use_cases.md

System for Cross-domain Identity Management:

Definitions, Overview, Concepts, and Requirements

Abstract

This document provides definitions,overview and selected use cases of the System for Cross-domain Identity Management (SCIM). It lays out the system's concepts, models, and flows, and it includes user scenarios, use cases, and requirements.

1. Introduction

This document provides the SCIM definitions, overview, concepts, flows, scenarios, and use cases. It also provides a list of the requirements derived from the use cases. The document's objective is to help with understanding of the design and applicability of the SCIM schema [RFC7643] and SCIM protocol [RFC7644]. Unlike the practice of some protocols like Application Bridging for Federated Access Beyond web (ABFAB) and SAML2 WebSSO, SCIM provides provisioning and de-provisioning of resources in a separate context from authentication (aka just-in-time provisioning). This document will describe the different construct that we have in the SCIM protocol and will provide the most typical use case that we will find in the implementation, will also help identify the interactions between the different constructs and guide on the roles that each has in the SCIM protocol. SCIM is a protocol where it relies on one-to-one interaction, in a client-server model. Any interaction is based on a trigger that will start a CRUD event on one or many resources.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lowercase as plain English words, absent their normative meanings. Here is a list of acronyms and abbreviations used in this document:

- **CRUD:** Create, Read, Update, Delete
- **RC:** Resource Creator
- **RU:** Resource Updater
- **RM:** Resource Manager
- **RS:** Resource Subscriber
- **RO:** Resource Object
- **RA:** Resource Attribute
- **ERC:** External Resource Creator
- **IaaS:** Infrastructure as a Service
- **JIT:** Just In Time
- **PaaS:** Platform as a Service
- **SaaS:** Software as a Service
- **IDaaS:** ID as a Service
- **IdM:** Identity Manager
- **SAML:** Security Assertion Markup Language
- **SCIM:** System for Cross-domain Identity Management
- **SSO:** Single Sign-On

2. SCIM Components and Architecture

2.1. Background and Context

The System for Cross-domain Identity Management (SCIM) specification is designed to manage resources and services in applications using standards to enable better interoperability, security, and scalability. The specification suite seeks to build upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. The intent of the SCIM specification is to reduce the cost and complexity of user management operations by providing a common user schema and extension model, as well as binding documents to provide patterns for exchanging this schema using standard protocols. In essence, make it fast, cheap, and easy to move resources in to, out of, and around the applications. The SCIM scenarios are overviews of user stories designed to help clarify the intended scope of the SCIM effort.

2.2. Implementation Concepts

2.2.1. Roles/Constructs

Constructs are the operating parties that take part in both sides of a SCIM protocol exchange and help identify the source of a given Trigger. A specific element can have one or more constructs roles, depending on the type of services that is delivering in the SCIM architecture. So far, we have identified the following SCIM constructs:

