

2012Q1 Overview of SCIM

Jan 30, 2012 - Chris Phillips – chris.phillips@canarie.ca

Technical Architect – Canadian Access Federation

SCIM Contributor

About this presentation...

- SCIM wouldn't exist if it weren't for all the contributors focusing their time and talent on the topic of provisioning.
- Check out the simplecloud.info and mailing list for all the contributors.
- SCIM has great initial momentum
 - Overtaking SPML for the preferred provisioning protocol
 - 'Heavy enough/Light enough' strikes a chord with implementers

Background

- **Intention**
 - designed to make provisioning user identity in cloud based applications and services easier
- **How**
 - to build upon experience with existing schemas and deployments
 - Intentional simplicity of development and integration
 - Based on authentication, authorization, and privacy models
- **Provides/ intended delivery of**
 - a common user schema and extension model
 - patterns for exchanging this schema using standard protocols
 - fast, cheap, and easy to move users in to, out of, and around the cloud.

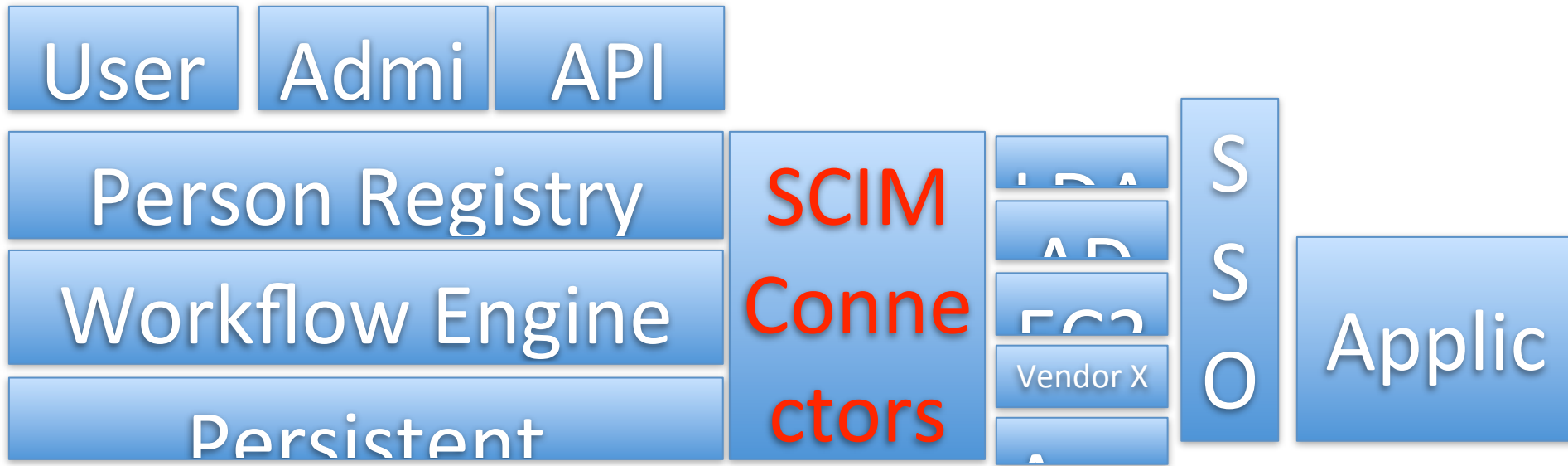
Why SCIM & Why Now?

- Stating the obvious:
Everyone provisions differently in absence of a standard → Paradox of choice
 - Too many options create confusion
 - Fragments effort and increases costs
- SCIM puts a stake in the ground
 - Enough implementers align to a single method & save \$
 - How? Consistency breeds ease of integration
 - Configure instead of custom code is the goal
 - ROI significant due to reduced complexity

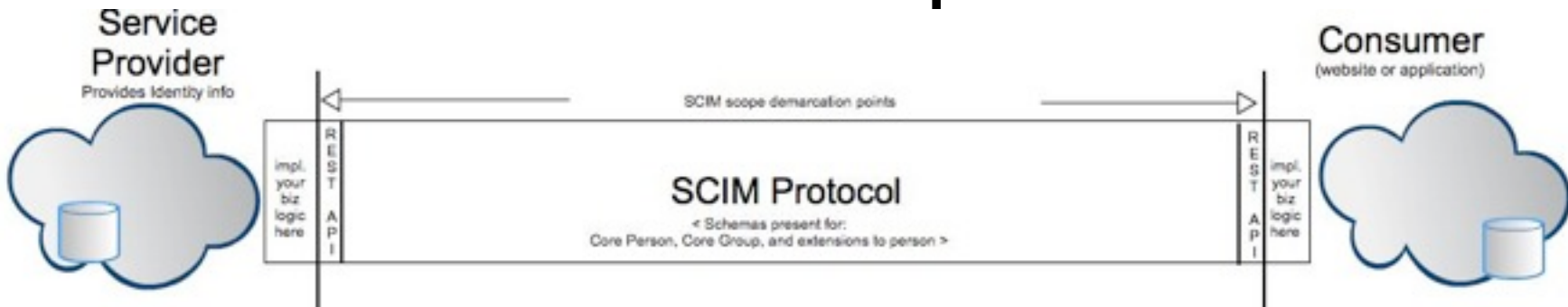
Terminology

- **Service Provider^[1]:**
 - A web application that provides identity information via the SCIM protocol.
 - **Consumer:**
 - A website or application that uses the SCIM protocol to manage identity data maintained by the Service Provider.
 - **Resource:**
 - The Service Provider managed artifact containing one or more attributes; e.g., User or Group
- [1] unfortunately this is contrary to SAML terminology in which case this may be considered

Where does SCIM play in the IDM Space?



