1 Novell Use Cases

1.1 Use Case: Per Tenant Identity Provider Configuration

1.1.1 Description/User Story

Multi-tenant service providers, whether they are SaaS, PaaS, or IaaS vendors, benefit from quick and easy addition of new customers – anyone with a credit card can add themselves on demand. However, to benefit from federated authentication, SSO, and other mechanisms that can improve security for their users they need to configure how their users can authenticate to the system, where and what kind of IdP they use, exchange meta-data, etc. Currently this is commonly done by the administrator via web forms that are unique to each service. As adoption of cloud services increases, this will become a significant management burden.

1.1.2 Goal or Desired Outcome

A tenant can quickly and securely manage their use of many cloud services using automated tools rather than navigating and manually configuring each service individually.

1.1.3 Notable Categorizations and Aspects

<table>
<tr><td>Categories Covered:
Infrastructure Trust Establishment
Account and Attribute Management</td><td>Applicable Deployment and Service Models:
Public
Community
Hybrid
Infrastructure-as-a-Service (IaaS)
Platform-as-a-Service (PaaS)
Software-as-a-Service (SaaS)</td></tr>
<tr><td>Actors:
Tenant Administrator
Multi-tenant Service Provider
Identity Provider</td><td>Systems:
None</td></tr>
<tr><td colspan="2">Notable Services:
Cloud Applications and Services
Cloud Identity Provider Services
Cloud Attribute Services
Identity Provider Discovery services</td></tr>
<tr><td colspan="2">Dependencies:
None</td></tr>
<tr><td colspan="2">Assumptions:
• Many... for example, wide spread adoption of federated authentication due to rapid adoption of cloud computing.</td></tr>
</table>

• The “Categories Covered” highlights the key aspects of this use case. It is assumed that all APIs and protocols used to accomplish the configuration would be follow appropriate General Identity Management, Authentication, Authorization, and Audit principles.

1.1.4 Process Flow

- A departmental manager in an enterprise (a tenant administrator) wants to configure all of the SaaS applications In use by that department to authenticate users via the enterprise IdP.
- Using an automated tool to manager her SaaS usage, she enters the IdP information once.
- The tool contacts the IdP and each SaaS application and uses standard protocols to communicate the configuration.

1.2 Use Case: Delegated Identity Provider Configuration

1.2.1 Description/User Story

Enterprises are outsourcing more of their applications and management of their IT infrastructure – including their identity provider services – to managed service providers or identity-as-a-service vendors. This results in a situation where an enterprise administrator which owns the business relationship with the service provider (the tenant administrator) does not manage the identity provider service. The identity provider service is controlled and managed by another company (the IdP administrator). This becomes a significant management burden when the tenant administrator needs to manage the identity services configuration (such as the exchange of metadata) between the identity provider and many cloud services.

1.2.2 Goal or Desired Outcome

The tenant administrator should be able to delegate access to their identity services configuration within a multi-tenant cloud service to the identity provider service. The identity provider service should be able to manage configuration issues such as meta-data exchange to all connected cloud services on behalf of a tenant. This should not require the identity provider to had access to the tenant administrator's authentication credentials.

1.2.3 Notable Categorizations and Aspects

Categories Covered:	**Applicable Deployment and Service Models:**
• Infrastructure Trust Establishment • Authentication • Authorization	• Public • Community • Hybrid • Infrastructure-as-a-Service (IaaS) • Platform-as-a-Service (PaaS)

	• Software-as-a-Service (SaaS)
Actors: • Tenant Administrator • Multi-tenant Service Provider • Identity Provider	**Systems:** • None
Notable Services: • Cloud Applications and Services • Cloud Identity Provider Services • Cloud Attribute Services • Identity Provider Discovery services	
Dependencies: • This use case depends on the Per Tenant IdP Configuration use case.	
Assumptions: • The “Categories Covered” section highlights the key aspects of this use case. It is assumed that all APIs and protocols used to accomplish the configuration would be follow appropriate General Identity Management, Account management, and Audit principles.	

1.2.4 Process Flow

A tenant administrator pulls out a credit card and signs up for a new cloud services for her users. Her identity services are provided by a third party.

She notifies the identity provider that she wants her users to have access to the new services.

The identity provider can exchange whatever configuration and meta-data is required with each new service on behalf of the tenant administrator without authenticating to each service as her.

1.3 Use Case: Association of a User and Tenant during Authentication

1.3.1 Description/User Story

NOTE: this is a rough idea of a use case. It's a situation we have seen many times, but there may not be a discrete set of viable solutions. Perhaps guidance is the best possible outcome.

When a user accesses a multi-tenant cloud service, the service needs to be able to associate the user with a tenant account. This may or may not be the same as associating the user with an IdP – there are many efforts to try to solve that issue as well and this use case may in fact be a variant of it.

Currently applications handle this issue in a variety of ways. For example, each tenant may essentially get their own application service instance by embedding the tenant identifier in the

domain name or path of the URI. Some applications pass it in as a parameter and some store it in a cookie.

This multitude of application variations further aggravates the problem of identity provider association, and makes it much more difficult to provide consistent federated identity services to multi-tenant systems.

A sample scenarios:

A manager in the sales department of AcmeWidgets wants her team to have access to a new SaaS application, WidgetTracker. She opens a new account with her department credit card. She wants her team to use their corporate user accounts for authentication – so they are provisioned, deprovisioned, and application access can be audited by the IT department. However, she's paying for the application from her departmental budget, so she only wants her team members to be able to use the service on her account. A manager in the design department has a similar need to sign up her team members for an account in WidgetTracker.

The problem is that tenant boundaries for cloud services are based on the pay-per-use model which often corresponds with departmental cost centers. In contrast, the corporate user accounts – the IdP – are usually built on corporate boundaries.

It has been suggested that this problem could be overcome by using attributes in the corporate directory to distinguish between departmental tenants. However, the explosive growth of SaaS applications is attributed in part to the low-friction, instantaneous addition of new tenants. Asking each department to submit a work order to the IT department for configuration changes in the corporate directory before they can access a new SaaS application would defeat this key characteristic.

1.3.2 Goal or Desired Outcome

That clear guidance be available from a respected authority (e.g. OASIS) to help cloud service providers provide multi-tenant capabilities in ways that can be more effectively integrated with federated identity services.

1.3.3 Notable Categorizations and Aspects

Categories Covered: • General Identity Management (IM) • Authentication • Authorization • Account and Attribute Management • Audit and Compliance	**Applicable Deployment and Service Models:** • Public • Community • Hybrid • Infrastructure-as-a-Service (IaaS) • Platform-as-a-Service (PaaS) • Software-as-a-Service (SaaS)
Actors: • Cloud Service Developers • Cloud Services	**Systems:** • None

• End User	
Notable Services: • Cloud Applications	
Dependencies: • None	
Assumptions: • None	

1.3.4 Process Flow

For developers:

- Read OASIS document
- Save time and produce better cloud service by structuring their application to associate users with tenant accounts in accordance with the document.
- Profit!

For users and administrators:

- Start to see consistency in service access
- more easily consume more services with less errors