SCIM
Internet-Draft
Intended status: Standards Track
Expires: 6 March 2026

P. Poreddy
Independent

September 2025

SCIM RoleAssignment Draft Specification v0.1
draft-poreddy-scim-role-assignment-00

Abstract

SCIM 2.0 defines "roles" and "entitlements" attributes on the User resource, but it lacks a standardized way to bind roles to specific scopes such as projects, tenants, or groups. This gap forces organizations to rely on group sprawl or non-standard encodings, preventing true interoperability. This document introduces a new SCIM resource type, _RoleAssignment_, which models scoped role bindings as first-class records, enabling portable provisioning, lifecycle governance, and compliance visibility.

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 5 March 2026.

Table of Contents

1. Introduction 2
1.1. Problem Statement 3
1.2. Proposed Solution 3
2. Requirements Language 4
3. Overview 4
4. Schema 4
4.1. Resource Type 4
4.2. Attributes (Top-Level) 4
4.3. subject Sub-Attributes 6
4.4. scope Sub-Attributes 6
4.5. role Sub-Attributes 7
4.6. grant Sub-Attributes 7
4.7. validity Sub-Attributes 8
4.8. status Semantics 8
4.9. Complete Example 9
5. Operations 9
5.1. Common Query Patterns 10
6. Error Handling 10
6.1. Error Example: Invalid Validity Window 10
6.2. Error Example: Conflicting Assignment 11
7. Backward Compatibility 11
8. Security Considerations 11
9. Privacy Considerations 11
10. IANA Considerations 11
11. Conformance 12
11.1. Server 12
11.2. Client 12
12. References 12
12.1. Normative References 12
12.2. Informative References 13
Appendix A. Change Log 14
A.1. draft-poreddy-scim-role-assignment-00 14
Appendix B. Author's Address 14
Author's Address 14

1. Introduction

1.1. Problem Statement

The SCIM protocol [RFC7643] [RFC7644] defines the User and Group resources and allows global roles to be attached to Users via the "roles" attribute. However, the specification does not provide a standardized way to associate roles with specific scopes such as projects, tenants, or device groups.

This limitation prevents SCIM from modeling the most common real-world requirement: assigning different roles to the same identity in different contexts.

For example, consider a user named Alice:

```
User: Alice
 +-- Global Role: Power User
 +-- Project A: Maintainer
 +-- Project B: Developer
 +-- Project C: ReadOnly
```

Today, there is no interoperable SCIM method to represent Alice’s per-project role bindings. Current workarounds include creating Groups for every {scope x role} combination, which leads to group sprawl and poor interoperability, or embedding scope names into free-form role strings, which are not machine-readable or portable.

These limitations are visible in real-world SCIM implementations: GitLab [GITLAB-SCIM] , Tanium [TANIUM-RBAC] , and scenarios in Microsoft Entra ID [AZURE-SCIM] .

1.2. Proposed Solution

This document introduces a new SCIM 2.0 resource, _RoleAssignment_, which makes scoped role bindings a first-class concept. Each RoleAssignment explicitly links a subject (for example, User), a scope (for example, Project), and a role (for example, Developer). Optional metadata such as validity periods, source system, and approver information enable lifecycle management and governance.

By standardizing RoleAssignments:

* Identity Providers can provision scoped roles in a portable way.
* Service Providers can expose and consume these assignments consistently.
* Auditors and governance systems can query "who has what role in which scope."

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview

The RoleAssignment resource complements existing SCIM resources. Whereas the User.roles attribute provides only coarse global roles, RoleAssignment expresses who (subject) has what role in which scope. This allows interoperable provisioning of scoped role bindings.

4. Schema

4.1. Resource Type

```
{
  "name": "RoleAssignment",
  "endpoint": "/RoleAssignments",
  "schema": "urn:ietf:params:scim:schemas:core:2.0:RoleAssignment",
  "schemaExtensions": []
}
```

4.2. Attributes (Top-Level)

The RoleAssignment resource defines the following top-level attributes.

Attribute	Type	Req	Multi	Mutability	Description
id	string	yes	no	readOnly	Provider-assigned unique identifier.
externalId	string	no	no	readWrite	Client-supplied correlation id.
subject	complex	yes	no	readWrite	The subject receiving the role.
scope	complex	yes	no	readWrite	The scope where the role applies.
role	complex	yes	no	readWrite	The role granted within the scope.
priority	integer	no	no	readWrite	Conflict resolution (higher wins). Default 0.
grant	complex	no	no	readWrite	Assignment metadata (source, reason, approver).
validity	complex	no	no	readWrite	Validity window in UTC.
status	string	no	no	readOnly	Computed lifecycle status.

