Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 31 October 2026

S. Kushwaha
Oracle Corporation

29 April 2026

SCIM DID/VC Binding Extension
draft-kushwaha-scim-didvc-binding-00

Abstract

This document defines an extension to the System for Cross-domain Identity Management (SCIM) for binding SCIM User resources to decentralized identity artifacts, including Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). The extension introduces a read-only SCIM schema extension for User resources that exposes binding state for discovery, a new SCIM resource type named IdentityBinding that records auditable linkage between a SCIM user and one or more DIDs and credential references, and an optional SCIM schema extension for ServiceProviderConfig that advertises server capabilities for DID and VC binding.

This specification intentionally does not define DID resolution, credential issuance, credential transport, or authentication flows. Instead, it defines how a SCIM service provider represents, discovers, queries, and manages binding state derived from those systems.

Table of Contents

1. Introduction

SCIM provides a standard protocol and schema model for provisioning and managing identities across administrative domains [RFC7643] [RFC7644]. SCIM supports new resource types and schema extensions, publishes schema metadata through Schema resources, publishes resource metadata through ResourceType resources, and provides service capability discovery through ServiceProviderConfig [RFC7643]. DID Core defines DID documents, verification methods, and proof-purpose relationships, while the Verifiable Credentials Data Model defines issuer, holder, verifier, status, schema, and validation concepts for verifiable credentials [DID-CORE] [VC-DATA-MODEL-2.0].

In many deployments, SCIM is the system of record for account lifecycle, while DID and VC systems are the system of record for decentralized identifiers and cryptographic trust. Today that linkage is usually proprietary: implementers invent local attributes, cannot query consistently, and cannot express lifecycle events such as DID deactivation, key rotation, or VC revocation in an interoperable SCIM form. This specification fills that gap by standardizing binding state, not by embedding full DID documents or full VC payloads.

1.1. 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.

SCIM resource representations in this document are encoded in JSON [RFC8259] as required by [RFC7643].

1.2. Scope

This specification applies to SCIM User resources only.

This specification defines schema and resource representations, discovery behavior, filtering, pagination, and PATCH expectations for binding resources, and lifecycle and privacy semantics for DID and VC bindings.

This specification does not define DID method semantics, DID resolution protocols, VC issuance or presentation protocols, proof formats, wallet protocols, or authentication and federation replacement.

1.3. Problem Statement

A provisioning client frequently needs to answer questions such as:

* Which SCIM user is bound to this DID?
* Which users have an active workforce or employment credential from a trusted issuer?
* What is the effect of credential revocation on the provisioned account?
* Can the service provider support pairwise DIDs, credential status checking, and external verifier delegation?

Base SCIM does not answer those questions in an interoperable way.

1.4. Design Goals

This specification has four goals:

1. preserve SCIM as the provisioning and lifecycle layer;
2. preserve DID and VC systems as the cryptographic trust layer;
3. minimize correlation and over-disclosure; and
4. define a small interoperable subset that clients and servers can implement consistently.

2. Architecture Overview

This specification uses a dual-layer model:

* A read-only User extension provides lightweight discoverability.

* A mutable IdentityBinding resource carries binding state, DID references, credential references, and lifecycle state.

A SCIM service provider MAY validate DID and VC evidence itself or MAY delegate validation to an external verification service. If validation is delegated, the wire protocol between the SCIM service provider and the verifier is out of scope. The SCIM service provider remains responsible for exposing standards-conformant SCIM resources and state transitions.

A service provider conforming to this specification MUST NOT persist full DID documents in SCIM resources and MUST NOT persist full VC payloads in SCIM resources. A service provider SHOULD store only references, identifiers, validation timestamps, and normalized outcomes.

3. Data Model

This specification defines three distinct status vocabularies: per-DID status reflects DID document state (such as whether a DID has been verified or deactivated), per-credential status reflects verifiable credential lifecycle state (such as validity-period or revocation outcomes), and binding-level status reflects the composite lifecycle of the IdentityBinding resource itself. These vocabularies are intentionally not unified. The aggregate User.bindingState additionally defines the value none to indicate a user with no bindings (see Section 3.1), which does not appear in any resource-level status.

