Co-ordination Use-Case Comparison Between SCIM Servers

IETF114 SCIM WG
July 29, 2022
Overall Scenario

• A domain (e.g. an employer) needs to co-ordinate SCIM accounts and events across multiple independent cloud domains

• Assumptions:
  • The life-cycle of accounts is often related but not 100% linked
    • E.g. suspension or deletion in one domain usually means a corresponding action in another domain (e.g. role revoke, suspension, or deletion)
  • Schema in each domain is likely different. Some attributes may be coordinated, mapped, or synchronized
  • Each domain has independent administration and value-added information
  • Linking domains may involve a "bootstrap" procedure
    • E.g. an initial load, re-conciliation, to link accounts across domains
  • Not every resource in each domain is coordinated
  • One domain MAY be considered a controller over the other
    • This implies the controller is aware of remote context schema and resources (can make SCIM Create, Modify, Delete requests)
Diagram – Cursor Paging

SCIM Requests

HTTP Client

Enterprise Domain A

HTTP Service

CRM Domain B

Logical DB copy

Reconciliation Process

Data reconciled per page or entire data set

Reconciliation actions

Multiple SCIM GET with Cursor Paging

Linked entity

Linked entity

Note: Arrows point in direction of HTTP Request
Diagram – Event Based

Enterprise Domain A

HTTP Client

Linked entity

Optional SCIM GET in minimal event profile

Real-time per-event change information

Reconciliation Process

Linked entity

Local action

Local Changes

CRM Domain B

HTTP Service

Note: Arrows point in direction of HTTP Request
Cases:

- Bootstrap and Recovery
- Controller Coordinated Provisioning
- Internal Domain Replication
- Bi-directional Coordinated Provisioning
- Coordinated Signaling
Case – Bootstrap or Recovery

• Description
  • On initial connection, a controller domain needs to transfer a bulk set of resources into a new or existing domain

• Outcome
  • After processing, the new domains are deployed
  • Resources between domain are linked (e.g. via externalid)
  • Are assumed to be out-of-date and require changing processing to keep both domains current
Case – Controller Coordinated Provisioning

• Description
  • A controlling domain coordinates and reconciles changes between domains by:
    • Reconciling remote changes: A controlling domain needs to know what remote resources have changed so that it may reconcile locally.
    • Reconciling local changes: A controlling domain needs to propagate local changes it has to keep the remote domain up-to-date.
      • before making those changes, the controller needs to know the current state of the remote domain (or deal with a lot of SCIM errors)
      • E.g. Sally is added to the CRM_Users group. For the CRM domain, the Controller needs to know if this is just a role change or whether Sally needs to be provisioned to the CRM Domain first before being granted the entitlement

• Outcome
  • The controller domain processes remote and local changes and performs reconciliation that may result in changes locally or remotely.
  • The change list MUST meet certain timelines to meet security and SLA objectives
    • This may be real-time (or near-real-time) to hourly.
Case – Internal Domain Replication/Coordination

• Description
  • Changes in a SCIM domain need to be propagated in some form between nodes in a timely fashion
  • There are likely many nodes, data-centers and cloud providers
  • Information could be in a master-slave tree network, but given distributed nature any node should be able to accept writes

• Challenge
  • Rather than a defined point-to-point directional control path, this case requires a many-to-many co-ordination or "bus" topology

• Outcome
  • Each node is made aware of changes by other nodes in the domain so that it may be reconciled locally
  • Any potential update collisions resolved by latest wins, regional, or schema priority weighting.
Case – Bi-directional Co-ordination

• Description
  • Each domain is mutually able to receive changes from the other domain for local reconciliation typically via a single mutual gateway
  • A single domain may be considered authoritative over resource life-cycle or specific attributes

• Outcome
  • Domains receive changes from each other and reconcile locally
  • As with replication, conflict resolution must be based on priority rules.
Case – Coordinated Signaling

• Description
  • A high level security signal is produced in one domain that may be of interest to another (e.g. password reset, login failures)

• Outcome
  • Upon receipt of a signal, the receiver processes the information in the local context and determines appropriate local action if any
  • The receiver may pass the signal on to other corporate security systems as appropriate
## Comparison

<table>
<thead>
<tr>
<th>Item</th>
<th>Cursor Paging</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique</td>
<td>Polling with SCIM cursor based paging query</td>
<td>Async Message delivered via backchannel via Bus, SET HTTP Push or SET HTTP Long Poll</td>
</tr>
<tr>
<td>Parallelism</td>
<td>None - single requestor calls single service provider</td>
<td>Can be pushed or pulled from SP domain to receiver domain. Many-to-many and bi-directional possible</td>
</tr>
<tr>
<td>Currency</td>
<td>Periodic polling</td>
<td>Real-time</td>
</tr>
<tr>
<td>Ease of implementation</td>
<td>Simple protocol extension to SCIM paging</td>
<td>Easy to re-use for “signals” and replication. Flexible transfer: SET/JWT are URL safe and independently secure Long Poll can re-use credentials (as with paging)</td>
</tr>
<tr>
<td>Implementation Challenges</td>
<td>* Difficult/impossible to implement for sharded data.  * May require temporary “cursor” stores in proxies  * Many databases limit cursors to subsets of data  * Client must process entire data set each cycle</td>
<td>* Requires implementation of message bus and/or SET Delivery methods  * Requires a new set of endpoints when using HTTP Push.  * Requires messaging infrastructure</td>
</tr>
<tr>
<td>Control</td>
<td>Master-slave – SCIM client controls linking and relationships using filters(?)</td>
<td>Mutual life-cycle. Per item linking Reconciliation action in control of event receiver</td>
</tr>
<tr>
<td>Bootstrap/Recovery</td>
<td>Via SCIM GET /* or Data export and import, SCIM Bulk Short-term reliability responsibility of receiver if back-paging not supported)</td>
<td>Via SCIM GET /* or Data export and import, SCIM Bulk Message Bus can be source of recovery. SET Transfer offers short term recovery. Reliability responsibility of receiver.</td>
</tr>
<tr>
<td>Replication</td>
<td>Requires a master-client hierarchy</td>
<td>Can be multi-master/bus based</td>
</tr>
<tr>
<td>Data costs / Information Risk</td>
<td>High – exposes all data on each cycle</td>
<td>Low – minimal profile only exposes “id”s of changed items. Replication exposes only changed data.</td>
</tr>
<tr>
<td>Security</td>
<td>Transport: TLS &amp; HTTP AuthZ (e.g. OAuth2), Data: Raw JSON</td>
<td>Transport: TLS &amp; HTTP AuthZ (OAuth2), Data: JWT Messages (signed and/or encrypted)</td>
</tr>
<tr>
<td>Signal cases</td>
<td>None – must be inferred in data</td>
<td>Support for RISC and other SET events</td>
</tr>
</tbody>
</table>

**Control**
- Master-slave – SCIM client controls linking and relationships using filters(?)

**Bootstrap/Recovery**
- Via SCIM GET /* or Data export and import, SCIM Bulk Short-term reliability responsibility of receiver if back-paging not supported)

**Replication**
- Requires a master-client hierarchy