SCIM
Internet-Draft
Intended status: Standards Track
Expires: 22 January 2026

P. J. Correia
Cisco Systems
P. Dingle
Microsoft Corporation

21 July 2025

System for Cross-domain Identity Management: Definitions, Overview, Concepts, and Requirements
draft-ietf-scim-use-cases-reloaded-01

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 use cases, and implementation considerations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 January 2026.

Table of Contents

1. Introduction

The System for Cross-domain Identity Management (SCIM) family of specifications [RFC7643] and [RFC7644] is designed to manage resources used in the practice of identity management that need to be communicated across internet domains and services, with users and groups as the default resources supported (and an extensibility model for additional resource definitions). The specifications have two primary goals: 1. A common representation of a resource object and its attributes. 2. Standardized patterns for how those resources can be operated on, including "CRUD" operations (Create, Read, Update, Delete) for resource objects and more advanced goals such as search filters, synchronization of large resource populations, etc. These goals are codified as a data model in [RFC7643], which defines resources, attributes, and default schemas, as well as a protocol definition built on HTTP in [RFC7644]. By standardizing the data

model and protocol for resource management, entire ecosystems can achieve better interoperability, security, and scalability.

This document provides definitions, overviews, concepts, flows, and use cases that implementers may need to understand the design and applicability of the SCIM schema [RFC7643] and SCIM protocol [RFC7644]. Unlike some protocols like Application Bridging for Federated Access Beyond Web (ABFAB) [RFC7832] and SAML2 WebSSO, SCIM provides provisioning and de-provisioning of resources in a separate context from authentication. While SCIM is a protocol that standardizes the movement of data only between two parties in an HTTP client-server model, this document discusses implementation patterns that use concepts beyond the core schema and protocol, which are necessary to understand how SCIM actions can fit into larger architectures.

2. 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 * ERC: External Resource Creator * IaaS: Infrastructure as a Service * IDaaS: Identity as a Service * IdM: Identity Manager * JIT: Just In Time * RC: Resource Creator * RU: Resource Updater * RM: Resource Manager * RS: Resource Subscriber * SRO: SCIM Resource Object * SROA: SCIM Resource Object Attribute * SaaS: Software as a Service * SAML: Security Assertion Markup Language * SCIM: System for Cross-domain Identity Management * SET: Security Event Token * SSO: Single Sign-On

3. SCIM Components and Architecture

The SCIM architecture is a client-server model centered on a normative concept of a "resource." Resources have types (such as a user or a group), and each unique instance of a resource type is represented by a JSON object, accessed via a standardized REST API. Each resource object can be managed individually or in bulk using actions that by default are specified in RFC9110 (HTTP GET, PUT, POST, etc.), but may also expand to concepts in extension documents, such as security event tokens (SETs). This model enables organizations to represent information about user populations and the groups those user populations are part of using the core specifications, and to extend to other important resources using extension drafts in the same family, with the high-level concept of performing SCIM actions on resource objects. SCIM actions result in

resource objects and associated data "moving" between the client and server, as clients actively push and pull information that reflects changes over time. This communication of data enables systems within domains and across domains to operate on the freshest possible version of object state.

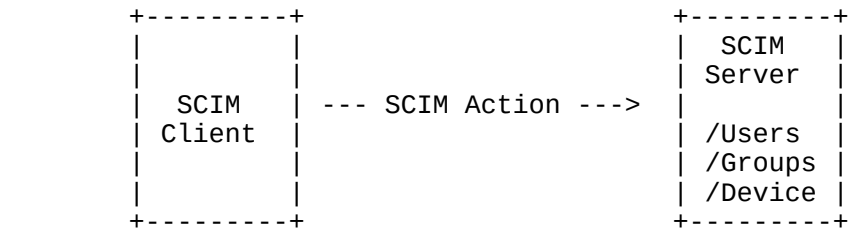

Figure 1: SCIM Components

The intent of the SCIM specification is to reduce the cost and complexity of resource management operations by providing common schemas and an extension model, as well as binding documents to offer patterns for exchanging this schema using standard protocols. In essence, it aims to make it fast, cheap, and easy to move resources into, out of, and around applications. The SCIM scenarios are overviews of user stories designed to help clarify the intended scope of the SCIM effort.

3.1. Implementation Concepts

To understand the use cases, we need to familiarize ourselves with five different concepts of the SCIM protocol: Data Models, Protocol Roles, Orchestrator Roles, Triggers, and Actions.

3.1.1. Data Models

SCIM defines two types of data entities: Resources and Attributes.

3.1.1.1. SCIM Resource Object (SRO)

A JSON object representing a user, group (or extension object like devices) used by the CRUD operations through the SCIM protocol. The Resource Object contains attributes defined by schemas such as those defined in [RFC7643] and can be implemented via the endpoints and parameters defined in [RFC7644]. Others SCIM Resource Object (SRO) maybe defined by IETF and register in IANA under SCIM Schema URIs for Data Resources, there is alo the possibility of using the SCIM protocol with private SCIM Resource Object (SRO) that will not even be register in IANA.

3.1.1.2. SCIM Resource Object Attribute (SROA)

A named element of a SCIM Resource Object (SRO). Attributes are defined in section 2 of [RFC7643] and include characteristics like cardinality (single or multiple values), data types (string, boolean, binary, etc.), and characteristics (required, unique, etc.).

3.1.2. Protocol Roles

SCIM is based on the HTTP protocol; HTTP client and server roles are defined in [RFC9110] and [RFC9112]. Any SCIM interaction requires one participant to be a SCIM server and the other to be a SCIM client.

3.1.2.1. SCIM Server (also known as a SCIM 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 and keys, as well as performance management such as API throttling.

3.1.2.2. SCIM Client

A website or application that uses the SCIM protocol to manage identity data maintained by the service provider. The client can initiate SCIM HTTP requests to a target SCIM Server. A SCIM Client is active software that can push or pull data between two parties.

3.1.3. Orchestrator Roles

Orchestrators are the operating parties that take part in a SCIM protocol exchange and ensure data is moving in the correct flows. An entity can have one or more orchestrator roles, depending on the overall architecture.

3.1.3.1. Resource Creator (RC)

An entity responsible for creating the SCIM Resource Object (SRO). Typically, this role is found in HR or Resource Management (RM) applications that are responsible for creating resources and their attributes.

3.1.3.2. Resource Updater (RU)

An entity responsible for updating specific SCIM Resource Object Attribute (SROA) of a SCIM Resource Object (SRO) or the RO itself. Typically, this role is used in conjunction with other SCIM roles that allow this SCIM entity to manage specific SCIM Resource Object Attribute (SROA) and/or SCIM Resource Object (SRO).

3.1.3.3. Resource Manager (RM)

An entity that aggregates or transforms SCIM Resource Object (SRO) from resource creators/updaters (RC/RU) and makes them available for Resource Subscribers (RS) using multiple SCIM interactions. An example of this role could be an Identity-as-a-Service (IDaaS) cloud service.

3.1.3.4. Resource Subscriber (RS)

An entity that consumes SCIM Resource Object (SRO) and typically doesn’t create new Objects or Attributes. An example would be a SaaS application that delivers a service and needs to create a database of Objects and would get those from an RM/RC/RU.

3.1.3.5. External Resource Creator (ERC)

An entity that has information about SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) but does not participate in SCIM flows. Examples include databases or internally-facing applications.

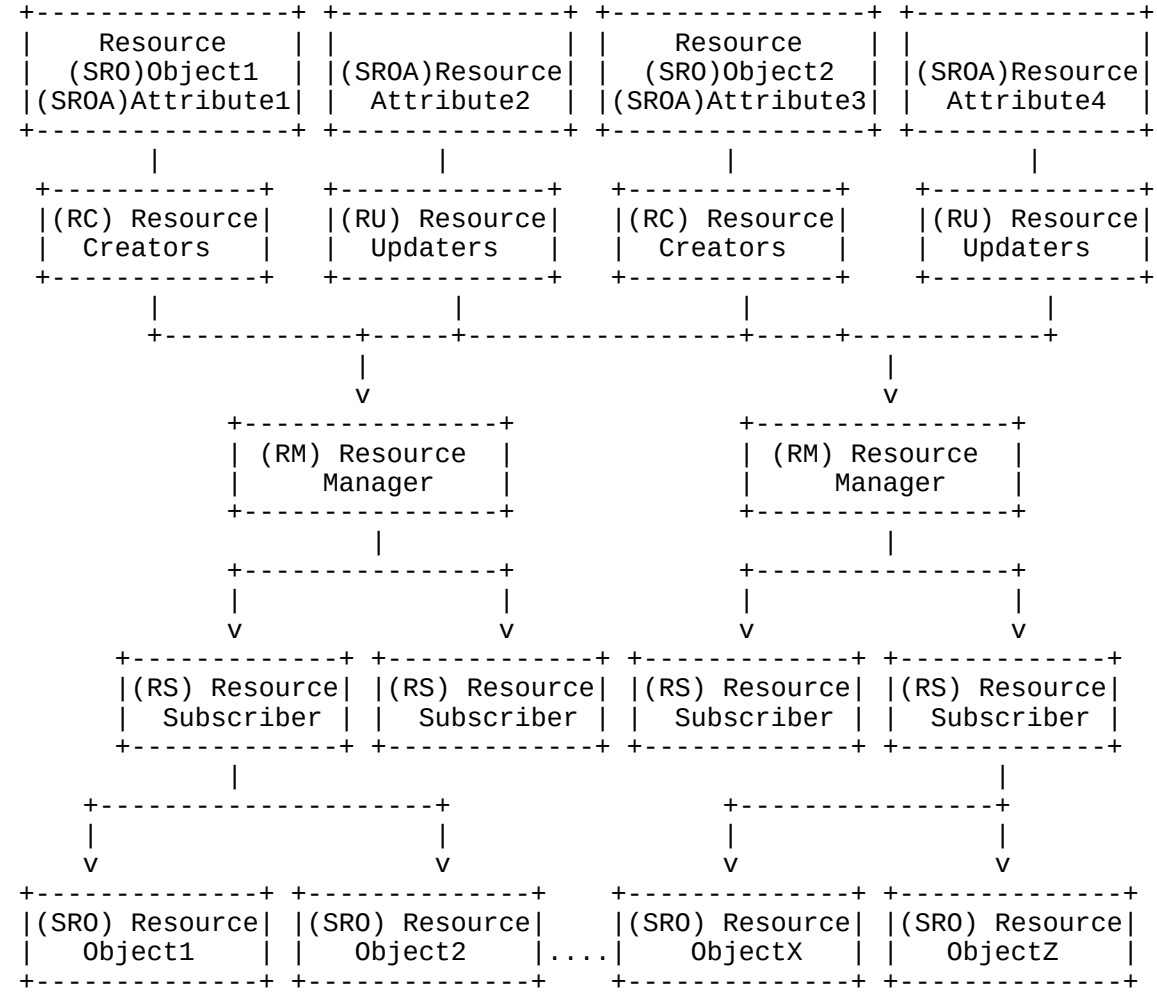

Figure 2: SCIM Orchestrators Roles

3.1.4. Triggers

Triggers are activities that may cause a SCIM action to occur. Triggers can result from business processes like a corporate hiring event, scheduled events such as a Unix bash script running as a cron job, or SSO just-in-time events arriving at a federated relying party that identifies a previously unseen user. Triggers can also be standardized events, such as those in the OpenID Shared Signals Framework. Triggers are used to initiate CRUD (Create, Read, Update, Delete) operations using SCIM Actions. The use cases described in this document can use one or multiple trigger mechanisms to achieve the goal of the SCIM element.