3.1. User Extension Schema

The schema URI for the User extension defined by this specification is urn:ietf:params:scim:schemas:extension:didvc:2.0:User.

This extension is a derived, read-only projection of the user’s decentralized identity binding state. Clients do not create bindings by writing this extension; they create or modify IdentityBinding resources.

Attribute	Type	MV	Req	Mutability	Values	Description
primaryDid	string	no	no	readOnly	-	The verified primary DID currently associated with the user, if any. Derived from the active primary IdentityBinding per the algorithm in Section 3.1.1.
bindingState	string	no	yes	readOnly	none, pending, active, suspended, revoked, rejected	Aggregate state derived from active bindings for the user. The value none indicates the user has no associated IdentityBinding resources and does not appear in binding-level status vocabularies.
bindingRefs	complex	yes	no	readOnly	-	References to IdentityBinding resources associated with the user.

Table 1: User Extension Attributes

Sub-Attribute	Type	Req	Mutability	Values	Description
value	string	yes	readOnly	-	The SCIM id of the IdentityBinding resource.
$ref	reference	yes	readOnly	-	URI of the IdentityBinding resource.
display	string	no	readOnly	-	Human-readable label for the binding.
primary	boolean	no	readOnly	-	Indicates whether this binding is the source of primaryDid.
status	string	yes	readOnly	pending, active, suspended, revoked, rejected	Current lifecycle state of the referenced binding.

Table 2: bindingRefs Sub-Attributes

A conforming server MUST derive primaryDid, bindingState, and bindingRefs from IdentityBinding resources, MUST NOT allow clients to directly mutate this extension, MUST omit primaryDid when no verified primary DID exists, and MUST ensure that at most one bindingRefs entry has primary=true.

3.1.1. Derivation of primaryDid

A server MUST derive User.primaryDid using the following algorithm:

1. Collect all IdentityBinding resources whose user.value matches the target user and whose binding-level status is active.

2. From that set, select the binding for which bindingRefs.primary=true is set in the User extension projection. At most one such binding is permitted.

3. Within that binding, select the dids entry with primary=true and status=verified.

4. Set User.primaryDid to the value of that DID entry. If no such entry exists, omit User.primaryDid.

A server MUST NOT set User.primaryDid from a binding that is not in active status, and MUST clear User.primaryDid when the binding that sourced it transitions out of active status or is deleted.

If multiple active bindings each contain a DID entry with primary=true, the server MUST reject creation or PATCH operations that would produce this ambiguous state with HTTP 400 and a SCIM error type of invalidValue.

3.2. IdentityBinding Resource

The IdentityBinding resource type uses the endpoint /IdentityBindings and the schema URI urn:ietf:params:scim:schemas:extension:didvc:2.0:IdentityBinding.

An IdentityBinding resource represents the auditable linkage between one SCIM User resource and one or more DIDs, together with optional references to credentials whose validation influences the state of the binding.

All IdentityBinding resources include the standard SCIM common attributes defined in Section 3.1 of [RFC7643], including id, externalId, and meta. The meta.lastModified timestamp records the time of the most recent update to the resource representation. The lastValidationAttempt attribute defined in this specification is distinct from meta.lastModified: it records when a validation cycle was last run against the DID and credential evidence, which may differ from when the resource itself was last written.

Attribute	Type	MV	Req	Mutability	Values	Description
user	complex	no	yes	immutable	-	Reference to the bound SCIM User.
correlationModel	string	no	yes	immutable	pairwise, shared, public	Privacy expectation for DID reuse across relying parties. See Section 3.3 for the relationship

						between this field and per-DID relationship values.
dids	complex	yes	yes	readWrite	-	DIDs associated with the binding.
credentials	complex	yes	no	readWrite	-	Credential references associated with the binding.
status	string	no	yes	readOnly	pending, active, suspended, revoked, rejected	Binding lifecycle state.
lastValidationAttempt	dateTime	no	no	readOnly	-	Timestamp of the most recent validation attempt, regardless of outcome. Updated on both successful and failed validation runs. Distinct from meta.lastModified, which reflects resource write time.
statusReason	string	no	no	readOnly	-	Human-readable explanation for the current status.