Table 1

Priority usage guidance: When multiple active assignments exist for the same subject and scope, the assignment with the highest priority takes precedence. If priorities are equal, implementation-defined resolution rules apply.

4.3. subject Sub-Attributes

Attribute	Type	Req	Multi	Mutability	Description
value	string	yes	no	readWrite	Identifier of the subject.
$ref	reference	no	no	readWrite	URI to the subject resource, when available.
type	string	no	no	readWrite	Subject type. Standard: "User", "Group". "ServiceAccount" MAY be used. Vendor-specific types SHOULD use prefixes.

Table 2

4.4. scope Sub-Attributes

Attribute	Type	Req	Multi	Mutability	Description
type	string	yes	no	readWrite	Scope type. Common: "project", "tenant", "organization", "application", "environment".
value	string	yes	no	readWrite	Provider-meaningful identifier of the scope.
$ref	reference	no	no	readWrite	URI to the scope resource, if available. Optional when scope is opaque.

Table 3

4.5. role Sub-Attributes

Attribute	Type	Req	Multi	Mutability	Description
name	string	yes	no	readWrite	Display name of the role.
value	string	no	no	readWrite	Stable identifier for the role.
$ref	reference	no	no	readWrite	URI to a role catalog entry, if available. Providers with catalogs SHOULD include $ref for discovery.

Table 4

Role Discovery Guidance: Since role.$ref is optional, servers SHOULD support filtering on role and scope to enable discovery, for example:

```
GET /RoleAssignments?attributes=role,scope&filter=scope.type eq "project"
```

Providers that expose a role catalog MAY align discovery with the SCIM Roles and Entitlements approach [SCIM-ROLES-ENTITLEMENTS] .

4.6. grant Sub-Attributes

Attribute	Type	Req	Multi	Mutability	Description
source	string	no	no	readWrite	Originating system or process.
reason	string	no	no	readWrite	Human-readable justification.

approver	string	no	no	readWrite	Approver identifier or reference.

Table 5

4.7. validity Sub-Attributes

Attribute	Type	Req	Multi	Mutability	Description
validFrom	dateTime	no	no	readWrite	Start of validity window (UTC).
validTo	dateTime	no	no	readWrite	End of validity window (UTC).

Table 6

4.8. status Semantics

The "status" attribute is readOnly and computed using the following rules:

1. If the referenced subject is inactive: status = "suspended".

2. If current time < validity.validFrom: status = "pending".

3. If current time > validity.validTo: status = "expired".

4. Otherwise: status = "active".

Special cases:

* If validity.validFrom is null, assignment is immediately eligible.

* If validity.validTo is null, assignment does not expire.

* If explicitly revoked via DELETE or PATCH: status = "revoked".

Required status values:

* "active": assignment is currently effective.

* "expired": validity window has ended.

* "pending": assignment created but not yet effective.

* "suspended": subject inactive or assignment temporarily disabled.

* "revoked": assignment was explicitly withdrawn.

4.9. Complete Example

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:RoleAssignment"],
  "id": "assignment-12345",
  "externalId": "ext-assign-001",
  "subject": {
    "value": "alice@company.com",
    "$ref": "https://example.com/scim/v2/Users/alice",
    "type": "User"
  },
  "scope": {
    "type": "project",
    "value": "web-app-proj",
    "$ref": null
  },
  "role": {
    "name": "Developer",
    "value": "developer",
    "$ref": null
  },
  "priority": 100,
  "grant": {
    "source": "HR-System",
    "reason": "New team member onboarding",
    "approver": "manager@company.com"
  },
  "validity": {
    "validFrom": "2025-09-01T00:00:00Z",
    "validTo": "2026-09-01T00:00:00Z"
  },
  "status": "active"
}
```

5. Operations

* _Create (POST):_ Servers MUST validate subject, scope, and role.

* _Replace (PUT):_ Missing required attributes cause 400 Bad Request.

* _Patch (PATCH):_ MUST be supported; invalid states SHOULD be rejected.

* _Delete (DELETE):_ Removes assignment and SHOULD revoke permissions.

* _Filter (GET):_ MUST support filters on subject, scope, role.

* _Pagination/Sorting:_ MUST support startIndex and count; SHOULD support sorting.

* _Bulk:_ MAY be supported. Clients SHOULD use externalId for idempotency.

5.1. Common Query Patterns

- User’s assignments:
  GET /RoleAssignments?filter=subject.value eq "alice@company.com"

- Scope permissions:
  GET /RoleAssignments?filter=scope.value eq "project-x"

- Role discovery within a scope type:
  GET /RoleAssignments?attributes=role&filter=scope.type eq "project"

- Expiring access before a date/time:
  GET /RoleAssignments?filter=validity.validTo le "2025-12-31T23:59:59Z"

6. Error Handling

* 400 Invalid Value for malformed attributes or invalid validity windows.

* 404 Not Found for unknown subject, scope, or role references.

* 409 Conflict for duplicate active assignments.

* 412 Precondition Failed for ETag mismatches on conditional updates.

Error responses SHOULD include "detail" and "scimType" per [RFC7644]
.

6.1. Error Example: Invalid Validity Window

