Peer-to-Peer Provisioning

Problem Statement

Provisioning systems are currently designed with a master-slave relationship between themselves and the resources that they provision. Traditionally, the only true peer-to-peer relationship between the provisioning system and another system is with the provisioning systems relationship with an identity management (human resources) system. In this relationship, the identity management system will notify the provisioning system of identity changes, which may then trigger provisioning actions to take place against its managed resources.

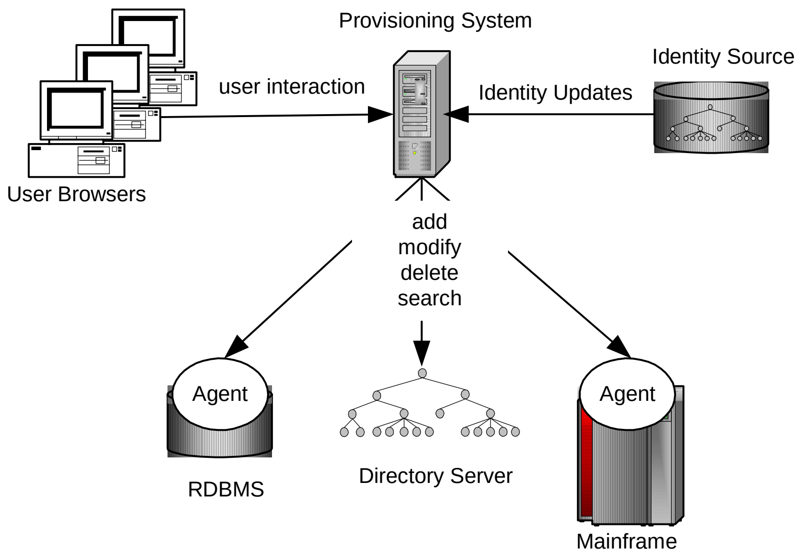

Figure 1: Typical Enterprise Provisioning Architecture

When solving the provisioning problem for an enterprise that is primarily concerned with internal or intranet resources, this architecture is sound. Problems may arise when an enterprise is also concerned with extranet resources and external partners access to these resources. In many situations, the enterprise can dictate to its partners to use the enterprise provisioning system to provision these extranet resources. The external identities can be added manually or through an additional identity feed through the existing identity management system interface. However, just as provisioning the resources through their direct interfaces has proven not to scale, this centralized provisioning architecture may also prove not to scale as enterprise collaborative relationships become more and more dynamic in the years to come. The key issue is that the business partners are working with more than one organization, therefore they end up with multiple delegated consoles and identity feeds.

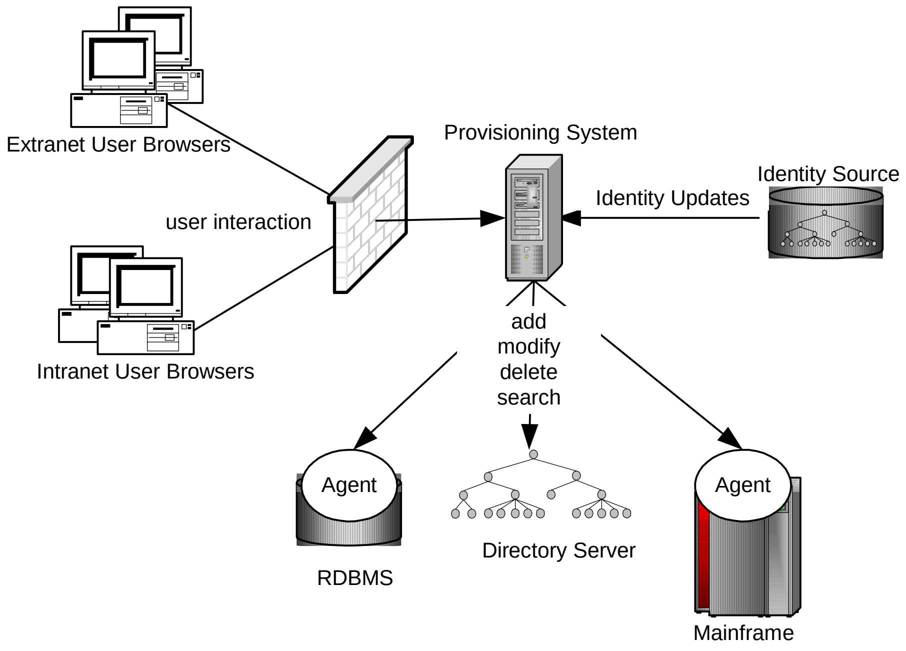

Figure 2: Typical Extranet/Intranet Provisioning Architecture

As more enterprises deploy their own provisioning systems, the need for a peer-to-peer relationship with partner enterprise provisioning systems will become more apparent. It may also be necessary in the cross-system provisioning environment to communicate between systems from two different vendors. For this reason, an easy to interpret XML schema and open standard transport protocols will be supported.