- **Resource Object (RO):** Is and object that is going to be manipulated (CRUD) by the different SCIM players, and in the end the ultimate goal to be pass across different systems and to make sure that consistent information is exchange. The Resource Object have attributes that are define by Schemas, an example of that is the SCIM Core Schema defines in [RFC 7643].
- **Resource Attributes (RA):** Is one element of the Resource Object (RO), it can have a single value or contain multiple values to describe a specific resource and its characteristics, an example of this can be the different attributes for user and/or groups under the SCIM Core Schema defined in [RFC 7643].
- **Resource Creator (RC):** Is an entity operating in a given service, is responsible of creating the Resource Object (RO) with is Resource Attributes (RA), typically we can see this role in HR or resource management applications that are responsible to create resources and be authorities for some or all its attributes.
- **Resource Updater (RU):** Is an entity that is responsible for update specific attributes (RA) of a Resource Object (RO). Typically, this role is use in conjunction with other SCIM roles that allow this SCIM entity to be authority for a specific Resource Attribute (RA)
- **Resource Manager (RM):** Is an entity that consolidated the resource Objects (RO) from the Resource Creators/Updaters (RC/RU) and make it available for the Resource Subscribers (RS), typically this entity/role is handle by the IDaaS.
- **Resource Subscriber (RS):** Is an entity that consumes Resource Objects (RO) but that is not authoritative to create them or any of its Resource Attribute (RA), normally this entity is only interested in part of the Resource Objects available in the Resource Manager (RM), typically it is an application that requires information on resources that it operate.
- **External Resource Creator (ERC):** Is an entity that has information about resources and its attributes, but that doesn’t understand SCIM, typically it is going to provide the information on the resources to the Resources Manager, using non SCIM protocols/mechanisms, an example of this would be a services that gets information about users from an LDAP server and provide it to an IDaaS using some kind of proprietary REST APIs.

```
   +-------------+ +-------------+    +-------------+ +-------------+
   |(RO) Resource| |(RA) Resource|    |(RO) Resource| |(RA) Resource|
   |   Object1   | |  Attribute1 |    |   Object2   | |  Attribute2 |
   +-------------+ +-------------+    +-------------+ +-------------+
          |               |                  |               |
   +-------------+ +-------------+    +-------------+ +-------------+
   |(RC) Resource| |(RU) Resource|    |(RC) Resource| |(RU) Resource|
   |  Creators   | |  Updaters   |    |  Creators   | |  Updaters   |
   +-------------+ +-------------+    +-------------+ +-------------+
          |               |                  |                |
          +--------+------+------------------+-------+--------+
                   |                                 |
                   v                                 v
             +----------------+               +----------------+
             | (RM) Resource  |               | (RM) Resource  |
             |    Manager     |               |    Manager     |
             +----------------+               +----------------+
                   |                                 |
             +----------------+               +----------------+
             |                |               |                |
             v                v               v                v
   +-------------+ +-------------+    +-------------+ +-------------+
   |(RS) Resource| |(RS) Resource|    |(RS) Resource| |(RS) Resource|
   |  Subscriber | |  Subscriber |    |  Subscriber | |  Subscriber |
   +-------------+ +-------------+    +-------------+ +-------------+
          |                                   |
   +-----------------+                   +----------------+
   |                 |                   |                |
   v                 v                   v                v
+-------------+ +-------------+    +-------------+ +-------------+
|(RO) Resource| |(RO) Resource|    |(RO) Resource| |(RO) Resource|
|   Object1   | |   Object2   |    |   Object1   | |   Object2   |
+-------------+ +-------------+    +-------------+ +-------------+
```

Figure 1: SCIM Roles Constructs

2.2.2. Mechanics behind Resource Object (RO) and/or Resource Attributes (RA)

Cover in the previous section it was stated that the RC/RU were authoritative over the RO/RA, that could be achieved using the mutability, concept introduced in [RFC 7644], where they would have readWrite/readOnly capabilities over them and this information would be pass to the RM. In more complex scenarios where the SCIM element doesn't has direct contact with the RC/RU that create/update a specific RO/RA, then the RM that received the original information will have the ReadWrite capabilities in the mutability field. this can be pass from RM to RM, with this mechanism we can prevent loops. When different components exist that have bi-direction connection, where they can update each other in different RA (Resource Attributes), there can only be on readWrite for a specific RA, so that we don't enter loops.

2.2.3. Triggers

Triggers are actions or activities that may cause a SCIM interaction to occur at a specific time. Triggers can occur as a result of business processes like a corporate hiring event, can be scheduled events such as a unix bash script running as a chron job, or can be just-in-time events such as SAML assertion arriving at a federated relying party that identifies a not-seen-before user. Triggers can also be standardized events, such as those in the OpenID Shared Signals Framework.
Triggers used to allow CRUD (Create, Read, Update, Delete) operations as it is designed to capture a class of use case that makes sense to the actor requesting it rather than to describe a protocol operation.