3.1.4.1. Periodic Intervals

A periodic interval trigger is a pre-configured agreement where a SCIM client or server performs an action at a specific time. This trigger is often recurring and typically initiates an action from the SCIM Client, though in some use cases it can be done by the SCIM Server. An example of a periodic interval trigger could be a UNIX cron job calling a script.

3.1.4.2. Events

Event triggers are activities, contexts, or notifications that could happen at any time. A SCIM client may be configured to perform a given SCIM action in response to a specific event, such as an entry written into an audit log, a signal of a corporate workflow completion, or a device management platform notification. SCIM actions could also be triggered by a Security Event Token (SET) as described in [RFC8417] or a SCIM event corresponding to [SCIM_Profile_for_Security_Event_Tokens].

3.1.4.3. Application Triggers

Application triggers occur when administrative or end-user interfaces are manipulated. An example of an application trigger might be a user modifying their profile information, resulting in a SCIM client performing an HTTP POST to update the user’s resource object at the SCIM server. Another example might be an Identity Administrator creating a new User in the IdM, who immediately wants to update one or more resource Subscribers (typically a SaaS application that is a SCIM Server).

3.1.4.4. SSO (Single Sign-On)

Single Sign-On triggers occur when a user authenticates via federated protocols such as SAML 2.0 or OpenID Connect. If a federated assertion arrives for a user who has not yet been provisioned into the destination application, the application may be triggered to perform just-in-time (JIT) provisioning. This trigger occurs in scenarios where a Single Sign-On flow happens, but not all the resource attributes for the user object are passed in the federated assertion, resulting in a SCIM action to push or pull the remaining needed attributes.

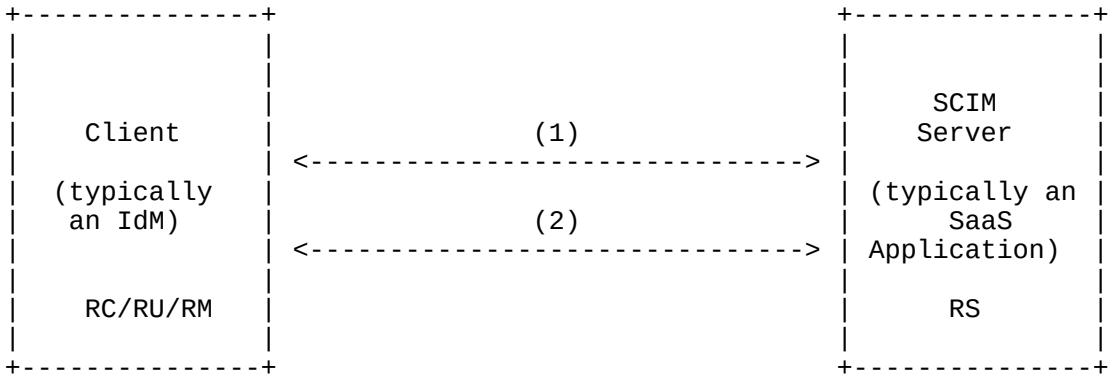

Figure 3: SCIM trigger using Single Sign-On

1. An SSO trigger creates the user and might create some SCIM Resource Object Attribute (SROA) of a SCIM Resource Object (SRO).

2. SCIM actions will then complement the attributes created initially through SSO JIT with additional SCIM Resource Object Attribute (SROA) of the previously created SCIM Resource Object (SRO). This use case combines the SCIM protocol with other protocols used for Single Sign-On, especially in the context of JIT (Just-in-Time Provisioning). This is particularly useful with protocols like SAML, which are limited by the number of characters in the URL.

3.1.5. SCIM Actions

The SCIM protocol defines interactions between two standardized parties that conform to HTTP RESTful conventions. The protocol enables CRUD operations by mapping these activities to HTTP verbs such as POST, PUT, GET, DELETE, etc. The protocol itself doesn’t assume a direction of data flow, and use cases discussed in section 4 are created using the orchestrator roles. A SCIM entity can have multiple roles depending on the objective of the use case being described.

3.1.5.1. Client active Push

A SCIM client uses HTTP verbs POST, PUT, or PATCH to create or update objects and/or attributes at a SCIM server. The SCIM client is actively "pushing" the data to the endpoint. This SCIM action can occur when the SCIM client is the primary Resource Creator/Updater (RC/RU). The most common and widely deployed example is a SCIM client providing information about a RO and its RA to a server, which is also called a SCIM Server in [RFC7643] and [RFC7644].

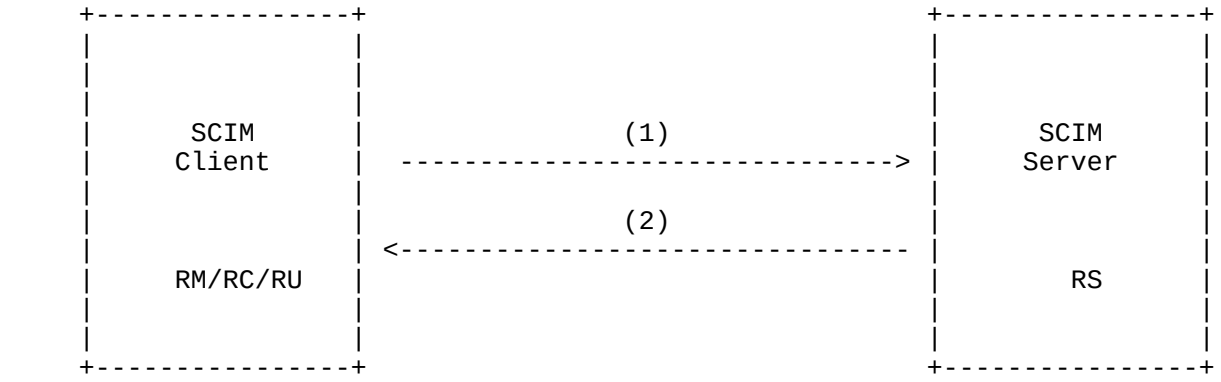

Figure 4: SCIM action for Client Active Push

1. There will be push using a HTTP POST, PUT, PATCH, DELETE depending on the operation that the Client want to achieve at the Server.

2. The Service Provider will return the RO/RA with additional metadata information to allow for audit.

3.1.5.2. Client Active Pull

A SCIM client uses the HTTP GET verb to request data from a SCIM server. With the action of an active pull, the client will fetch one or multiple objects from the SCIM server. Client active pulls can be used in situations where a client needs to maintain a synchronized large body of objects, such as a device list or user address book, without the need to track individual SCIM Resource Object (SRO) or SCIM Resource Object Attribute (SROA). There are also cases where the client performs a one-time pull of only one specific RO from a server that manages many ROs. For example, a mobile app (SCIM Client) may fetch the current license entitlement from a Device Manager (SCIM Server).

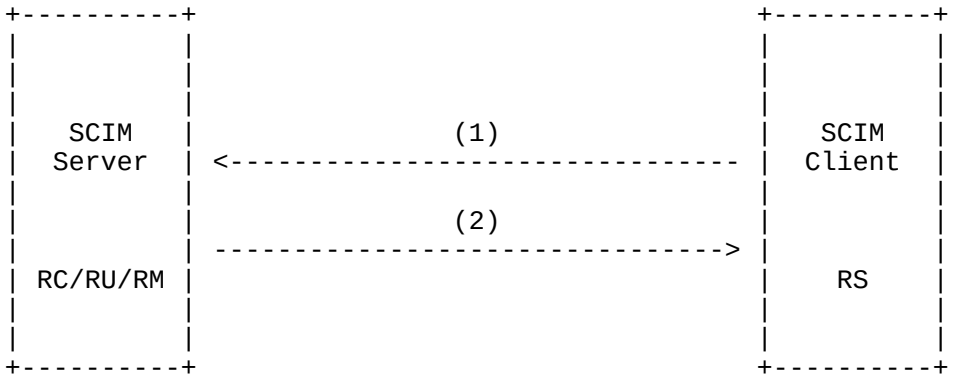

Figure 5: SCIM action for Client Active Pull

1. The SCIM client will perform an HTTP GET to obtain the selected list of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA).

2. The SCIM Server will return the RO and its RA along with additional metadata information to allow for auditing.

3.1.5.3. Active Dynamic Query

A SCIM client uses the HTTP GET verb to request data from a SCIM server. With the action of an active pull, the client will fetch one or multiple objects from the SCIM server. The response data from the SCIM server will include a Dynamic Query (DQ) token that allows the client to subsequent active pulls that will only return RO objects that have changed (including references to deleted objects). The data returned from a dynamic query is usually much smaller, and allows a client to focus only on processing incremental changes rather than performing a full sync every time. With this kind of action, SCIM reconciliations are possible, where the SCIM client can resolve inconsistencies created over time between the client and the SCIM server. This SCIM actions is not cover by an RFC yet, and will need to bedetailed in a RFC.

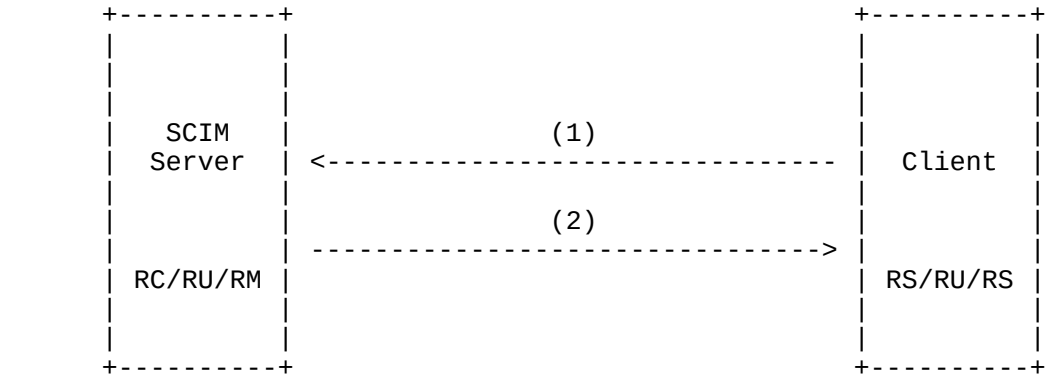

Figure 6: SCIM action for Client Active Dynamic Query

1. The SCIM client will perform an HTTP GET requesting a delta list of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) since the previous SCIM action.