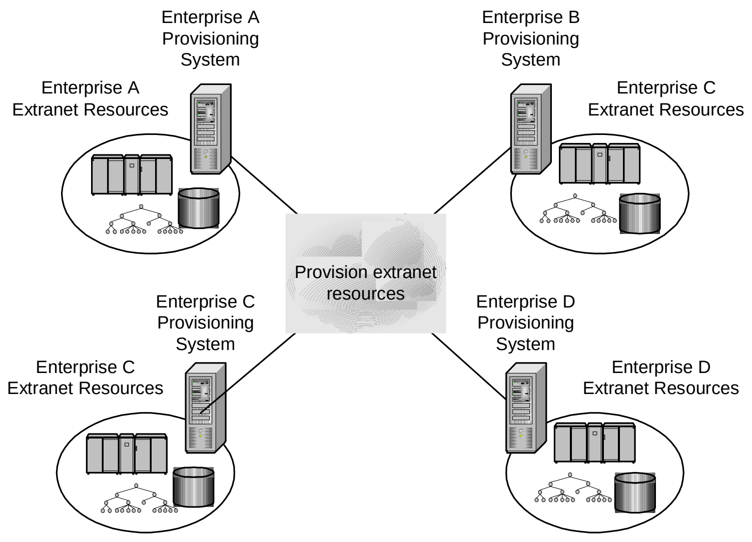
Enterprise A Provisioning System
Enterprise A Extranet Resources
Enterprise B Provisioning System
Enterprise C Extranet Resources
Provision extranet resources
Enterprise C Provisioning System
Enterprise C Extranet Resources
Enterprise D Provisioning System
Enterprise D Extranet Resources

Figure 3: Future Collaborations

Another key for peer-to-peer provisioning is to minimize the need, if any, to replicate identity across provisioning systems. Of course, identity must be used within the provisioning process, but replication of the identity on both peer systems should not be required. This will be addressed by introducing a trusted source of identity within the process of peer-to-peer provisioning. This does not require involvement of a third system, although it could, but that either of the two peers or a third party system is used as the trusted source. The establishment of this trust relationship is key to this communication, but will not be included within this interface definition. It will be assumed that this relationship has been established out-of-band.

Peer-to-Peer Provisioning Solution

The goal of this solution is to make it transparent to the end-user whether the local provisioning system or an external one is managing the resources being provisioned. In order to meet this goal, two areas of focus are needed, administration on both systems to establish trusted communications and the communications themselves. However in this revision, the area of administration will be accomplished out-of-band and not described here. The communications and interaction between the two systems will be the primary focus.

The interaction can be described first with the assumption that the administration on both systems mentioned above has already been implemented. The interaction can be separated into two different models of execution, a push model and a pull model. Both models begin with the provisioning request issued by the requesting system to the owner system. In the push model, the resource identifier, the provisioning parameters of that resource, and the assertions of the subject user attempting to be provisioned are all passed within the provisioning request. If the subject user is authorized access to the requested resource, the owner system will then provision that resource with the supplied parameters and any additional parameters it specifies for the user.

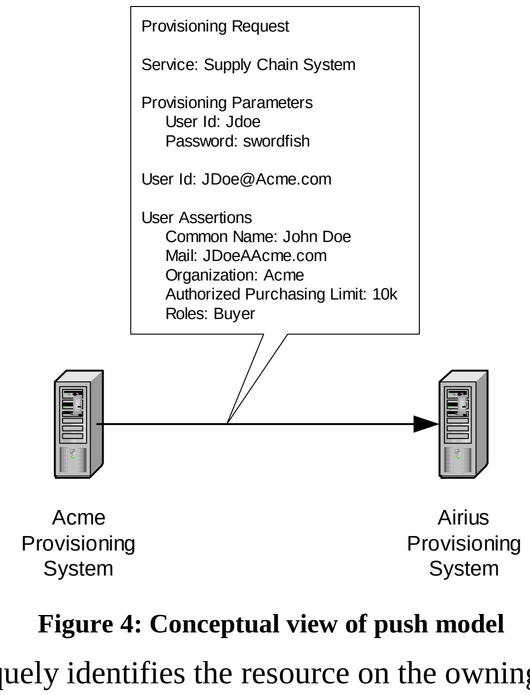

Figure 4: Conceptual view of push model

The resource identifier uniquely identifies the resource on the owning system. The provisioning parameters are properties of the user within the context of the resource to be provisioned. The user id and password to be used for authentication to the resource are examples of provisioning parameters. The user assertions are attributes of the user requesting to be provisioned. This information can be used by the owner system to determine whether the user is authorized access to the resource.