Table 3: IdentityBinding Attributes

Sub-Attribute	Type	Req	Mutability	Description
value	string	yes	immutable	The SCIM id of the bound User.
$ref	reference	yes	immutable	URI of the bound User.
display	string	no	readOnly	Human-readable display name for the user.

Table 4: user Sub-Attributes

Sub-Attribute	Type	Req	Mutability	Values	Description
value	string	yes	readWrite	-	DID URI.
relationship	string	yes	readWrite	primary, pairwise, delegated, recovery	Role of the DID in the binding. See Section 3.3 for the distinction between this field and the binding-level correlationModel. The value recovery designates a DID that is authorized to execute key recovery operations for the primary DID of the binding, as defined by the applicable DID method; its presence does not affect binding-level status derivation.

verificationMethod	string	no	readWrite	-	DID URL of the verification method used for proof evaluation.
proofPurpose	string	no	readWrite	authentication, assertionMethod	DID proof purpose expected for verification.
controller	string	no	readWrite	-	DID controller identifier when known.
primary	boolean	no	readWrite	-	Indicates the preferred DID for the binding.
status	string	yes	readOnly	pending, verified, deactivated	Per-DID validation status. The value deactivated aligns with the DID Core specification's terminal state for a DID document [DID-CORE].

Table 5: dids Sub-Attributes

Sub-Attribute	Type	Req	Mutability	Values	Description
credentialId	string	no	readWrite	-	Identifier of the VC, if present.
types	string	yes	readWrite	-	VC type values relevant to the binding. Multi-valued.
issuer	string	yes	readWrite	-	Issuer identifier,

					commonly a DID or URI.
holder	string	no	readWrite	-	Holder identifier, if expressed by the ecosystem. Implementers SHOULD omit this field unless explicitly required, as holder DIDs can create correlation risk (see Section 9).
credentialSubjectId	string	no	readWrite	-	Credential subject identifier, if available. Implementers SHOULD omit this field unless explicitly required, as subject DIDs can create correlation risk (see Section 9).
statusRef	reference	no	readWrite	-	Reference to credential status information.
schemaRef	reference	no	readWrite	-	Reference to credential schema information.
validFrom	dateTime	no	readWrite	-	Start of

					credential validity, if known.
validUntil	dateTime	no	readWrite	-	End of credential validity, if known.
status	string	yes	readOnly	pending, active, revoked, expired, unknown	Server-normalized outcome for the credential reference.

Table 6: credentials Sub-Attributes

A conforming server MUST require at least one dids entry in every IdentityBinding, MUST permit at most one dids entry with primary=true, MUST require relationship=primary on any DID entry with primary=true, and MUST set status=pending on creation unless the server can validate synchronously before returning the resource.

A conforming server MUST NOT set status=active unless at least one DID entry has status=verified. The server MUST update User.primaryDid when the binding becomes active or inactive per the algorithm in Section 3.1.1, and MUST reject creation of more than one active IdentityBinding containing the same DID value within a single administrative scope (the SCIM service provider, or in multi-tenant deployments, a single tenant) when the binding's correlationModel is pairwise, regardless of which User is referenced.

3.3. Relationship Between correlationModel and dids.relationship

The correlationModel attribute on an IdentityBinding and the relationship sub-attribute on individual DID entries operate at different levels of abstraction and are not redundant.

correlationModel is a binding-level declaration of the privacy expectation for DID reuse across relying parties. A value of pairwise means the DIDs in this binding are intended for use only with this relying party and MUST NOT be reused at other services. A value of shared means the DIDs may be shared across a defined set of relying parties. A value of public means the DIDs are publicly shareable without restriction.

dids.relationship describes the functional role of a specific DID within the binding (e.g., primary for the main authentication DID, delegated for a capability-delegation DID, recovery for a key-recovery DID). A binding with correlationModel=pairwise may contain multiple DID entries with different relationship values; the correlation model applies to all of them collectively.