2. The SCIM Service Provider will return the delta list of RO and their RA along with additional metadata information for auditing purposes.

3.1.5.4. Domain Replication Mode

This is an action specifically for triggers that are events. In this mode, there is an administrative relationship spanning multiple operational domains. Data shared in events typically uses the full mode variation of change events, including the data payload attribute. This eliminates the need for a callback to retrieve additional data. "Domain-Based Replication" events (DBR) are used to synchronize resource changes between SCIM service providers within a common administrative domain.

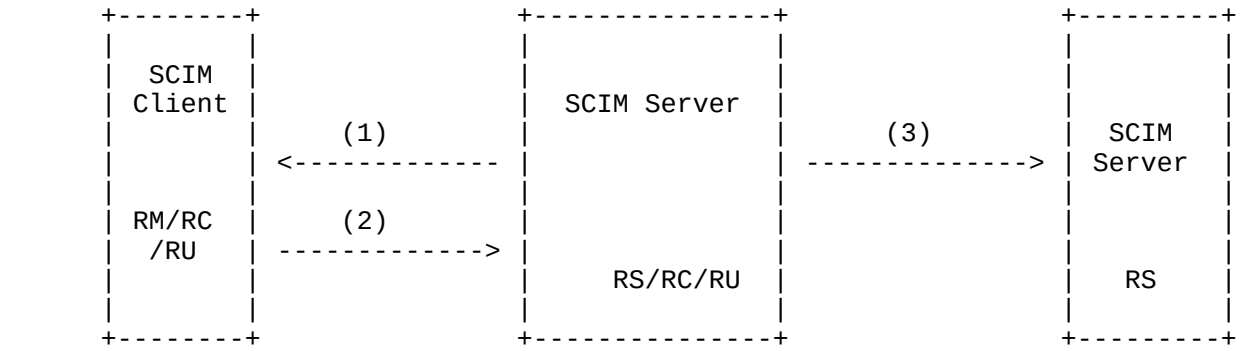

Figure 7: SCIM actions aggregated by a SCIM server then transmitted via SCIM Events using Domain Replication Mode

1. SCIM Action.

2. SCIM Response.

3. Event SCIM:prov:op id:xyz

3.1.5.5. Co-Ordinated Provisioning

In these relationships, an Event Publisher and Receiver [SCIM_Profile_for_Security_Event_Tokens] typically exchange resource change events without exchanging data. For the receiver to know the value of the data, the Event Receiver usually makes calls back to the SCIM Event Publisher domain to receive a new copy of the data (e.g., using a SCIM GET request). In any Event Publisher and Receiver relationship, the set of SCIM resources (e.g., users) that are linked or coordinated is managed within the context of an event feed, which MAY be a subset of the total set of resources on either side. For example, an event feed could be limited to users who have consented to the sharing of information between domains. To support this capability, "feed" specific events are defined to indicate the addition and removal of SCIM resources from a feed.

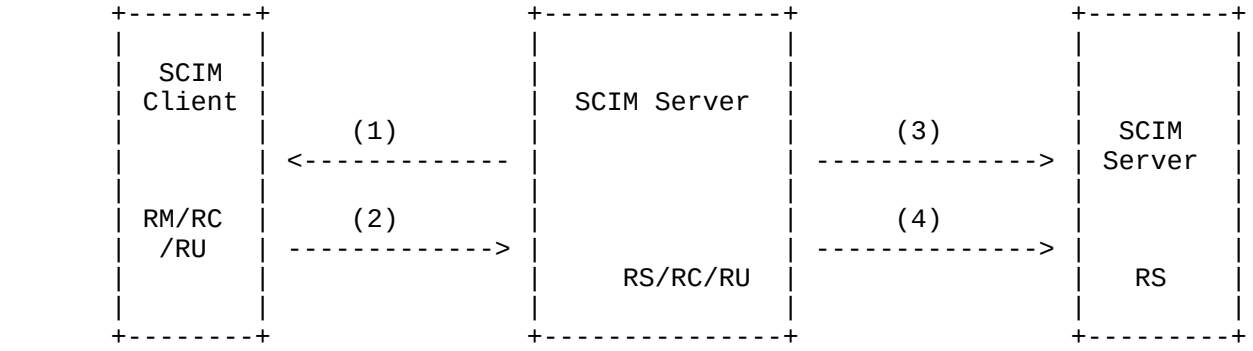

Figure 8: SCIM actions aggregated by a SCIM server then transmitted via SCIM Events using Co-Ordinated Provisioning

1. SCIM Action.

2. SCIM Response.

3. Event SCIM:prov:op id:xyz

4. SCIM Active Pull

4. SCIM Use Cases

This section describes some common SCIM use cases, explaining when, where, why, and how they are found in cross-domain environments. The ultimate goal is to provide guidance for developers working on common models, explaining the challenges and components involved. Because SCIM is a protocol where two entities exchange information about resources across domains, the use cases explain how the different components can interact to support simple to complex architectures for cross-domain resource management. Orchestrator roles are mapped to the use cases to simplify the explanation of the multiple functions of the SCIM elements. The use cases build on each other, starting with simple cases and ending with the most complex ones.

4.1. Use Cases for Orchestrator Roles

4.1.1. Resource Subscriber (RS)

A Resource Subscriber (RS) receives data from a remote corporate data store. This is a very common and simple SCIM use case, where the SCIM Resource Object (SRO) and its SCIM Resource Object Attribute (SROA) are created by another party. The CRUD operations on these resources trigger specific actions to facilitate the information exchange between two entities, typically the SCIM Client and Server. The Resource Subscriber (RS) will decide which SCIM Resource Object Attribute (SROA) to consider and how the SCIM Resource Object (SRO) will appear in its resource database. Typically, we find this kind of use case in small to mid-sized organizations, and it is usually seen in on-premises deployments.

4.1.1.1. Single-Tenant Resource Subscriber (RS)

Resource Subscriber (RS) in a single tenant that can either be the SCIM Client or SCIM Server. Typically, we see this in an on-premise application.

4.1.1.1.1. Single-Tenant Resource Subscriber that is the SCIM Server

It is common today for the SCIM Client, typically performing the roles of RM (Resource Manager), RC (Resource Creator), and RU (Resource Updater), to perform CRUD operations on the database of the RS (Resource Subscriber) using the Active Push method. This action delivers SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) to the single-tenant RS. A good example would be an on-premises application (most commonly a single-tenant application) that creates its own database of objects for its own use, obtaining the objects from a central IdM (Identity Management) system.

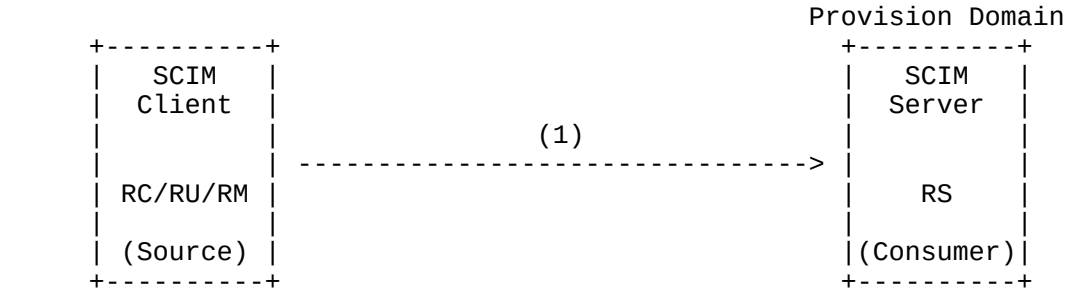

Figure 9: Single-Tenant Resource Subscriber that is the SCIM Server

1. SCIM action - SCIM Client performs Active Push

4.1.1.1.2. Single-Tenant Resource Subscriber that is the SCIM Client

The SCIM Client, which is the RS (Resource Subscriber), will perform CRUD operations on its own database using the Active and/or Delta Pull methods. Source information is available in the SCIM server, which is the IdM (Identity Management) system and is responsible for the roles of RM (Resource Manager), RC (Resource Creator), and RU (Resource Updater) for the SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). A good example would be an on-premises application (most commonly a single-tenant application) that creates its own database of objects, such as devices, from a central IdM (Identity Management) system. This option is a good solution for situations where the RS (Resource Subscriber) is not reachable from the IdM.

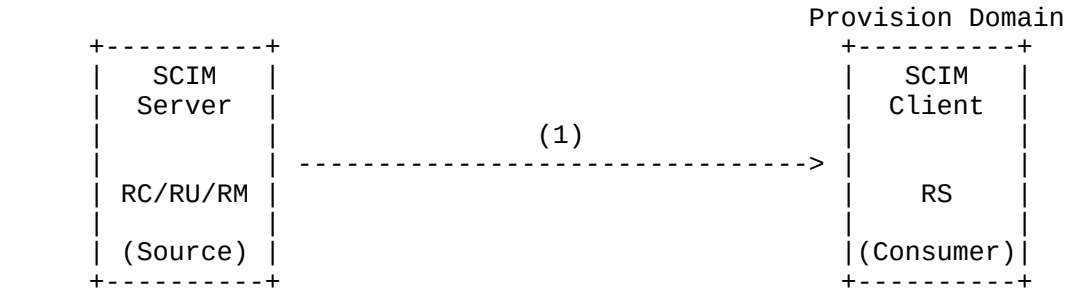

Figure 10: Single-Tenant Resource Subscriber that is the SCIM Client

1. SCIM action - SCIM Client performs Active/Delta Pull

4.1.1.2. Multi-Tenant Resource Subscriber

It only differs from the Single-Tenant Resource Subscriber (RS) by supporting multiple tenants. Typically, we see this in SaaS applications.

4.1.1.2.1. Multi-Tenant Resource Subscriber that is the SCIM Server

It is the most common today for the SCIM Client, typically performing the roles of RM (Resource Manager), RC (Resource Creator), and RU (Resource Updater), to perform CRUD operations on the database of the RS (Resource Subscriber) using the Active Push method. This action delivers SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) to the multi-tenant RS (Resrouce Subscriber). A good example would be a SaaS application (most commonly a multi-tenant applications) that creates its own database of objects for its own use, obtaining the objects from a central IdM (Identity Management) system.