- **Instruction to Create SCIM Resource -** Service On-boarding Trigger: This is a service for the on-boarding activity in which a business action such as a new hire or new service subscription is initiated. An example of this could be the RC (Resource Creator) pushes the RO (Resource Object) to the RM (Resource Manager).
- **Notification of Creation of a SCIM Resource –** Service Notification of creation Trigger: This is a service for the on-boarding activity in which a business action such as a new hire or new service subscription is initiated. An example of this could be the RC (Resource Creator) send an event to RM (Resource Manager) notifying him that an resource has been created. This trigger can send the information of the RO (Resource Object) was created and provide its RA (Resource Attributes) or can just provide the information on the it was created and expect that the RM pull the RO/RA from the RC.
- **Instruction to Update SCIM Resource -** Service Change Trigger: An "update SCIM resource" trigger is a service change activity as a result of a resource moving or changing its service level. An example of this could be the RC (Resource Creator) or RU (Resource Updater) pushes the update of RO (Resource Object) or its RA (Resource Attributes) to the RM (Resource Manager).
- **Notification of Update SCIM Resource -** Service Notification of Change Trigger: An "update SCIM resource" trigger is a service change activity as a result of a resource moving or changing its service level. An example of this could be the RC (Resource Creator) or RU (Resource Updater) sends an event to RM (Resource Manager) notifying him that RO (Resource Object) or RA (Resource Attributes) has been updated. This trigger can send the information of the RO updated and provide its RA or can just provide the information on the it was updated and expect that the RM pull the RO/RA from the RC or RU.
- **Instruct to Delete SCIM Resource -** Service Termination Trigger: A "delete SCIM resource" trigger represents a specific and deliberate action to remove a resource from a given SCIM service point. An Example of this could be the RC (Resource Creator) or RU (Resource Updater) pushes the delete operation to the RM (Resource Manager).
- **Notification of Deletion of a SCIM Resource –** Service Notification of termination Trigger: A "delete SCIM resource" trigger represents a specific and deliberate action to remove a resource from a given SCIM service point. An example of this could be the RC (Resource Creator) or RU (Resource Updater) to send an event to the RM (Resource Manager) notifying him that a resource has been deleted. This trigger can send the information of the RO (Resource Object) was deleted.

3. SCIM Use Cases

This section we will describe the most common SCIM use cases, and will explain when, where, why and how we find them in the cross domain environment for resources managing. This list by no way tries to be exhaustive and complete, its ultimate goal is to guide developers for the possibility of such models and will try to explain their challenges and components. As mention before SCIM is a protocol for cross domains where two entities exchange information about a resource, with the use cases we try to go further and explain on how the different components can interact to allow from simple to complex architectures for cross domain resource management. Typically each use case add something on top of the previous one, starting in the most simple one, and finishing in the most complex ones, to make it easier the explanation, assume that what was describe in the previous use case applies to the use cases that come after.

3.1. Single RM/RC/RU and multiple RS

This is very common and simple SCIM use case. We have the IdM/Device Managers/etc. do all CRUD operation with the resources, then using the trigger mechanisms the resource information reach the Resource Subscribers. The RS (Resource Subscriber) will take the decision on which RA (Resource Attributes) to consider and how the Resource Object will show in their resource database. Typically we can find this kind of use case in small to mid size organization, where there is no structure method to handle the resources and typically in Organization that start with a blank sheet of paper or it is a greenfield Organization.

3.2. One or more ERC with single RM/RC/RU and multiple RS

This is the most common use case, because it allow the organization to adopt SCIM protocol for CRUD operations of their resources. In this use case the organization already have an existent database of resources that is going to be the source of truth for the Resource Manager. At no point in time the SCIM RM will provide SCIM operation with that External Resource Creator. Normally this ERC, specially if we are talking about user Identity, will have a User database that can be accessible using LDAP or can provide information of their user attributes by doing an SAML Single Sign-On using Just in time Provision. Most of the IDaaS also provide softwares that allow them to get resource information by using proprietary protocols. It is common to see HTTP REST to get the information from the ERC to the RM. Typically in this use case the RM will become the new source of truth for the resources of our Organization, will add extra Resource Attributes and ignore other RA that existed in the ERC. Some organization that already realize that going forward the RM will be the authority answer for the Resources Object and Attributes, will start create new Resource Objects in this service. The Resource Subscribers will consume all the resource information from the RM. Typically we will see this use case in small to mid size organization where resources were organized in a non standard and non open platform for Resources Management, where it isn't possible to cut/replace everything with a new system.