The combination correlationModel=shared with relationship=pairwise on an individual DID entry is permitted when one DID in the binding is used only pairwise while others in the same binding are shared.

3.4. ServiceProviderConfig Extension

The schema URI for the ServiceProviderConfig extension defined by this specification is urn:ietf:params:scim:schemas:extension:didvc:2.0:ServiceProviderConfig.

This extension advertises the server's DID and VC binding capabilities without modifying the core ServiceProviderConfig schema.

Attribute	Type	MV	Req	Mutability	Values	Description
enabled	boolean	no	yes	readOnly	-	Indicates whether DID/VC binding is currently enabled for new IdentityBinding creation. A server MAY deploy the extension schema while setting this to false to indicate temporary suspension of new binding intake.
verificationDelegation	string	no	yes	readOnly	internal, external, mixed	Whether verification is performed by the SCIM server (internal), an external

						verifier (external), or both depending on DID method or credential type (mixed). When mixed, clients SHOULD NOT assume a specific verification path for any given DID method or credential type without out-of-band knowledge.
supportedDidMethods	string	yes	no	readOnly	-	DID methods accepted by the service provider.
supportedCredentialChecks	string	yes	no	readOnly	status, schema, validity, holderBinding	VC checks the server is capable of evaluating.
supportedCorrelationModels	string	yes	yes	readOnly	pairwise, shared, public	Correlation models supported by the server.

Table 7: ServiceProviderConfig Extension Attributes

4. Processing Model

4.1. Creation

A client creates a User resource using normal SCIM semantics.

A client creates an IdentityBinding by POSTing to /IdentityBindings. The client MUST provide user, correlationModel, and at least one dids entry. The client MAY include credentials. The server MUST verify that user.value references an existing User resource and MUST return HTTP 400 with SCIM error type invalidValue if the reference cannot be resolved.

Upon creation, the server MUST create the resource in pending status unless synchronous validation has already completed successfully, MUST record normalized validation results in per-DID and per-credential status fields, and MUST update the derived User extension when binding state changes. The server MUST return HTTP 201 with a Location header pointing to the created resource.

4.2. Validation

This specification requires that DID control be verified before a binding may become active. How a client presents DID control evidence to the SCIM server (e.g., via a challenge-response protocol, a Verifiable Presentation submitted in a related flow, or a proof in a request header) is out of scope for this specification. Implementations SHOULD consult relevant specifications such as OpenID for Verifiable Presentations or DIF Presentation Exchange for proof transport mechanisms that can be integrated with SCIM deployment.

To validate DID control, the server or delegated verifier MUST resolve the DID and verify that the verification method used for proof checking is authorized for the appropriate proof-purpose relationship in the DID document [DID-CORE]. The server MUST re-resolve DID documents when revalidating a binding, and MUST NOT rely on cached DID resolution results whose age exceeds the server’s configured revalidation window.

To validate a credential reference, the server or delegated verifier MUST, where applicable, verify cryptographic integrity, evaluate issuer and controller consistency, evaluate validFrom and validUntil, evaluate credential status when status information is available, and evaluate credential schema when schema information is available [VC-DATA-MODEL-2.0].

This specification does not define proof formats or verifier APIs.

4.3. State Transitions

The server MUST implement the following binding-state semantics:

From	To	Condition
pending	active	Required DID proof and all required credential checks succeed.
pending	suspended	Validation attempt is temporarily inconclusive before the binding has ever been active (e.g., DID resolver unavailable at creation time).
pending	rejected	Validation fails conclusively before the binding has ever been active.
active	suspended	Revalidation is temporarily inconclusive, such as unavailable resolver or unknown credential status.
active	revoked	DID deactivation, credential revocation, or policy failure is confirmed.
suspended	active	Revalidation succeeds.
suspended	revoked	Negative status is later confirmed.
rejected	pending	Client corrects DID evidence via PATCH and requests re-evaluation (see Section 4.3.1).

Table 8: Binding State Transitions

Once a binding has transitioned to active, subsequent conclusive validation failure results in transition to revoked, not rejected. The rejected state applies only to bindings that never achieved active status.