In the pull model, the requesting system provides the above-mentioned information in its provisioning request except user assertions. Instead, only the user's reference id is delivered to the owner system. The owner system then can request the assertions about the user that are needed for the owner system to make an authorization decision. Several requests may be issued if a step-wise decision process is implemented by the owner system.

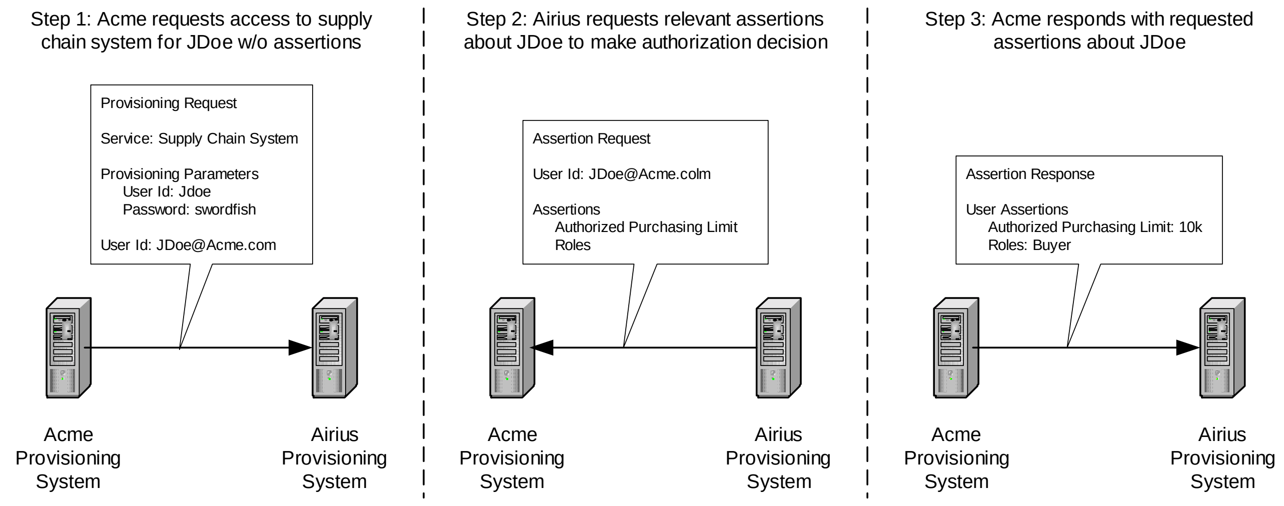
Step 1: Acme requests access to supply chain system for JDoe w/o assertions
Provisioning Request
Service: Supply Chain System
Provisioning Parameters
User Id: Jdoe
Password: swordfish
User Id: JDoe@Acme.com
Acme Provisioning System
Airius Provisioning System
Step 2: Airius requests relevant assertions about JDoe to make authorization decision
Assertion Request
User Id: JDoe@Acme.colm
Assertions
Authorized Purchasing Limit
Roles
Acme Provisioning System
Airius Provisioning System
Step 3: Acme responds with requested assertions about JDoe
Assertion Response
User Assertions
Authorized Purchasing Limit: 10k
Roles: Buyer
Acme Provisioning System
Airius Provisioning System

Figure 5: Conceptual view of pull model

The interaction solution cannot be complete without addressing the need to chain provisioning requests between more than two servers. Each of the messages described above can be forwarded from one system to another. Because not all systems may use the same user assertions to make authorization decisions or even the same set of provisioning parameters, additional provisioning parameters and user assertions may be inserted into the request as it is forwarded to other systems. For example, between Acme and Airius, the roles assertion may be used for authorization decisions. However, resource provider Apps'R'Us is the host of the resource that Airius is providing to Acme, and between Airius and Apps'R'Us a special categorization of user is defined instead of role. Airius uses the roles assertion to determine the category assertion that Apps'R'Us understands. Figure 6 below illustrates this example in a push model.