3.3. One or more RC/RU, with single RM/RC/RU/RS and multiple RS

In this use case, the the CRUD operation for the RO (Resource Object) and its RA (Resource Attributes) does not belong to the RM (Resource Manager), this is done in a separate SCIM entity, the Resource Creator/Resource Updater. A good example of this is use case on Users where Organization have their HR application, and the lifecycle of the resource (typically groups and Users) is done by that

application. We could also have this use case where the RM is extended with the Roles of RC/RU for extra RA (Resources Attributes) that are not authoritative by the "HR System", but normally that bring more complexity to the authority models for the CRUD operation of the resources.
Typically we will see this use case in mid to large organization where no structure method to handle the resources and they start fresh or it is a greenfield.

3.4. One or more ERC, one or more RC/RU, with single RM/RC/RU/RS and multiple RS

In this use case, the Resource information is in a ERC (External Resource Creator), and the entity that has the role of RC/RU (example given before the HR System) consumes information from the ERC. To avoid delays or loops the RM will also get original information from the ERC, just like the RC/RU. The RC/RU, either in the ERC or in the "HR application" can add extra Resource Attributes, so from a model perspective, the RM get its authoritative Information from both systems the RC/RU and from the ERC. In this model there need to be careful thoughts so that we avoid loops where specific Resource Attributes write over and over again by the ERC and RC/RU. Typically we will see this use case in mid to large organization where resources were organized in a non standard, non open platform for Resources Management and it isn't possible to cut/replace everything with a new system.

3.5. One or more ERC, one or more RC/RU, with single RM/RC/RU/RS and multiple RS/RU

In this use case we add the capability of the Resource Subscriber to be also an Resource Update, it is very common that an SaaS application can be authoritative for specific RA and add extra details to the RO. Typically we will see this use case in large organization where resources were organized in a non standard, non open platform for Resources Management and it isn't possible to cut/replace everything with a new system. Those organization start to adopt many application that brings attributes to the different resources that already exist in the system.

3.6. One or more ERC, one or more RC/RU/RS, with single RM/RC/RU/RS and multiple RS/RU

In this use case we introduce the possibility of the RC/RU (example given before the HR System) be interested in the attribute that was created updated by the RS/RU (also known as the SaaS application), an example could be adding the business email that was created by the mail service (that came from RS/RU) to the HR information service (the RC/RU/RS element) Typically we will see this use case in large organization where resources were organized in a non standard, non open platform for Resources Management and it isn't possible to cut/replace everything with a new system. hose organization start to adopt many application that brings attributes to the different resources that already exist in the system, but they need to have all the important attributes of Resources in a application in our examples "HR application"

3.7. One or more ERC, one or more RC/RU/RS, with one or more RM/RC/RU/RS and multiple RS/RU

In this use case we introduce the possibility of having multiple Resource Managers, where the information from the RO/RA is consolidated across different domains/services. As in the previous 3 uses cases we need to have careful thoughts so that we avoid loops where specific Resource Attributes write over and over again by the ERC and RC/RU, having now extra consideration for the fact that now we can have multiple Resource Managers. Typically we will see this use case in large organization, or between organization that have their own business to business communication and have the need for exchange information about Resources. Many other good example can be provided like organizations that by merging or acquisition, arrive to a situation where multiple RM exist, and their IT departments have to merge Resource information.

4. SCIM standardized Concepts

The SCIM protocol defines interactions between two standardized parties that conform to HTTP RESTful conventions. The protocol enables CRUD activities by corresponding those activities to HTTP verbs such as POST, GET, DELETE etc. The protocol itself doesn't assume a direction of data flow, and use cases discussed in section 3 can be accomplished by entities in either protocol role.

4.1 SCIM Server or Service Provider

An HTTP web application that provides identity information via the SCIM protocol. A SCIM Server is a RESTful API endpoint offering access to a data model that can be used to push or pull data between two parties. SCIM servers have additional responsibilities such as API Security, managing client identifiers & keys as well as performance management such as API throttling.

4.2 SCIM Client

A website or application that uses the SCIM protocol to manage identity data maintained by the service provider. The client initiates SCIM HTTP requests to a target service provider. A SCIM Client is active software that can call one or more SCIM servers in order to push or pull data between two parties.