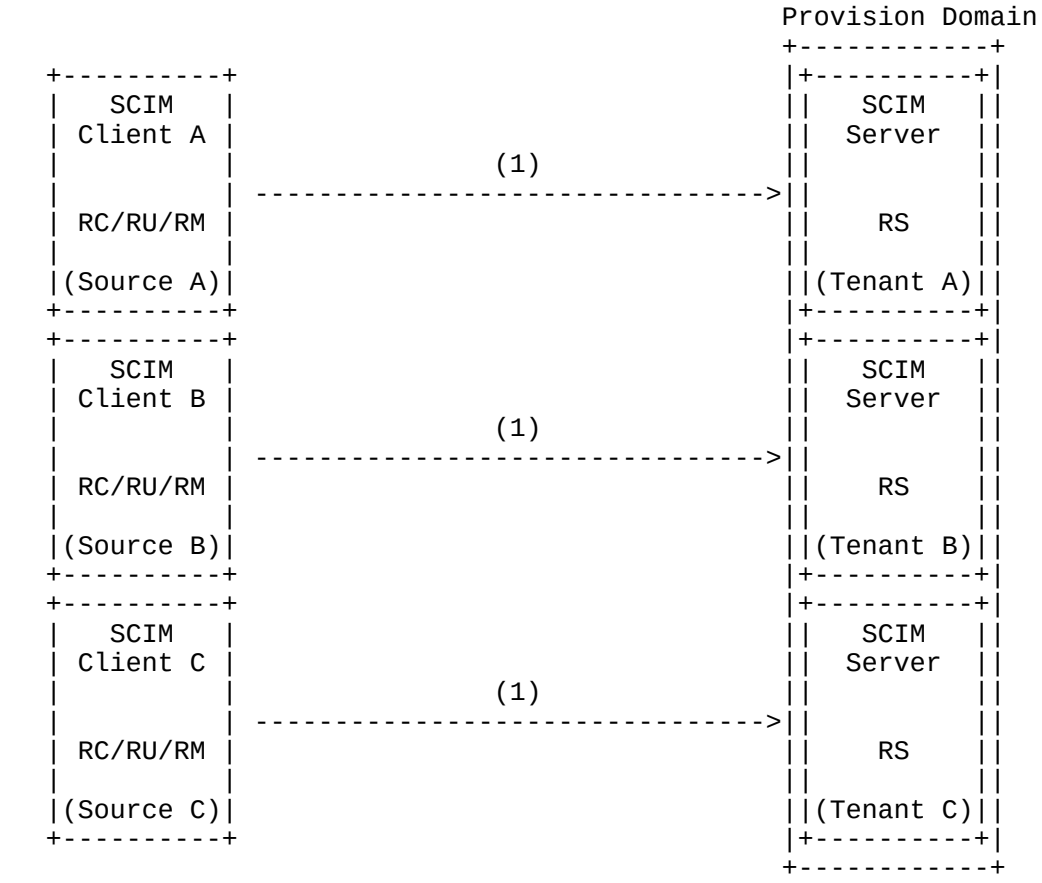

Figure 11: Multi-Tenant Resource Subscriber that is the SCIM Server

1. SCIM action - SCIM Client performs Active Push

4.1.1.2.2. Multi-Tenant Resource Subscriber that is the SCIM Client

The SCIM Client, which is the RS (Resource Subscriber), will perform CRUD operations on its own database using the Active and/or Delta Pull methods. Source information is available in the SCIM server, which is the IdM (Identity Management) system and is responsible for the roles of RM (Resource Manager), RC (Resource Creator), and RU (Resource Updater) for the SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). A good example would be a SaaS application (most commonly a multi-tenant application) that creates its own database of objects for each of its tenants, using a central IdM (Identity Management) system.

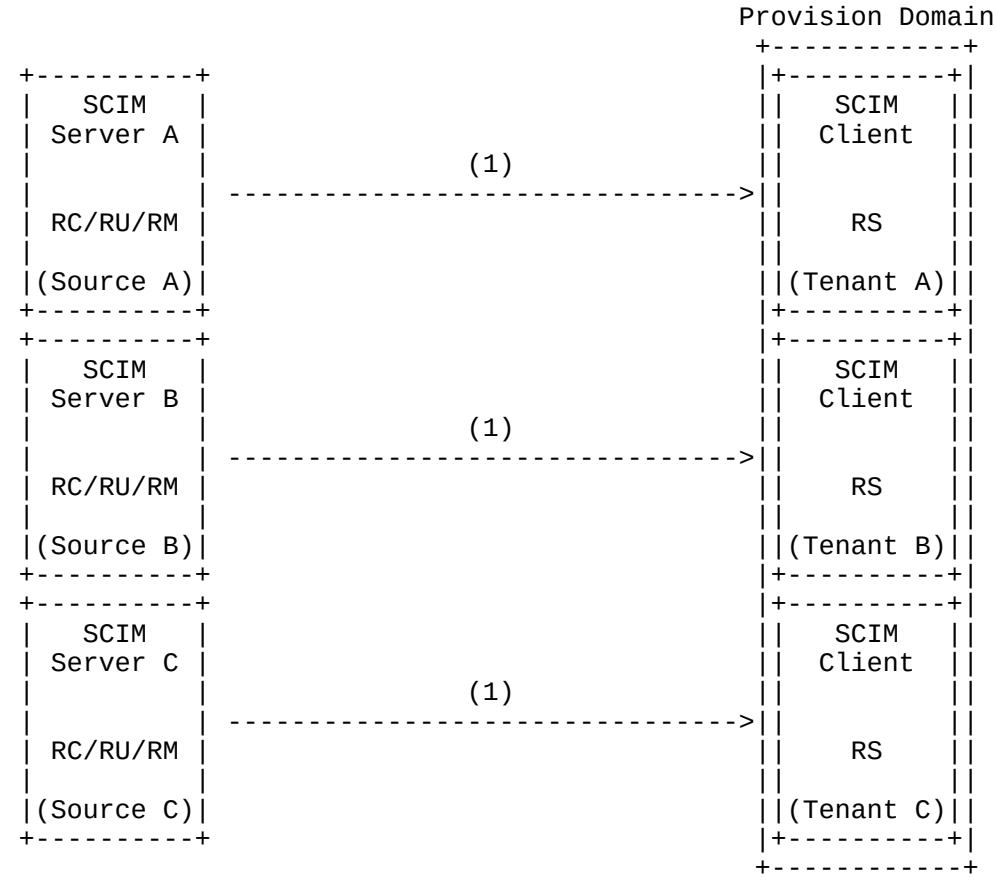

Figure 12: Multi-Tenant Resource Subscriber that is the SCIM Client

1. SCIM action - SCIM Client performs Active/Delta Pull

4.1.2. Resource Creator (RC/RU)

Single-tenant provisioning is done using a Resource Creator/Updater (RC/RU), which is responsible for creating the objects that will be passed across different systems. This is a very common and simple SCIM use case, where the SCIM Resource Object (SRO) and its SCIM Resource Object Attribute (SROA) are created. The CRUD operations on these resources trigger specific actions to facilitate the information exchange between two entities, typically the SCIM Client and Server. It is the responsibility of the Resource Creator/Updater to pass all relevant SCIM Resource Object Attribute (SROA) for that specific RS/RM. Typically, we find this kind of use case in small to mid-sized organizations, mainly in on-premises systems, where there

is no structured method to handle the resources.

4.1.2.1. Single-Tenant Resource Creator/Updater (RC/RU)

Resource Creator/Updater in a single tenant that can either be the SCIM Client or SCIM Server. Typically, we see this in an on-premise application.

4.1.2.1.1. Single-Tenant Resource Creator/Updater that is the SCIM Client

It is common today for the SCIM Client, typically performing the roles RC (Resource Creator) and RU (Resource Updater) to perform CRUD operations on the database of the RS (Resource Subscriber) or RM (Resrouce Manager) using the Active Push method. This action delivers SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) from a single-tenant provision service to a Consumer. A good example would be traditional on-premises HR (Human Resource) applications that creates SCIM Resource Object (SRO) either in central IdM (Identity Management) system or directly in a target aplications.

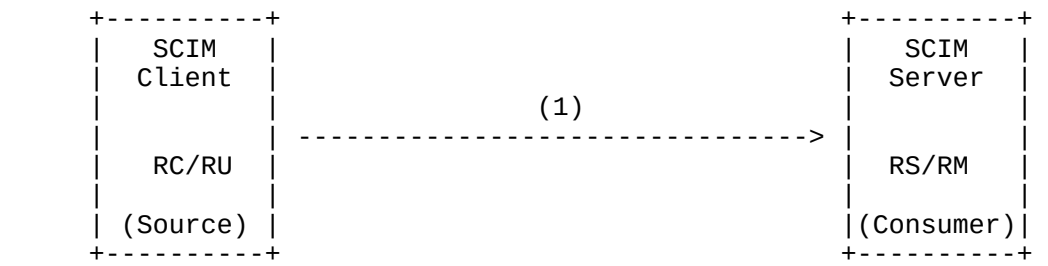

Figure 13: Single-Tenant Resource Creator/Updater that is the SCIM Client

1. SCIM action - SCIM Client performs Active Push

4.1.2.1.2. Single-Tenant Resource Creator/Updater that is the SCIM Server

The SCIM Client, which can be the RS (Resource Subscriber) or RM (Resource Manager), will perform CRUD operations on its own database using the Active and/or Delta Pull methods. Source information is available in the SCIM server, which is the source system responsible for the roles of RC (Resource Creator) and RU (Resource Updater) for the SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). A good example would be a traditional HR on-premises application (most commonly a single-tenant application) that

creates its own database of objects and provides them to a SCIM client. The SCIM client can either be an RS (Resource Subscriber), typically a standalone application that requires object information from the HR application, or an RM (Resource Manager), such as an on-premises IdM that will consolidate and add additional SCIM Resource Object Attribute (SROA) to the SCIM Resource Object (SRO). This option is a good solution for situations where the RS (Resource Subscriber) or RM (Resource Manager) is not reachable from the HR application.