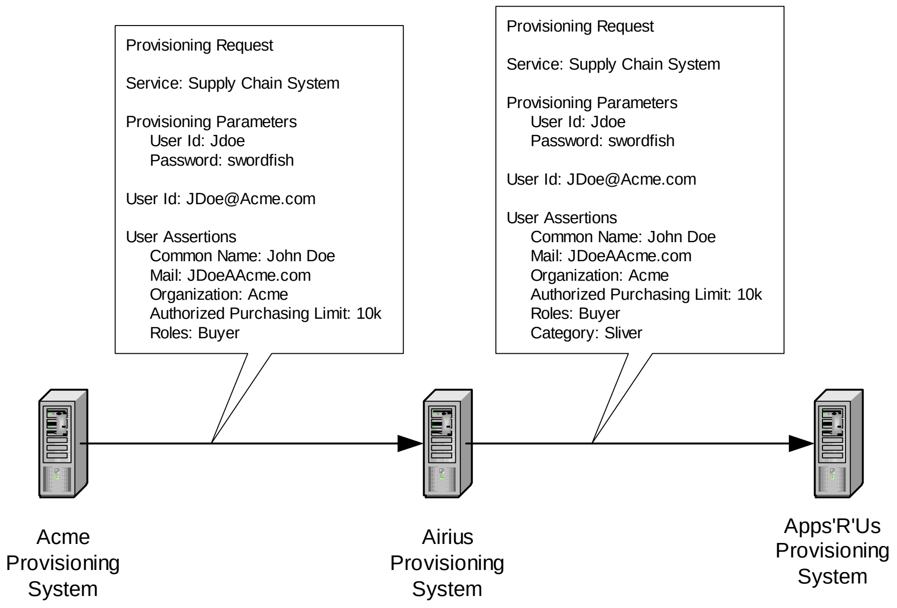

Figure 6: Chain of requests within push model

Note that the roles assertion is still provided to Apps'R'Us although it doesn't use it. This is because the initial request was just appended and forwarded. Another scenario could be that Airius issues a completely separate request to Apps'R'Us and links the two requests through its own internal processes.

The pull model can also support chaining, however the pulling of user assertions does not have to be forwarded back to the originator of the request if an intermediate system can provide the assertions itself.

In all of the situations described so far, a trust relationship has been assumed between the provisioning systems involved in the process. There is the possibility, however, of a scenario where the owner system does not have a trust relationship with the requesting system. In fact, the owner system may not wish to set up trust relationships with any requestors. In order for this model to be effective, the owner system establishes a trust relationship with a third party and requires all requestors to set up their own trust relationship with that same third party. The owner system then verifies that a provisioning request is trusted by checking the request for a certification assertion made by that third party. This certification assertion is obtained from the third party by the requesting system as the first step of the provisioning process by requesting the assertion values for the subject user from the third party itself. It is important to note that the third party is not receiving provisioning requests and forwarding them to the owner system, but merely providing a certificate user assertions for a request that the requesting system issues directly to the owner system.

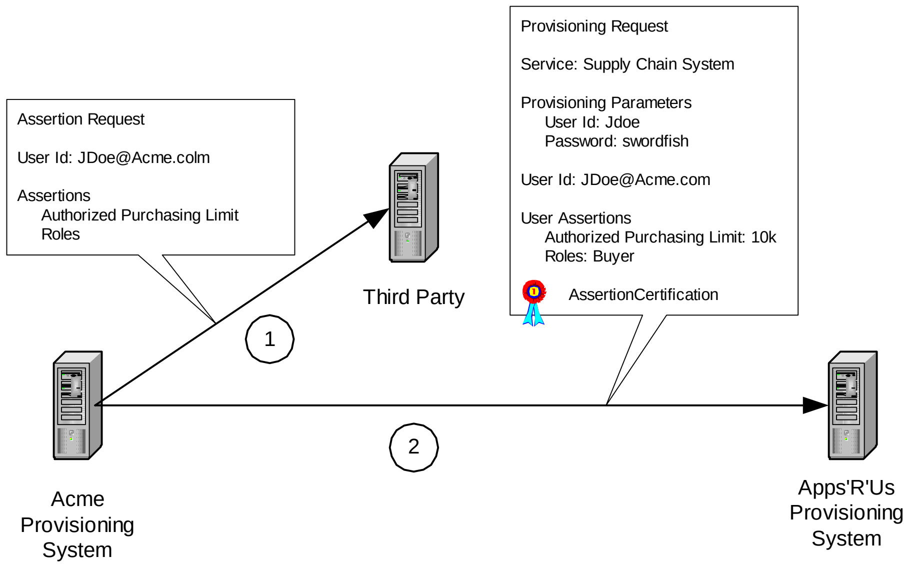

Figure 7: Using a third party for trusted user assertions