Service Object (PSO) to a pre-defined service offered by a Provisioning System Point (PSP). In order to fulfill the request, this PSP may take the data passed in this SPML document, construct its own SPML document and send it to Provisioning Service Target (PST). The PST represents an independent resource that provides an SPML-compliant service interface. Alternatively, the PSP could use a non-SPML mechanism to interact directly with a resource to fulfill the request from the RA.

Federation & Provisioning Models

This section discusses the variations in both the federation and provisioning models of identity management.

Federation Models

As discussed previously, the federation model manifests itself as the exchange of some set of identity attributes between enterprises (one acting as an Identity Provider, the other as a Service Provider) in order to enable appropriate services and access for users at the Service Provider. Different flavors of federation vary in the nature of this identity, with consequent implications for privacy, scalability and maintainability.

Federation models can be (simply) categorized as either attribute-based or identifier-based.

Attribute-based

Providers can agree to refer to users only through attributes that describe their positions, roles and entitlements rather than by a specific identifier. Generally, the advantage of this model for a Service Provider enterprise is that it will be freed from the burden of tracking user accounts/passwords for the employees of its business partners and can instead define permissions in terms of the more general (and likely, less variable) attributes and roles. For the Identity Provider enterprise, the benefit is that it need not establish and manage identifiers for individual employees, along with the consequent management burden associated with employee job changes, turnover, etc.

Identifier-based

Instead of using attributes or roles to refer to users, providers can choose to use persistent identifiers.

Global Identifier

Providers can agree to use a *global* identifier for a particular user. The identifier is global if the same value is used for the same user by other providers. For example, if a company enables SSO for its employees to all its business partners by using the employee number as the identifier, that employee number is a global identifier. Critically, such an identifier enables inappropiate collusion regarding particular employee's activities in a way that pseudonyms do not. For instance, if a large manufacturer used a global identifier for its employees with its suppliers, those suppliers might be able to gain a competitive advantage over the manufacturer through analysis of the aggregated buying patterns of particular employees.

Pseudonymous Identifier
Enterprises can agree to refer to users through randomly generated and opaque pseudonyms. Rather than an enterprise using a meaningful identifier such as an employee number in its SSO assertions, it would use a random string unique to that business partner. Using pairwise unique pseudonyms prevents trivial correlation between multiple business partners, thereby better protecting employee privacy and potentially competitive advantage than the global model.

Relative to the attribute-based model, referring to users by a specific identifier (whether global or pseudonymous) allows the actions of particular employees to be tracked and differentiated. This is likely relevant for transactions of high value or sensitivity for which regulatory audit requirements may impose such individual tracking.

Provisioning Models
Somewhat (but not completely) orthogonal to the nature of the identity shared for a user between two enterprises is the mechanism by which any necessary identifiers and/or attributes are *first* established and subsequently managed, i.e., how is the user account initially established, changed and subsequently deleted.

Provisioning models can be categorized along two (mostly) orthogonal axes: trigger and multiplicity.

Trigger
Provisioning models can differ on what initiates or triggers a provisioning operation. We refer to the two models as Just-In-Time and Batch.

Just-in-Time
As the name suggests, the "Just-in-Time" (JIT) provisioning model has the provisioning operation happening only at such time as it is first needed. Therefore, there is no need for any prior messages between the two enterprises (although no message is required for the establishment of this identifier, there will almost certainly be other communications between the two enterprises in order to enable SSO, etc., between them). Instead, the two enterprises establish the identifier simultaneously (or almost so) the first time it is required. As a result, the JIT model may serve to distribute the messaging and processing load associated with provisioning—the vagaries of user access—ensuring that peaks and valleys of load are smoothed out.