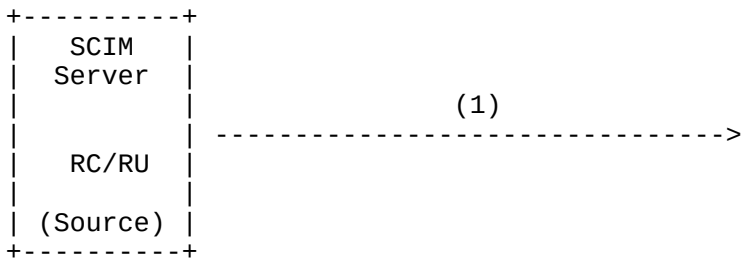

Figure 14: Single-Tenant Resource Creator/Updater that is the SCIM Server

1. SCIM action - SCIM Client performs Active/Delta Pull

4.1.2.2. Multi-Tenant Resource Creator/Updater (RC/RU)

It only differs from the Single-Tenant Resource Creator/Updater (RC/RU) by supporting multiple tenants. A typically would be an HR SaaS aplication.

4.1.2.2.1. Multi-Tenant Resource Creator/Updater that is the SCIM Client

It is common today for the SCIM Client, typically performing the roles of RC (Resource Creator) and RU (Resource Updater), to perform CRUD operations on the database of the RS (Resource Subscriber) or RM (Resource Manager) using the Active Push method. This action delivers SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) from a multi-tenant provision service to a consumer. A good example would be any new SaaS HR (Human Resources) application that creates SCIM Resource Object (SRO) either in a central IdM (Identity Management) system or directly in target applications.

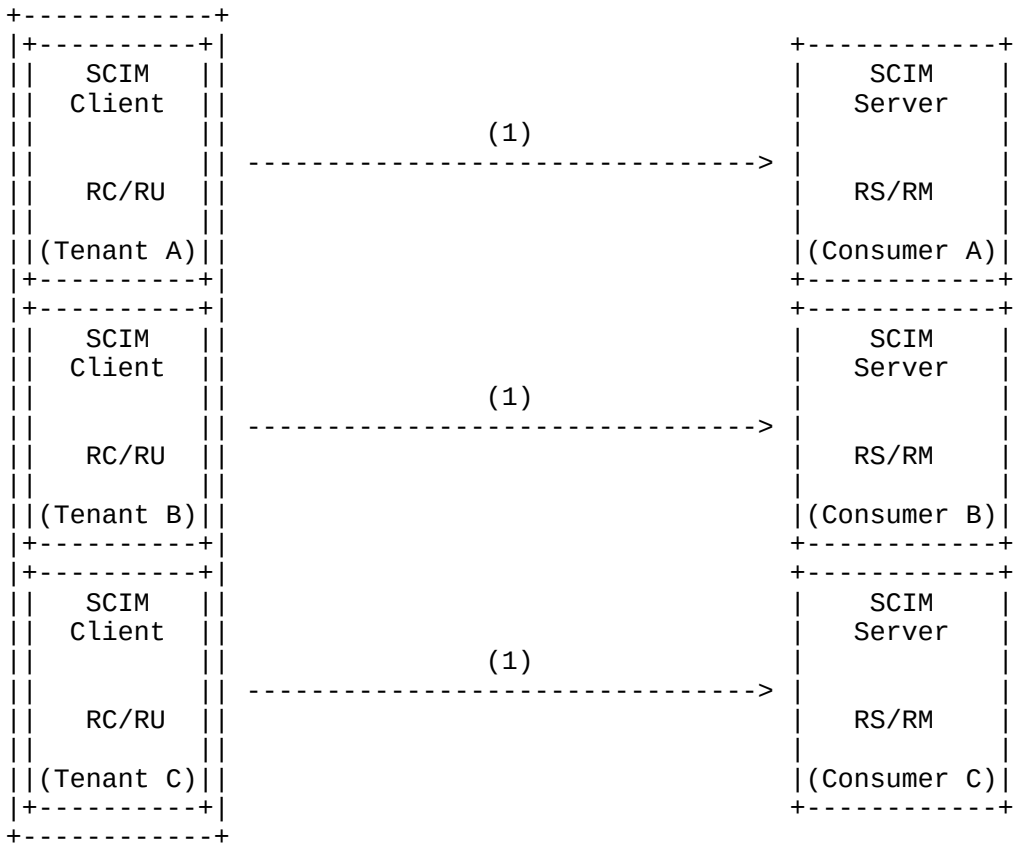

Figure 15: Multi-Tenant Resource Creator/Updater that is the SCIM Client

1. SCIM action - SCIM Client performs Active Push

4.1.2.2.2. Multi-Tenant Resource Creator/Updater that is the SCIM Server

The SCIM Client, which can be the RS (Resource Subscriber) or RM (Resource Manager), will perform CRUD operations on its own database using the Active and/or Delta Pull methods. Source information is available in the SCIM server, which is the source system responsible for the roles of RC (Resource Creator) and RU (Resource Updater) for the SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). A good example would be a SaaS HR (Human Resource) application (most commonly a multi-tenant application) that has its own database of objects and provides them to a SCIM client. The SCIM client can either be an RS (Resource Subscriber), typically a

standalone application that requires object information from the HR application, or an RM (Resource Manager), such as an on-premises IdM that will consolidate and add additional SCIM Resource Object Attribute (SROA) to the SCIM Resource Object (SRO).

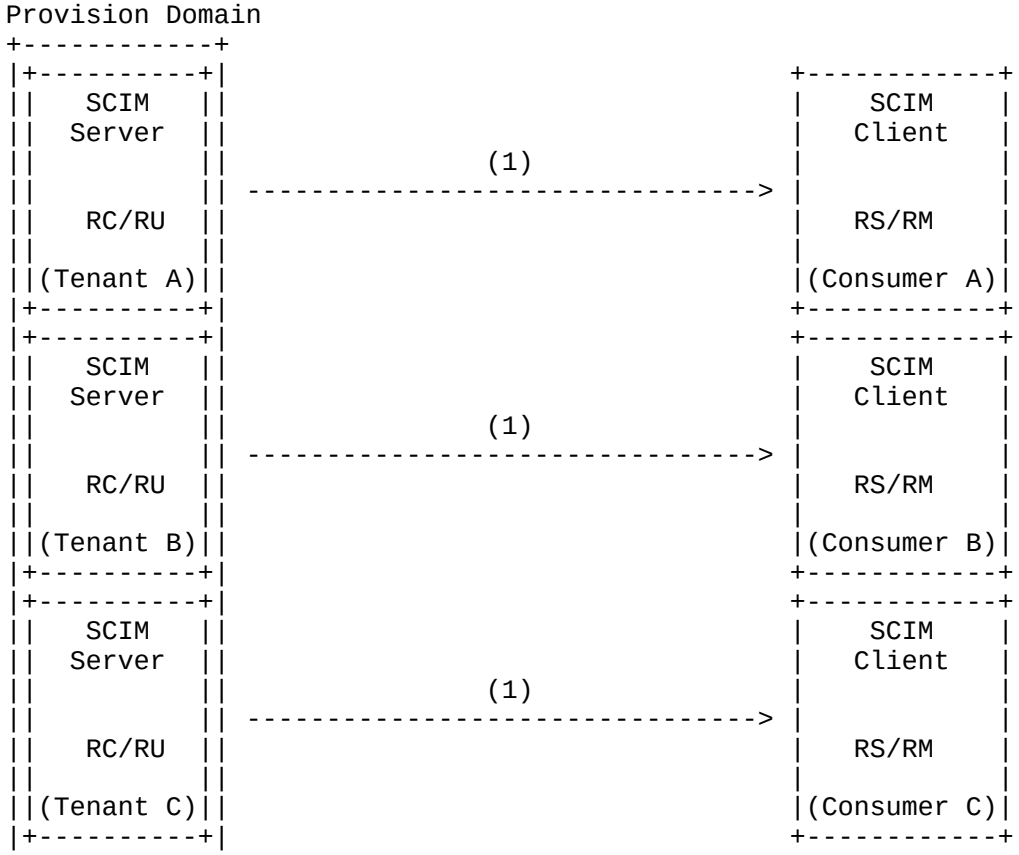

Figure 16: Multi-Tenant Resource Creator/Updater that is the SCIM Server

1. SCIM action - SCIM Client performs Active/Delta Pull

4.1.3. Resource Management (RM)

Typically, one or more upstream object databases populate the Resource Manager (RM), which then provides that resource information to downstream services requiring specific sets of the populated objects. The scenarios described in the next chapter will always outline the concept of upstream services, which are normally the sources of the objects, and downstream services, which are typically

the consumers of the objects. A single-tenant Resource Manager (RM) will receive SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) from an upstream entity, which can be either SCIM or non-SCIM. A good example of a non-SCIM upstream source would be connectors that synchronize users and groups using an HTTP REST interface to copy those objects from a database using legacy protocols like LDAP. Normally, the Resource Manager (RM) will accept objects from multiple sources, and it is its responsibility to understand which SCIM Resource Object Attribute (SROA) to obtain from each source. There might also be independent agreements for different groups of SCIM Resource Object (SRO). The Resource Manager (RM) can also assume the roles of Resource Creator (RC) and Resource Updater (RU), where some or all of the SCIM Resource Object (SRO) or some of their SCIM Resource Object Attribute (SROA) are created locally. These kinds of deployments are very common in greenfield deployments.

4.1.3.1. Single-Tenant Resource Manager (RM)

Single-Tenant Resrouce Manager are tipically Identity Manager (IdM) that are on-premises, where the upStream is tipically also on-premise but the Downstream can iether be on-premise, Cloud or hybrid application.

4.1.3.1.1. Single-Tenant Resource Manager that is the SCIM Server

The upstream service will provide one or more sources of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). If the source is a SCIM Client, it will use the Active Push method to deliver that information to the Resource Manager, which will be the SCIM Server and the consumer of those Resource Objects. The same Resource Manager will act as a SCIM server for the downstream consumer, which will be the SCIM Client performing the actions of Active/Delta Push. This is a partial implementation used by some IdM systems today, where they obtain Resource Objects from legacy databases using non-SCIM protocols and provide SCIM Resource Object (SRO) to downstream services, typically SaaS applications that need to create their own database of Resource Objects.