```
   {
     "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
     "status": "400",
     "scimType": "invalidValue",
     "detail": "validity.validFrom must be before validity.validTo"
   }
```

6.2. Error Example: Conflicting Assignment

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:Error"],
  "status": "409",
  "scimType": "uniqueness",
  "detail": "Active assignment exists for subject 'alice@company.com' with role 'admin' i
n scope 'project-x'"
}
```

7. Backward Compatibility

Providers MAY continue to expose global roles via User.roles and coarse-grained membership via Groups. Providers SHOULD offer mapping guidance between legacy models and RoleAssignments. Dual-writing (maintaining both representations) is RECOMMENDED during migration.

8. Security Considerations

* RoleAssignment creation and modification are privileged operations.

* Servers MUST validate references to prevent privilege escalation.

* Lifecycle events SHOULD be logged with actor and justification.

* Replay and bulk abuse MUST be mitigated with rate limiting and idempotency.

9. Privacy Considerations

RoleAssignments may expose organizational structures and access patterns. Sensitive metadata SHOULD follow least-privilege disclosure.

10. IANA Considerations

This specification requests registration of the following SCIM schema:

URN: urn:ietf:params:scim:schemas:core:2.0:RoleAssignment
Specification: this document
Contact: IETF SCIM Working Group
Change Controller: IESG

Experimental namespace MAY also be used:
URN: urn:ietf:params:scim:schemas:extension:role:1.0:RoleAssignment

URNs are assigned and interpreted in accordance with [RFC8141] .

11. Conformance

11.1. Server

* MUST implement RoleAssignment and advertise it in /ResourceTypes.
* MUST validate subject, scope, and role.
* MUST enforce authorization on RoleAssignment operations.
* MUST support GET, POST, PUT, PATCH, DELETE, filtering.
* SHOULD support sorting and bulk operations.

11.2. Client

* MUST construct RoleAssignments per schema.
* MUST process error responses per [RFC7644] .
* SHOULD use externalId for idempotency.
* SHOULD honor ETag preconditions.

12. References

12.1. 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/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

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

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/info/rfc8141>.

12.2. Informative References

[GITLAB-SCIM]
Documentation, G., "Set up SCIM for GitLab groups", Online accessed September 2025, September 2025, <https://docs.gitlab.com/administration/settings/scim_setup/>.

[TANIUM-RBAC]
Documentation, T., "Role-based access control in the Tanium Console", Online accessed September 2025, September 2025, <https://help.tanium.com/bundle/ug_console_cloud/page/platform_user/console_roles.html>.

[AZURE-SCIM]
Learn, M., "Issue provisioning multiple roles to a SCIM app", Online accessed September 2025, September 2025, <https://learn.microsoft.com/en-us/answers/questions/1632657/issue-provisioning-multiple-roles-to-a-scim-app>.

[SCIM-ROLES-ENTITLEMENTS]
Zollner, C., "Roles and Entitlements Extension for SCIM", Work in Progress, Internet-Draft, draft-zollner-scim-roles-entitlements-extension-02, June 2025, <https://datatracker.ietf.org/doc/draft-zollner-scim-roles-entitlements-extension/>.

Appendix A. Change Log

A.1. draft-poreddy-scim-role-assignment-00

Initial version. Defines RoleAssignment schema, attributes, and operations. Adds priority, complete example, error examples, and query patterns. Includes error handling, backward compatibility, and IANA registration. Updates BCP14 boilerplate; replaces non-ASCII; converts ASCII tables to RFCXML tables; updates Roles/Entitlements reference to -02; cites RFC8141.

Appendix B. Author’s Address

Prithvi Poreddy
 Email: <prithvikrishnab4u@gmail.com>

Author’s Address

Prithvi Poreddy
Independent
United States of America
Email: prithvikrishnab4u@gmail.com