A key aspect of the JIT model is that the operation is typically (but not exclusively) triggered by the user himself. As an example, for an employee of one enterprise to have their local account federated to a business partner, they would typically login to their own employee account. At that point, they would be given the opportunity to SSO to one or more business partners (the list of which would be customized to their job position). Along with their browser request for the remote resource would be an assertion carrying a newly-generated pseudonym (and any other relevant attributes). The business partner, recognizing that the pseudonym is one it had not previously seen, would create a new account and assign privileges commensurate with the communicated attributes.

Note: The next time that employee SSO's in, there would be no need to communicate attributes; the pseudonym alone would be sufficient to allow the business partner to recognize the employee and grant them the right permissions.

The fact that the user likely plays a role in the establishment of an identifier and the sharing of attributes makes the JIT model particularly appropriate when such identity sharing should only occur with the explicit consent of the employee. For example, in establishing federated identifiers with an outsourced health benefits provider, an enterprise might choose the JIT model to ensure that its employees are suitably informed as to the nature of the identity shared, etc.

Batch

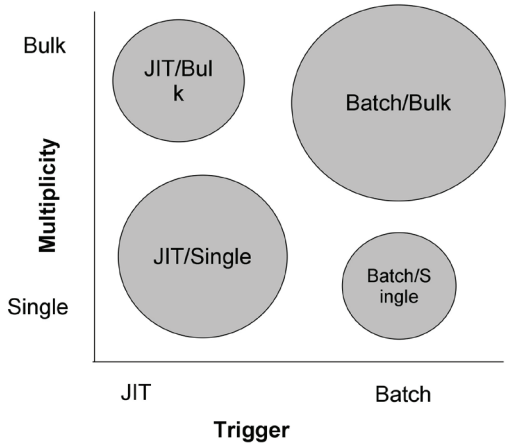
In the batch-triggered provisioning model, the provisioning operations are initiated not by the actions of the user or some administrator, but rather by some other criteria—typically time-based. For instance, two enterprises might exchange an XML document with account information for appropriate employees every night at midnight. The document could be retrieved by FTP or comparable channel by the targeted business partner, the document then parsed as input to the process of creating suitable accounts for those employees.

More and more, rather than a static XML document exchanged between partners, the batch model manifests itself as the exchange of XML messages—either pushed/pulled between enterprise partners.

Multiplicity

Distinct from the temporal categorization of provisioning operations are the multiplicity of the documents/messages, i.e., are accounts provisioned singly or as part of a bulk operation along with many other colleagues?

The following diagram identifies the relationship between the temporal and multiplicity aspects of provisioning:



- SAML's pseudonymous federation mechanism is an example of the **JIT/ Single** combination – the federated identifier is established for a particular user when first made necessary by that user's actions and not sooner.
- The **JIT/Bulk** combination implies a provisioning operation initiated on an on-demand basis for a large number of employees—as might occur when an HR admin enters a large number of employees in an acquisition scenario.
- The **Batch/Single** combination implies a provisioning operation for a single employee, but triggered by some predefined criteria and not the actions of that user. While likely less common, this mode might be relevant in situations where the specifics of a given account warrant a provisioning operation (e.g., update the corresponding pseudonym when the number of federated sign-on operations exceeds some threshold).
- The **Batch/Bulk** combination is that in which a large number of separate accounts are provisioned simultaneously based on some pre-defined trigger criteria.

Importantly, different provisioning milestones of a full account lifecycle may use different models. For instance, a pseudonym might be established using JIT/Single, but subsequent management operations use Batch/Bulk.

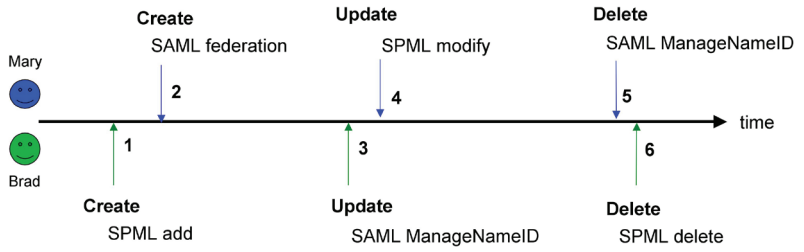
Federated Provisioning at Work

We can examine the relevance of the federation and provisioning models presented above by considering key milestones in the lifecycle for two different enterprise employees of Company A, specifically in the context of their interactions with Company A's business partner Enterprise B.