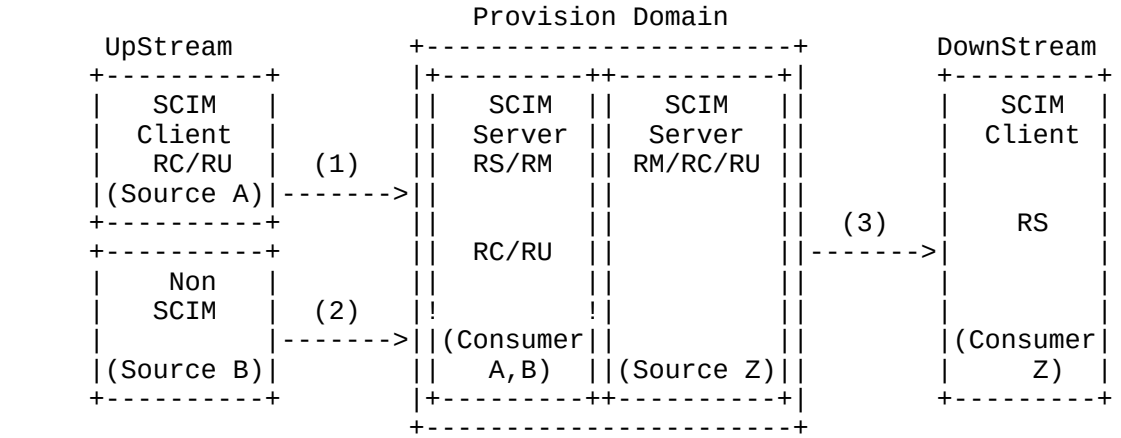

Figure 17: Single-Tenant Resource Manager that is the SCIM Server

1. SCIM action - SCIM Client performs Active Push

2. Non SCIM action

3. SCIM action - SCIM Client performs Active/Delta Pull

4.1.3.1.2. Single-Tenant Resource Manager that is the SCIM Client

The upstream service will provide one or more sources of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). If the source is a SCIM Server, the Resource Manager, which will act as a SCIM Client, will use the Active/Delta Pull method to obtain that information. The same Resource Manager will act as a SCIM Server for the downstream consumer and will perform the action of pushing a select group of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA) to the consumer service. This is a partial implementation used by some IdM systems today, where they obtain Resource Objects from legacy databases using non-SCIM protocols and provide SCIM Resource Object (SRO) to downstream services, typically SaaS applications that need to create their own database of Resource Objects.

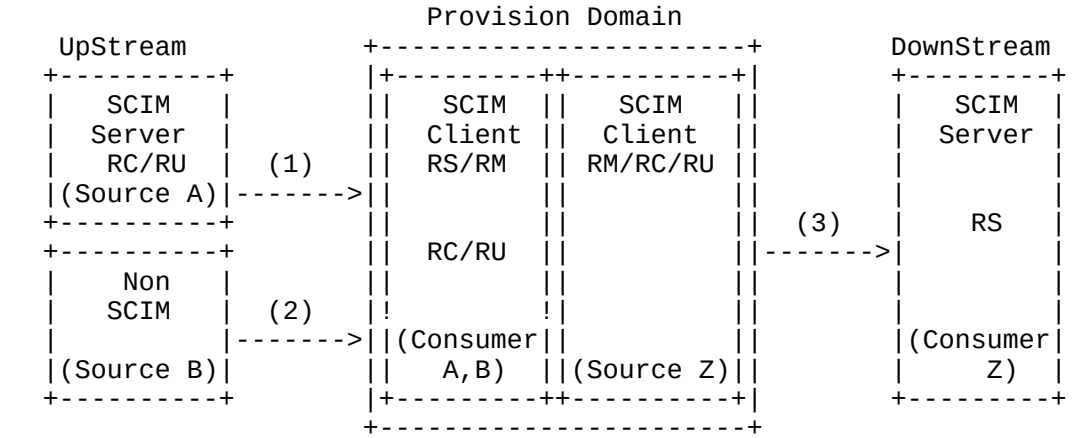

Figure 18: Single-Tenant Resource Manager that is the SCIM Client

1. SCIM action - SCIM Client performs Active/Delta Pull

2. Non SCIM action

3. SCIM action - SCIM Client performs Active Push

4.1.3.1.3. Single-Tenant Resource Manager that is the SCIM Server and SCIM Client

The upstream service will provide one or more sources of SCIM Resource Object (SRO) and their SCIM Resource Object Attribute (SROA). This scenario we will use as SCIM action Active/Delta Pull from the UpStream to the Resource Manager and the same action from it to the DownStream, for the scenarios where the initial Source is a SCIM server and the final Consumer is the SCIM Client. This scenarios we will use as SCIM action Active Push from the UpStream to the Resource Manager and the same action from it to the DownStream, for the scenarios where the initial Source is a SCIM Client and the final Consumer is the SCIM Server. This is a partial implementation used by some IdM systems today, where they obtain Resource Objects from legacy databases using non-SCIM protocols and provide SCIM Resource Object (SRO) to downstream services, typically SaaS applications that need to create their own database of Resource Objects.

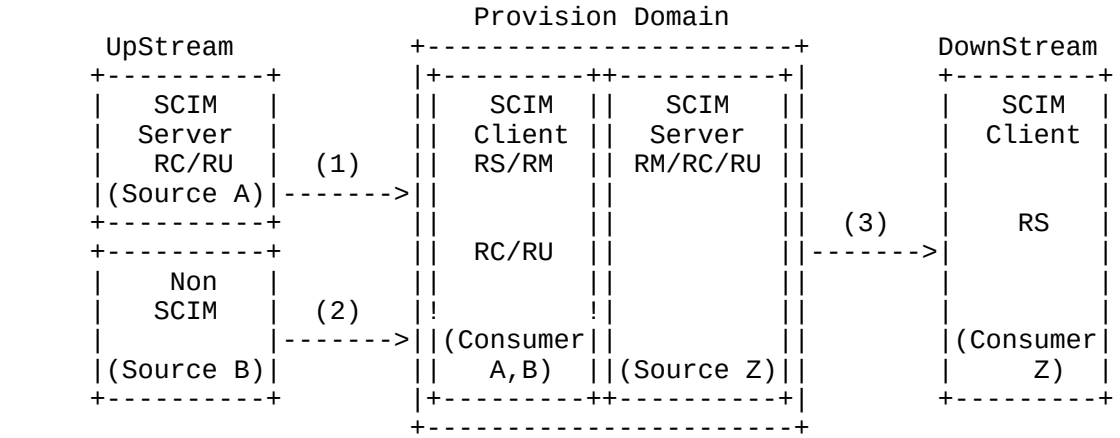

Figure 19: Single-Tenant Resource Manager that is the SCIM Client and SCIM Server

1. SCIM action - SCIM Client performs Active/Delta Pull

2. Non SCIM action

3. SCIM action - SCIM Client performs Active/Delta Pull

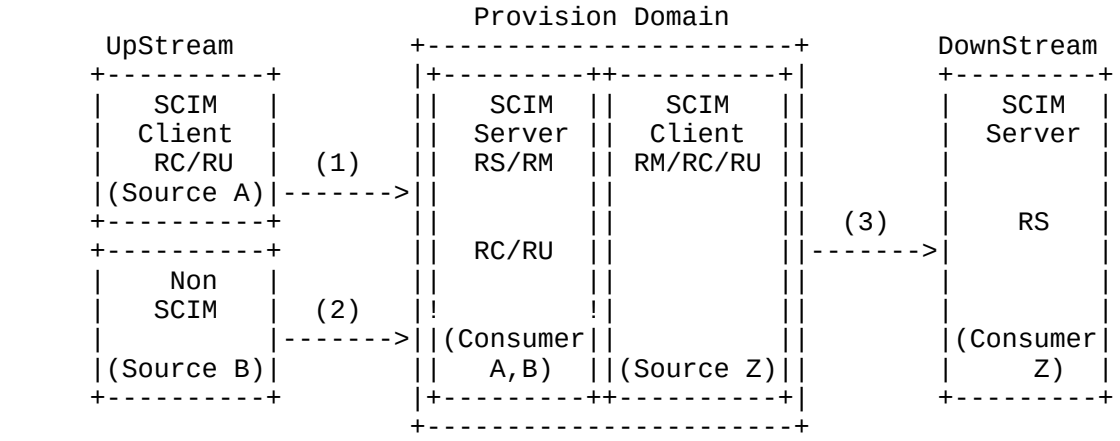

Figure 20: Single-Tenant Resource Manager that is the SCIM Server and SCIM Client

1. SCIM action - SCIM Client performs Active Push

2. Non SCIM action

3. SCIM action - SCIM Client performs Active Push

4.1.3.2. Multi-Tenant Resource Manager (RM)

Multi-Tenant Resource Manager are typically Identity Manager (IdM) that are cloud base, normally designated as IDaaS, where the upStream and Downstream are either on-premise or Cloud base.

4.1.3.2.1. Multi-Tenant Resource Manager that is the SCIM Server

Same information as Single-Tenant Resource Manager that is the SCIM Server but the Provision domain has multiple Tenants

4.1.3.2.2. Multi-Tenant Resource Manager that is the SCIM Client

Same information as Single-Tenant Resource Manager that is the SCIM Client but the Provision domain has multiple Tenants

4.1.3.2.3. Multi-Tenant Resource Manager that is the SCIM Server and SCIM Client

Same information as Single-Tenant Resource Manager that is the SCIM Server and SCIM Client but the Provision domain has multiple Tenants

4.2. Specific Implementations

4.2.1. Partner Device Registry

An important step in making a device work is to provide its details from the manufacturer to the customer. The SCIM Resource Object (SRO) of the device, provided by the manufacturer, includes its SCIM Resource Object Attribute (SROA), such as certificates, pairing protocols, and other relevant details.

4.2.1.1. Manufacturer details provided to customer by vendor that is the SCIM client

The manufacturer is the multi-tenant SCIM client and will push details of devices acquired by specific customers to their SCIM servers. The customer will provide the SCIM server and will receive information from the acquired devices. Additionally, the customer will manage the attributes of those devices, assuming the roles of Resource Subscriber (RS), Resource Updater (RU), and Resource Manager (RM). After the initial creation of the SCIM Resource Object (SRO) in the customer’s device database, it will be the server’s responsibility to add and update the SCIM Resource Object Attribute (SROA). Typically, the device will reach out to a device manager in the customer’s network, which will provide the SCIM server endpoint to the manufacturer. This task can also be done manually at the time of the device acquisition, allowing a SCIM push of the SCIM Resource

Object (SRO) to the customer’s device management platform.