4.3 Use Case mapping to RFC 7643 and 7644

The use case described before needs to be mapped to [RFC 7643] and [RFC 7644], we will bring the concepts of RO (Resource Object), RA (Resource Attribute), RC (Resource Creator), RU (Resource Updater), RM (Resource Manager) and RS (Resource Subscriber) to the concepts of SCIM Client, Server, Resource and Attribute.

4.3.1 Client active Push

Client will use HTTP PUSH to create a RO and will use HTTP PATCH/PUT to update its RA. In this section we will cover the basic constructs and will not detail the most complex use case describe before, sicne they would be just adding new elements to basic constructs describe bellow.

4.3.1.1 Resource Object creation from Client to Server

In this model we will have a Client that is going to provide information about a RO and its RA to a Server, that can also be called as Service Provider in [RFC 7643] and [RFC 7644].

```
+----------------+                                    +----------------+
|                |                 (1)                |                |
|                | --------------------------------> |                |
|                |                                    |                |
|                |                 (2)                |Service Provider|
|     Client     | <-------------------------------- |     Server     |
|   (typically   |                                    |  (typically a  |
|    an IdM)     |                 (3)                |   Application) |
|                | --------------------------------> |                |
|    RM/RC/RU    |                                    |       RS       |
|                |                 (4)                |                |
|                | <-------------------------------- |                |
+----------------+                                    +----------------+
            Figure 2: 4.3.1.1 SCIM  Flow and Entities map
```

(1) Before creating an RO or update it or its RA the SCIM client will always do an HTTP GET to get an update from the SCIM Service Provider.
(2) Service Provider will provide it RO and RA for that resource asked by the SCIM Client.
(3) Based on the RO and RA returned by the SP (Service Provider), there will be a HTTP POST, PUT, PATCH depending on the operation that the Client want to achieve.
(4) the Service Provider will return the RO and its RA with additional metadata information to allow for audit.
In the use cases that we saw before,it is related to section 3.1, where the SCIM client will map to the RM/RC/RU and the Server will map into RS.

4.3.1.2 Resource Object creation from a Creation Entity

In this model we will have a Client that is going to provide information about a RO and its RA to a Server, can also be called as Service Provider in [RFC 7643] and [RFC 7644], in this model the Client is just responsible for a limit set of attributes and do not do any management overall, and the Resource management function resides on the Server.

```
+----------------+                                    +----------------+
|                |                 (1)                |                |
|                | --------------------------------> |                |
|                |                                    |                |
|                |                 (2)                |Service Provider|
|     Client     | <-------------------------------- |    / Server    |
|   (typically   |                                    |  (typically an |
|    an HR       |                 (3)                |      IdM)      |
|   Application) | --------------------------------> |                |
|                |                                    |     RM/RS      |
|     RC/RU      |                 (4)                |                |
|                | <-------------------------------- |                |
+----------------+                                    +----------------+
            Figure 3:  4.3.1.2 SCIM  Flow and Entities map
```

(1) Before creating an RO or update it or its RA the SCIM client will always do an HTTP GET to get an update from the SCIM Service Provider.
(2) Service Provider will provide it RO and RA for that resource asked by the SCIM Client.
(3) Based on the RO and RA returned by the SP (Service Provider), there will be a HTTP POST, PUT, PATCH depending on the operation that the Client want to achieve.
(4) the Service Provider will return the RO and its RA with additional metadata information to allow for audit.
In the use cases that we saw before, it is related to part of section 3.3, where the SCIM client will map to the RC/RU and the Server will map into RM/RS, the SCIM client is also sometimes called as the "HR Application", because it responsibilities are only on be the creator and updater of the RO and specific number of its RA, the client in this case has no responsibilities in doing any management of the Resources, typically done by an IdM.