Both Mary and Brad's job duties at Company A require them to deal with Enterprise B–Mary as a lawyer and Brad as a buyer. We will see that Jane and Brad's different circumstances and roles within Company A will determine that the mechanisms by which these interactions are provisioned and occur are different.

The lifecycles of the linkage between both enterprises for both Mary and Brad will have operations comparable to create, update and delete. However, the specifics of how these operations occur will differ, specifically in how they are manifested in standardized messages between Company A and Enterprise B.

The following figure shows the timeline for both Mary and Brad:



Milestone 1

Company A and Enterprise B establish a federated identifier for Brad. Brad is one of the first small group of employees for which federated SSO between the two enterprises is enabled, so the two providers use an SPML message to create his account at Enterprise B and to establish the initial federated pseudonym for him. Similar accounts for Brad's Company A colleagues are also created, each with a different pseudonym.

The SPML message from Company A to Enterprise B might appear as follows:

```
<spml:batchRequest>
  <spml:addRequest requestID="aa">
   <spml:data>
    <saml:NameID type="persistent">jf65ghY</saml:NameID>
   <spml:data>
  </spml:addRequest>
  <spml:addRequest requestID="bb">
   <spml:data>
    <saml:NameID type="persistent">s3gF54L</saml:NameID>
   <spml:data>
  </spml:addRequest>
</spml:batchRequest>
```

The <spml:batchRequest> element acts as a container for the individual <spml: addRequest> elements, each differentiated by a 'requestID' so that they can be treated separately with respect to returning data or errors.

Note: SPML uses 'batch' in its <batchRequest> message name to refer to what we have called 'bulk' operations. As indicated earlier, the two often (but not always) occur together.

Milestone 2
Company A and Enterprise B establish a federated identifier for Mary. As Mary is a new hire and her job responsibilities are not completely certain, Company A decides to not provision an account for her at Enterprise B immediately. Instead, Company A will request that an account be created for her only at such time as first necessary. The first time that Mary clicks on the 'Enterprise B' icon on her customized Company A intranet page, her browser gets redirected to Enterprise B with an unsolicited SAML Response message that will serve to provision her account there.

The SAML message from Company A to Enterprise B might appear as follows:

```
<saml:Response inResponseTo="">
 <saml:Assertion>
  <saml:Subject>
   <saml:NameID>k3Gf5lU</saml:NameID>
  </saml:Subject>
  <saml:AuthnStatement>
  </saml:AuthnStatement>
 </saml:Assertion>
</saml:Response>
```

Upon receipt of the above, Enterprise B would create the necessary account for Mary and be prepared for future subsequent SSO operations in which Company A would again present the 'k3Gf5lU' identifier.

Milestone 3
Company A updates Brad's account at Enterprise B using SAML's NameIDManage message structure. The federated pseudonym previously established (through the inital SPML message) is updated due to Company A's policies for protecting Brad's privacy. The update operation is initiated as a result of Brad's frequent interactions with Enterprise B; at some predefined threshold of visits, the update message is kicked off.

The SAML message from Company A to Enterprise B might appear as follows:

```
<saml:ManageNameIDRequest>
 <saml:NameID>jf65ghY</saml:NameID>
 <saml:NewID>k7gd34Hg</saml:NewID>
</saml:ManageNameIDRequest>
```

Enterprise B would discard the previously established 'jf65ghY' identifier for Brad and replace it with the new ' k7gd34Hg' identifier.

Milestone 4
Company A uses the bulk functionality found within SPML to update the Enterprise B accounts of Mary and other colleagues.