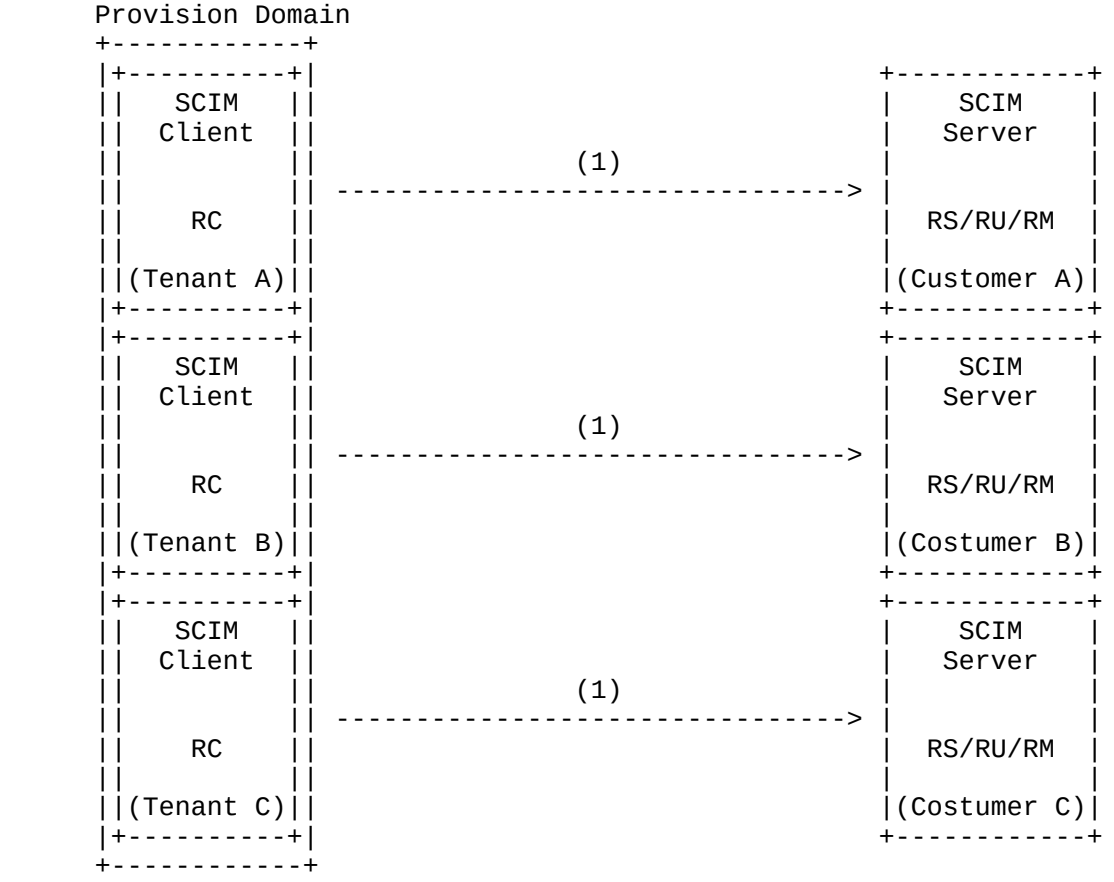

Figure 21: Manufactor is the SCIM client and push RO to Customers

1. SCIM action - SCIM Client performs Active Push

4.2.1.2. Manufacturer details provided by requesting it from Customer that is the SCIM client

The manufacturer is the multi-tenant SCIM server that holds the details of the Resource Objects, which it can provide to customers who acquire them. The customer will provide a SCIM client that will perform an Active Pull of the Resource Objects acquired from a specific manufacturer. The SCIM client will have the roles of Resource Subscriber (RS), Resource Manager (RM), and Resource Updater (RU), because after creating the SCIM Resource Object (SRO) in its object database, it will be responsible for updating and modifying that object. This use case is especially interesting for customers

whose Device Manager is not reachable from the Internet. In such cases, the Device Manager will act as a SCIM client and perform the action of pulling the SCIM Resource Object (SRO) from the multi-tenant SCIM server provided by the manufacturer.

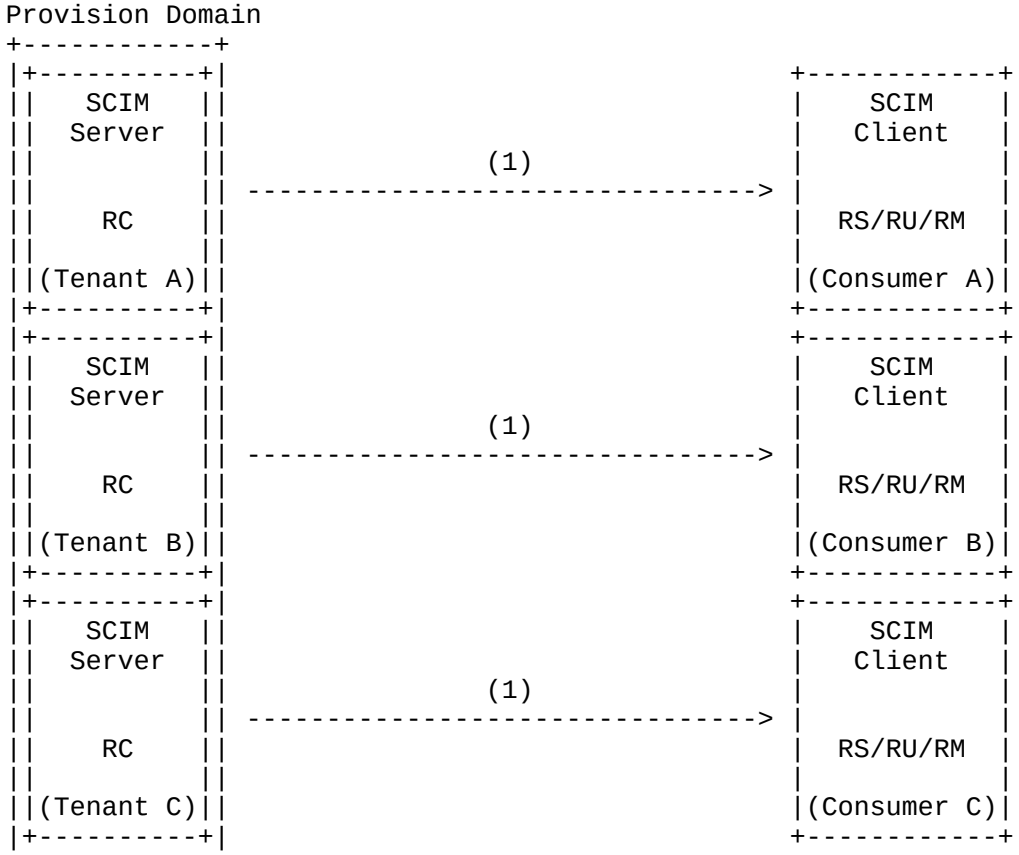

Figure 22: Manufactor is the SCIM Server and Customers Pull information about Device

1. SCIM action - SCIM Client performs Active Pull

4.2.2. Device Identity Creation from Commissioner Tool

When devices are initially provisioned from the client application (mobile application, web application, etc.), the client application will allow for the provision of additional details about the devices that are specific to that installation. Whether the commissioning tool is already SCIM-enabled or the client application includes the commissioning tool, there will ultimately be a SCIM action to perform

an Active Push. This action will provide the additional SCIM Resource Object Attribute (SROA) to be added to the SCIM Resource Object (SRO) that is maintained in the device manager.

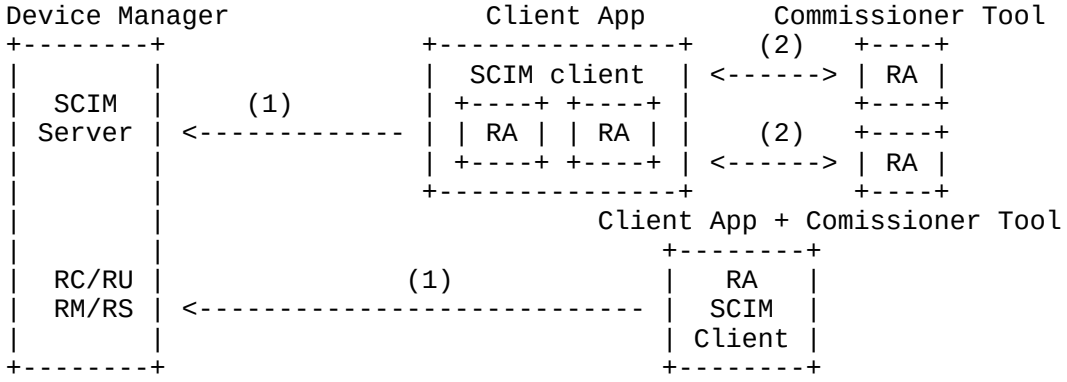

Figure 23: Commissioner tool provide Resource Attribtues to Device Manager

1. SCIM action - SCIM client performs Active Push

2. Non SCIM action

4.2.3. Client Applications gets directory Services

The client application retrieves information about all devices and their attributes from the Device Manager for their environments. The client application typically downloads the full list of devices daily during non-working hours, with an optional on-demand sync. SCIM clients should only be able to access the devices that they manage.

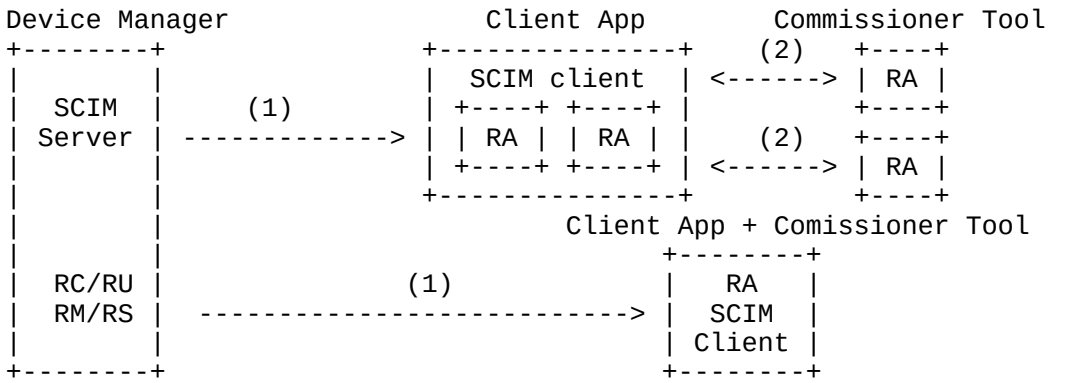

Figure 24: Device manager provides RO and RA to the Devices through Commissioner Tool

1. SCIM action - SCIM client performs Active Pull

2. Non SCIM action

4.2.4. Provide Credetials to manage Device

The Device Manager can provide Resource Attributes to the client application so that the devices can be configured using the commissioning tool. For example, the Device Manager can provide credentials to the device using the client application as the gateway. Through the commissioning tool, which can be a single entity, these credentials can be delivered to the device.