SCIM endpoints



- GET** Retrieves a complete or partial Resource
- POST** Creates a new Resource
- PUT** Modifies a Resource with a complete, consumer specified Resource (replace)
- PATCH** Modifies a Resource with a set of consumer specified changes (partial update) or changes a User password
- DELETE** Deletes a Resource.

Resource	Endpoint	Operations	Description
User	/User	GET, POST, PUT, PATCH, DELETE	Read/Modify Users
User Query/Listing	/Users	GET	Retrieve User(s) via ad hoc queries
Group	/Group	GET, POST, PUT, PATCH, DELETE	Read/Modify Groups
User Query/Listing	/Groups	GET	Retrieve Group(s) via ad hoc queries
Change User Password	/User/{userId}/password	PATCH	Change a User's password
User Schema	/Schema	GET	Retrieve a specified User schema
User Schemas	/Schemas	GET	Retrieve all Service Provider supported schemas

Schema

- Started from portable contacts schema^[1]
 - Some pieces derived from participants needs
- Handles a variety of attribute types ^[2]:
 - Single valued, multivalued, and complex types
 - Allows for significant flexibility,
 - Implementers will have to understand how their datamodel maps to SCIM
- Philosophically Speaking, it's a core schema + extensions
 - Partitions customizations much like LDAP schema extensions

[1] <http://www.portablecontacts.net/draft-schema.html>

[2] <http://www.simplecloud.info/specs/draft-scim-core-schema-01.html>

Terminology (Con't)

- **Singular Attribute:**
 - A Resource attribute that contains 0..1 values; e.g., displayName.
- **Multi-valued Attribute:**
 - A Resource attribute that contains 0..n values; e.g., emails.
- **Simple Attribute:**
 - A Singular or Multi-valued Attribute whose value is a primitive; e.g., String.
- **Complex Attribute:**
 - A Singular or Multi-valued Attribute whose value is a composition of one or more Simple Attributes.
- **Sub-Attribute:**
 - A Simple Attribute contained within a Complex Attribute.

JSON Complex Attribute Fragment

```
{
  "name":"emails",
  "type":"complex",
  "multiValued":true,
  "multiValuedAttributeChildName":"email",
  "description":"E-mail addresses for the user. The value SHOULD be canonicalized by the Service Provider, e.g. bjensen@example.com instead of bjensen@EXAMPLE.COM. Canonical Type values of work, home, and other.",
  "schema":"urn:scim:schemas:core:1.0",
  "readOnly":false,
  "required":false,
  "caseExact":false,
  "subAttributes":[
    {
      "name":"value",
      "type":"string",
      "multiValued":false,
      "description":"E-mail addresses for the user. The value SHOULD be canonicalized by the Service Provider, e.g. bjensen@example.com instead of bjensen@EXAMPLE.COM. Canonical Type values of work, home, and other.",
      "readOnly":false,
      "required":false,
      "caseExact":false
    },
    {
      "name":"display",
      "type":"string",
      "multiValued":false,
```

Schema Mappings

- Mappings exist from SCIM to
 - LDAP inetOrgPerson, groups
 - AD person record, groups
- Still fluid are SCIM -> SAML
 - Current thinking:
 - Have 'High Fidelity' 1:1 SCIM:SAML profile
 - Have 'Lower Fidelity' SCIM to eduperson map
 - Still hot topic, but hoping that leadership from within SCIM group will have guiding hand in mapping to save time/effort for others

Usage Scenarios

- See scenarios doc [1]
- Where does SCIM play with the various techniques?
 - See Tom Zeller's lightning talk[2](Internet2) depictions of the situations/user stories:
 - Plots discussions regarding SPML, SAML, and SCIM, against LDAP

[1] <http://www.simplecloud.info/specs/draft-scim-scenarios-03.html>

[2] <https://spaces.internet2.edu/display/ACAMPIdSummit2011/Lightning+Talk+Topics+and+Slides>

License - OWF

- Licensing is OWF (Open Web Foundation)
 - Cisco, Ping Identity, Salesforce, unBoundID + others already signed on
 - CANARIE signed on as a formal way to contribute from higher ed
 - Google engaged, late ~2011Q3 and contributing

Timing

- SCIM 1.0 released Dec 15, 2011
 - Targetting IETF82(Paris) or 83(Vancouver) for BOF
- Implementations and SDKs^[1] already exist
 - unBoundID already shipping with SCIM implementation
 - Implemented as the spec evolved
 - map SCIM to inetOrgPerson in LDAP?

[1] <http://www.unboundid.com/blog/2011/07/26/the-unboundid-scim-sdk/>

Things to Think About

- Coverage is primarily on person provisioning activities and mechanics therein
 - Light coverage on groups
 - No coverage (as of yet) on privacy or other special areas
- Governance and how to 'grow the spec' to a 2.0 stage is 'light' - suggest and it will be reviewed by mailing list participants, votes on direction by OWF signatories.
 - Very lightweight so nimble, but may not be familiar to some
- Design pattern pushes complexity to extensions
 - Unclear on the good/bad design pattern
 - Encourage debate and recommendations what should be core for next round

Is Simple Really Simple?

- RESTful API calls- keeps it simple & lightweight
 - ChrisP: this is the 'SPML is too big value proposition'. It will be more simple than SPML....but hard to escape complexity of hard problems.
- Still have deal with what happens when the method is invoked on either end:
 - How well it happens here is going to make or break you (use XACML? How much intelligence? How portable?)



Parting Thoughts

- SCIM offers a compelling & consistent vision for provisioning practices.
 - Flexible & extensible
 - Your choice on fidelity/richness of schema
 - Designed to simplify interop without heavy infrastructure requirements
- Like any protocol, adoption will drive the utility & network effect
- A number of vendors are on board already, advocate to yours to enable this feature