4.3.1.3 Resource Object creation from a Creation Entity and consumption from an Application

In this model we will have a Client that is going to provide information about a RO and its RA to a Server, can also be called as Service Provider in [RFC 7643] and [RFC 7644], in this model the Client is just responsible for a limit set of attributes and do not do any management overall, the Resource management function resides on the Server, that is also a client to an server that is the final recipient of the information RO and its RA.

```
+----------------+                        +---------------------------+                        +-
|                |          (1)           |                           |          (1)           |
|                | ---------------------> |                           | ---------------------> |
|                |                        |                           |                        |
|                |          (2)           |Service Provider           |          (2)           |S
|     Client     | <--------------------- |     / Server        Client| <--------------------- |
|   (typically   |                        |       (typically an       |                        |
|    an HR       |          (3)           |           IdM)            |          (3)           |
|   Application) | ---------------------> |                           | ---------------------> |
|                |                        |       RM/RS/RC/RU         |                        |
|     RC/RU      |          (4)           |                           |          (4)           |
|                | <--------------------- |                           | <--------------------- |
+----------------+                       +----------------------------+                        +-
            Figure 4:  4.3.1.3 SCIM  Flow and Entities map
```

(1) Before creating an RO or update it or its RA the SCIM client will always do an HTTP GET to get an update from the SCIM Service Provider.
(2) Service Provider will provide it RO and RA for that resource asked by the SCIM Client.
(3) Based on the RO and RA returned by the SP (Service Provider), there will be a HTTP POST, PUT, PATCH depending on the operation that the Client want to achieve.
(4) the Service Provider will return the RO and its RA with additional metadata information to allow for audit.
In the use cases that we saw before, it is related to section 3.3, where the SCIM client on the left will map to the RC/RU and the Server in the middle will map into RM/RS, the SCIM client is also sometimes called as the "HR Application", because it responsibilities are only on be the creator and updater of the RO and specific number of its RA, the client in this case has no responsibilities in doing any management of the Resources, typically done by an IdM.
The center component as describe is the Server for the client on the left and will act as the Client for the server on the right. Typically the Server on the right is an application that wan tto consume RO and its RA.

4.3.1.4 Resource Object creation from a Creation Entity and consumption from an Application when different Resource Attributes are generated in different entities

In this model we will have a Client that is going to provide information about a RO and its RA to a Server, can also be called as Service Provider in [RFC 7643] and [RFC 7644], in this model the Client is just responsible for a limit set of attributes and do not do any management overall, the Resource management function resides on the Server, that is also a client to an server that is the final recipient of the information RO and its RA.

```
+----------------+                        +---------------------------+                        +-
|                | ---------(1)---------> |                           | ---------(1)---------> |
|        Client  | <--------(2)---------- |Service Provider     Client| <--------(2)---------- |S
|                | ---------(3)---------> |    / Server               | ---------(3)---------> |
|   (typically   | <--------(4)---------- |                           | <--------(4)---------- |
|      an HR     |                        |       (typically an       |                        |
|   Application) |                        |           IdM)            |                        |
|     RC/RU/RS   | <--------(1)---------- |       RM/RS/RC/RU         | <--------(1)---------- |
|                | ---------(2)---------> |                           | ---------(2)---------> |
|Service Provider| <--------(3)---------- |Client     Service Provider| <--------(3)---------- |
|       / Server | ---------(4)---------> |                  / Server | ---------(4)---------> |
+----------------+                        +---------------------------+                        +-
            Figure 4:  4.3.1.4 SCIM  Flow and Entities map
```

(1) Before creating an RO or update it or its RA the SCIM client will always do an HTTP GET to get an update from the SCIM Service Provider.
(2) Service Provider will provide it RO and RA for that resource asked by the SCIM Client.
(3) Based on the RO and RA returned by the SP (Service Provider), there will be a HTTP POST, PUT, PATCH depending on the operation that the Client want to achieve.
(4) the Service Provider will return the RO and its RA with additional metadata information to allow for audit.
In the use cases that we saw before, it is related to section 3.6, where the SCIM client on the top left will map to the RC/RU and the Server in the middle left will map into RM/RS, the SCIM client is also sometimes called as the "HR Application", because it responsibilities are only on be the creator and updater of the RO and specific number of its RA, the client in this case has no responsibilities in doing any management of the Resources, typically done by an IdM.
The center component as describe is the Server for the client on the left and will act as the Client for the server on the right. Typically the Server on the right is an application that wan tto consume RO and its RA.
In addition to the models before now the "HR Application also subscribe to RA that are created by the RS and reported by the RM, the Application will be the creator of specific attributes.