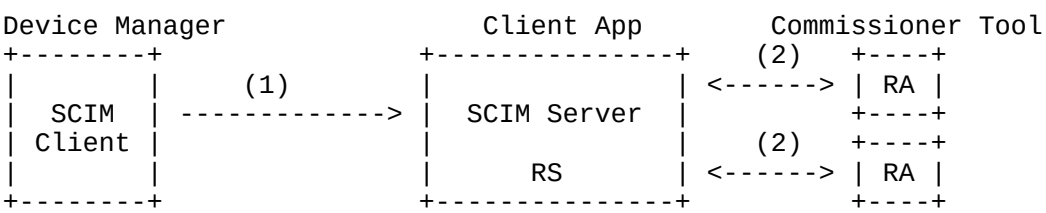

Figure 25: Device Manager provide Resource Attributes to the Commissioner tool to configure device

1. SCIM action - SCIM client performs Active Push

2. Non SCIM action

4.2.5. Enterprise "Last Mile" Applications

When provisioning to line of business Enterprise applications, implementers are often dealing with software that cannot be easily modified. As a result, it may be necessary to perform system integrations that run at the application layer, the database layer, or the framework layer in order to insert or change user data sourced from SCIM infrastructure. Connectors often use a client active pull over a periodic interval to keep the application in sync. It is also common for this pattern to include a just-in-time SSO trigger, so that should a new user try to access the line of business application before the resource has been created by the periodic active pull, they are created instead based on the contents of the user’s SAML assertion and then managed going forward by SCIM active pulls.

4.2.6. RA authority in SaaS Application

Sometimes, not all the SCIM Resource Object Attribute (SROA) of a SCIM Resource Object (SRO) are owned (created) by the Resource Creator (RC) or Resource Updater (RU). Very specialized SCIM Resource Object Attribute (SROA) can be the responsibility of a SaaS application. For example, an IdM should create user records with standard attributes like first name, last name, home address, etc., but the SaaS application should define the email attribute if that SaaS application is an email server.

4.2.6.1. Implementers Provision Domain is a SCIM Client and a SCIM server

The implementer’s domain acts as the SCIM Client and is the authority for regular attributes such as first name, last name, home address, etc., of a user. These attributes are created and updated by the Provision Domain, which functions as the Resource Manager (RM), Resource Creator (RC), and Resource Updater (RU). The application is the authority for one or more specific SCIM Resource Object Attribute (SROA), such as the email address of a given user. This means the application will serve as the Resource Manager (RM), Resource Creator (RC), and Resource Updater (RU) for those specific attributes only. Both the Provision Domain and the application will function as both the SCIM Client and SCIM Server for the respective SCIM Resource Object Attribute (SROA) they are responsible for. They will use the SCIM action of Active Push to pass the RSCIM Resource Object Attribute (SROA) of the SCIM Resource Object (SRO) to their counterpart. Thus, both the roles of SCIM Server and SCIM Client exist within the Provision Domain and the application.

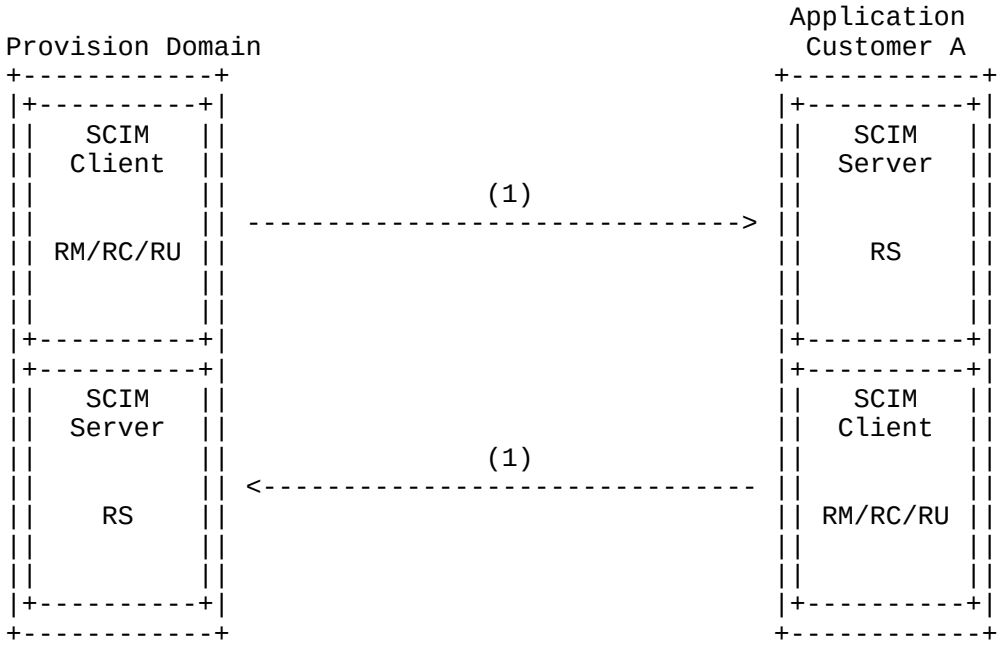

Figure 26: Single Ro with diferent RA authority implemented between the Provision Domain and the customer SaaS App

1. SCIM action - SCIM Client performs Active Push

4.2.6.2. Implementers Provision Domain is a SCIM Client

The implementer’s domain acts as the SCIM Client and is the authority for regular attributes, such as first name, last name, home address, etc., of a user. These attributes are created and updated by the Provision Domain, which functions as the Resource Manager (RM), Resource Creator (RC), and Resource Updater (RU). The application is the authority for one or more specific SCIM Resource Object Attribute (SROA), such as the email address of a given user. This means the application will serve as the Resource Manager (RM), Resource Creator (RC), and Resource Updater (RU) for those specific attributes only. In this use case, since the Provision Domain is always the SCIM Client and the application is always the SCIM Server, the Active Push method will be used for the regular attributes of the SCIM Resource Object (SRO). The Active/Delta Pull method will be used to retrieve the specialized SCIM Resource Object Attribute (SROA) that are the responsibility of the application.

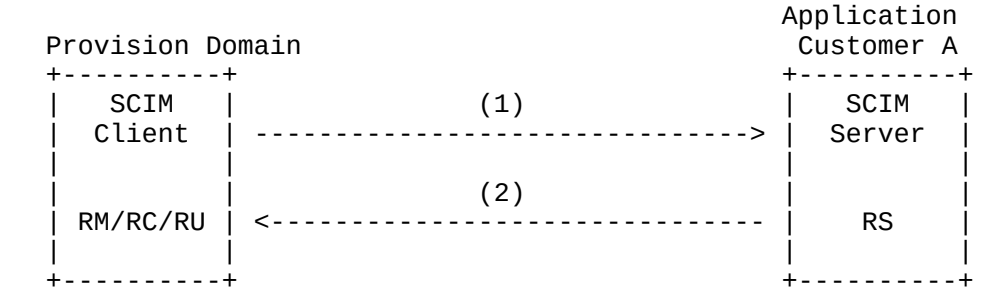

Figure 27: Single RO with diferent RA authority implemented between the Provision Domain and the customer SaaS App

1. SCIM action - SCIM Client performs Active Push

2. SCIM action - SCIM Client performs Active/Delta Pull

4.2.7. Reconciliations

Because of inconsistencies or mistakes in the SaaS App Resource Objects and it attributes might change and there is no visibility of the IdM that it happens. System will do reconciliation to make sure that SCIM Resource Object (SRO) and its SCIM Resource Object Attribute (SROA) are consistent across different systems. If there is a new attributes from SCIM Server in the Delta Pull, the SCIM client will do a push to fix it and make again synchronize

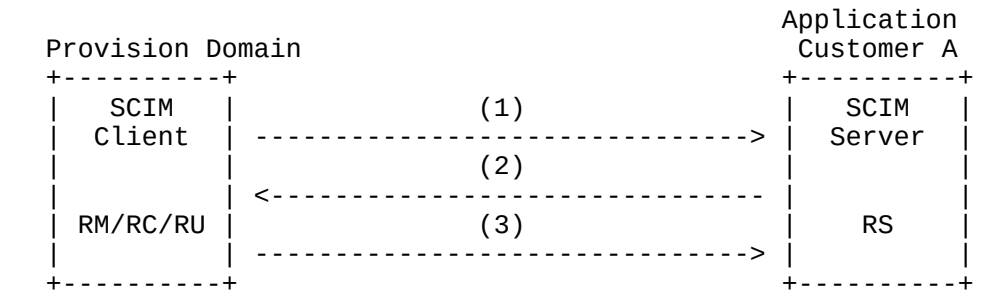

Figure 28: Reconciliation of RO/RA between IDM and Application

1. Regular SCIM action - SCIM Client performs Active Push

2. SCIM action - SCIM Client performs Active/Delta Pull

3. Remediation SCIM action - SCIM Client performs Active Push

5. Security Considerations

Authentication and authorization must be ensured for SCIM operations to guarantee that only authenticated entities can perform SCIM requests and that the requested SCIM operations are authorized. SCIM resources (e.g., Users and Groups) can contain sensitive information. Therefore, data confidentiality must be ensured at the transport layer. There can be privacy issues that extend beyond transport security, such as moving personally identifiable information (PII) offshore between different SCIM elements. Regulatory requirements must be met when migrating identity information between different jurisdictions (e.g., countries and states may have differing privacy regulations). 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 instead 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. IANA Considerations

There are no additional IANA considerations to those specified [RFC7643] and [RFC7644].

7. Acknowledgements

The editor would like to acknowledge the contribution to this draft versions of this document: Dean Saxe, Beyond Identity Eliot Lear, Cisco Systems

8. References

8.1. 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, <https://www.rfc-editor.org/rfc/rfc2119>.

8.2. 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, <https://www.rfc-editor.org/rfc/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, <https://www.rfc-editor.org/rfc/rfc7644>.

[RFC7832] Smith, R., Ed., "Application Bridging for Federated Access Beyond Web (ABFAB) Use Cases", RFC 7832, DOI 10.17487/RFC7832, May 2016, <https://www.rfc-editor.org/rfc/rfc7832>.

[RFC8417] Hunt, P., Ed., Jones, M., Denniss, W., and M. Ansari, "Security Event Token (SET)", RFC 8417, DOI 10.17487/RFC8417, July 2018, <https://www.rfc-editor.org/rfc/rfc8417>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/rfc/rfc9112>.

[SCIM_Profile_for_Security_Event_Tokens] Hunt, P., Cam-Winget, N., Kiser, M., and J. Schreiber, "SCIM Profile for Security Event Tokens", June 2025, <https://datatracker.ietf.org/doc/draft-ietf-scim-events>.

Authors' Addresses

Paulo Jorge Correia
Cisco Systems
Email: paucorre@cisco.com

Pamela Dingle
Microsoft Corporation
Email: pamela.dingle@microsoft.com