The SPML message from Company A to Enterprise B might appear as follows:

```
<spml:batchRequest>
  <spml:modifyRequest requestID="aa">
   <psoID ID="2244" targetID="target2"/>
   <spml:modification modificationMode="replace">
    <spml:component path="Person/ID">
    <spml:data>
     <saml:NameID type="persistent">jf65ghY</saml:NameID>
    <spml:data>
   </spml:component>
   </spml:modification>
```

```
  </spml:modifyRequest>
....
</spml:batchRequest>
```

Milestone 5
To account for Mary taking a leave of absence, Company A defederates her account at Enterprise B using SAML's NameIDManage message structure. Company A defederates Mary by sending the request to Enterprise B indicating that it will no longer use the 'jf65ghY' identifier.

The SAML message from Company A to Enterprise B might appear as follows:

```
<saml:ManageNameIDRequest>
 <saml:NameID>jf65ghY</saml:NameID>
</saml:ManageNameIDRequest>
```

It is the absence of a <saml:NewID> that indicates that this is a delete operation.

It is worth noting that, were Mary able to authenticate to Enterprise B through some other mechanism (e.g., some preexisting account/password that predated SAML SSO) then simply defederating her may not guarantee that her ability to access Enterprise B is removed.

Milestone 6
Company A, finding itself doing less business with Enterprise B, decides to reduce the number of its lawyers managing the relationship. Company A uses the batch functionality found within SPML to delete the Enterprise B accounts of Mary and other colleagues having the same role.

The SPML message from Company A to Enterprise B might appear as follows:

```
<spml:batchRequest>
  <spml:deleteequest requestID="aa">
   <psoID ID="2244" targetID="target2"/>
  </spml:deleteRequest>
  <spml:deleteequest requestID="bb">
   <psoID ID="2236" targetID="target4"/>
  </spml:deleteRequest>
</spml:batchRequest>
```

Implementation Considerations

While the standards and technologies exist today to implement a complex scenario like the one presented above, a phased approach can produce immediate ROI. As with all federated relationships and business process flows, implementation of compatible software is a pre-requisite for both partners. Today, the availability of SAML-capable federation infrastructure has achieved near ubiquity in enterprise environments. However, SPML is available in some commercial software products but is still in the very early adopter phase.

Many partners are already using SAML to implement the JIT/Single use case with a few additional configurations in their federation server. This provides a full lifecyle where users are provisioned and deprovisioned as needed with each SSO event. In this model, existing proprietery batch process can be maintained and triggered as needed.

Ultimately, SPML should be layered on top of SAML to provide a mixed mode of operation similar to the "Provisioning at Work" scenario. When this has

been successfuly patterned in a pair-wise relationship, the ROI, business case and phased approach can be presented to other partners. Over time, this will enable a reduction and consolidation of the IT resources devoted to one-off and proprietary inter-enterprise provisioning processes.

Conclusions

This paper has examined the federated identity model and its variations, as well as the provisioning model and its variations.

Although often presented as mutually exclusive, these two models need not be—the provisioning model, when applied between enterprises, can potentially be a significant enabler of the federated model.

In short,

- The SPML-based provisioning model applied between business partners can enable federated identity operations.
- The provisioning capabilities of SAML, while less powerful, can still be useful and, in principle, enterprises could use either as applicable.

Consider a phased approach starting with JIT Provisioning with SAML that can be rolled out across your partner base today.

About Ping Identity Corporation

Ping Identity's dedication to delivering secure Internet single sign-on software and services for over 150 customers worldwide has earned us recognition as the market leader in federated identity management. PingFederate®, the world's first rapidly deployable identity federation software, provides organizations' users safe access to Internet applications without the need to re-login. With PingFederate and PingEnable, Ping Identity's expert support, services, and methodologies, external connections can be operational in less than a week. Download a free trial at www.pingidentity.com. For more information, dial toll-free 877.898.2905 or + 1 303.468.2882, or email sales@pingidentity.com.

Ping Identity™