4.3.2 Client Active Pull

In a client active pull scenario, the primary flow of data moves from the SCIM Server in the role of RU to one or many SCIM clients acting primarily in the RS role. Clients chose when and how often to make HTTP GET calls to the server, based on the size of the object population the client is tracking, the frequency of the data change, and the use case, for example the synchronization of a registry of objects vs. point updates when an event takes place. These factors may result in clients periodically polling a large set of SCIM Server objects to check for changes. The Client active pull can be the best implementation choice when working with resource subscribers that are unable to deploy a SCIM server. Examples of cases where the client active pull is used include situations where a client needs to maintain a synchronized large body of objects, such as a device list or user address book.

```
+----------------+                         +--------------------------+
|                | <---------(1)---------- |                          |
|      SCIM      | <---------(2)---------- |       SCIM Client        |
|     Server     | <---------(3)---------- |     HTTP GET /users      |
|                | <---------(4)---------- |        every hour        |
|     (eg an     |                         +--------------------------+
|  address book) |
|                |                         +--------------------------+
|      RC/RU     |                         |       SCIM Client        |
|                | <---------(1)---------- | 1)HTTP GET on SSO trigger |
|                | <---------(2)---------- | 2)HTTP GET on SSO trigger |
+----------------+                         +--------------------------+
            Figure 4.3.2:  Client Active Pull
```

4.3.2.1 Resource Object creation from Server to Client

In this scenario, creation of a new resource object at the SCIM server would result in a new object available via a SCIM call from the client. The client may discover the newly created object by a GET on the resource endpoint (eg /users), thereby discovering the new object, or some kind of trigger event might occur that prompts the SCIM client to perform an explicit HTTP GET (either through a specific query or through communication of the object's identifier).

4.3.2.2 Resource Object Server consumption from Client

Consumption of objects occur via API calls to the SCIM Server. The SCIM client could choose to poll the SCIM Server regularly in order to consume information or the SCIM client could be triggered to make a more specific call. One example of a triggered SCIM Client call could be an SSO trigger, where a user has just performed a federated login to the client domain, and the client is looking for updated information about that user.

4.3.2.2 Resource Object creation or update from Client to Server

When a client needs to update an object at the SCIM server, the client sends an HTTP POST to the SCIM Server. This could happen for example when a given client "owns" a specific device record that is part of a central device repository at the SCIM server.

4.3.3 Client active Pull and Push

5. Security Considerations

Authentication and authorization must be guaranteed for the SCIM operations to ensure that only authenticated entities can perform the SCIM requests and the requested SCIM operations are authorized. SCIM resources (e.g., Users and Groups) can contain sensitive information. Thus, data confidentiality MUST be guaranteed at the transport layer. There can be privacy issues that go beyond transport security, e.g., moving personally identifying information (PII) offshore between different SCIM elements. Regulatory requirements shall be met when migrating identity information between jurisdictional regions (e.g., countries and states may have differing regulations on privacy). Additionally, privacy-sensitive data elements may be omitted or obscured in SCIM transactions or stored records to protect these data elements for a user. For instance, a role-based identifier might be used in place of an individual's name. Detailed security considerations are specified in Section 7 of the SCIM protocol [RFC7644] and Section 9 of the SCIM schema [RFC7643].

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, http://www.rfc-editor.org/info/rfc2119.

6.2. Informative References

[RFC7643] Hunt, P., Ed., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Core Schema", RFC 7643, DOI 10.17487/RFC7643, September 2015, http://www.rfc-editor.org/info/rfc7643.

[RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, http://www.rfc-editor.org/info/rfc7644.

Acknowledgments

Authors' Addresses

Paulo Jorge Correia
Cisco Systems
Lagoas Park, Edificio 12 2740-269 Porto Salvo
Portugal
Email: paucorre@cisco.com

Pamela Dingle
Microsoft Corporation

Email: pamela.dingle@microsoft.com