The revoked state is terminal. To re-establish trust after a binding reaches the revoked state, a client MUST create a new IdentityBinding resource; a server MUST NOT transition a revoked binding back to pending, active, or suspended.

A server MUST NOT treat unknown or missing validation results as equivalent to active.

4.3.1. Re-evaluation of Rejected Bindings

A client MAY request re-evaluation of a binding in the rejected state by submitting a SCIM PATCH request that replaces or updates one or more dids entries (for example, correcting an invalid verificationMethod, replacing an unresolvable DID value, or adding a new DID entry). Upon receiving such a PATCH, the server MUST transition the binding to pending status and MUST initiate a new validation cycle. The server MUST NOT require the client to delete and recreate the resource solely to trigger re-evaluation after a correctable rejection.

4.4. Deletion

A service provider implementing this specification MUST support DELETE for IdentityBinding resources.

Upon successful deletion of an IdentityBinding resource, the server MUST remove the corresponding entry from the associated User's bindingRefs and MUST clear User.primaryDid if the deleted binding was the source of that value. The server MUST recalculate User.bindingState after deletion.

A server MAY reject deletion of a binding in active status and require the client to first transition it to a non-active state via PATCH. If such a restriction is imposed, the server MUST document this behavior in its ServiceProviderConfig or deployment documentation.

A server MUST return HTTP 204 on successful deletion.

5. Query, Filtering, PATCH, and Pagination

5.1. Filtering

SCIM filter expressions and fully qualified attribute notation are defined by [RFC7644]. Filter attribute values in request URIs MUST be percent-encoded as required by RFC 3986; the examples in this section show decoded values for readability.

A service provider implementing this specification MUST support GET filtering on /IdentityBindings using the eq operator for the following attributes:

* user.value

* dids.value

* dids.relationship

* credentials.issuer

* credentials.types

* status

* externalId (a common attribute defined in Section 3.1 of [RFC7643] and inherited by all SCIM resource types)

A service provider implementing this specification MUST additionally support the co (contains) and pr (present) operators for the multi-valued string attributes credentials.types and dids.value. This allows clients to filter by partial VC type values (e.g., credentials.types co "EmploymentCredential") and to find resources that have at least one credential or DID present.

A service provider implementing the User extension MUST support eq filtering on urn:ietf:params:scim:schemas:extension:didvc:2.0:User:primaryDid.

A service provider implementing this specification MUST support the and logical operator in combination with the attributes above. Support for the or and not operators is OPTIONAL.

```
GET /IdentityBindings?filter=dids.value eq "did:example:123"
```

Figure 1: Example Filter by DID (values shown decoded; apply percent-encoding in actual requests)

```
GET /IdentityBindings?filter=status eq "active" and credentials.issuer eq "did:example
:issuer:acme"
```

Figure 2: Example Combined Filter

```
GET /IdentityBindings?filter=credentials.types co "EmploymentCredential"
```

Figure 3: Example Filter Using co Operator on Multi-valued types

```
GET /Users?filter=urn:ietf:params:scim:schemas:extension:didvc:2.0:User:primaryDid eq
"did:example:123"
```

Figure 4: Example User Filter by primaryDid

5.2. PATCH

A service provider implementing IdentityBinding MUST support SCIM PATCH as defined by [RFC7644]. PATCH request bodies MUST conform to the PatchOp message format defined in Section 3.5.2 of [RFC7644], which requires the schemas member and an Operations array wrapping all individual operation objects.

Clients SHOULD target explicit paths.

A server MUST return HTTP 400 with a SCIM error type of mutability when a PATCH attempts to modify an attribute declared as immutable (such as user or correlationModel) or readOnly (such as status or lastValidationAttempt).

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "dids[value eq \"did:example:123\"].primary",
      "value": true
    }
  ]
}
```

Figure 5: Example PATCH to Set a Primary DID Flag

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "credentials[issuer eq \"did:example:issuer:acme\"].schemaRef",
      "value": "https://issuer.example/schemas/employment-v1"
    }
  ]
}
```

Figure 6: Example PATCH to Update a Credential Schema Reference

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "dids[value eq \"did:example:old\"].verificationMethod",
      "value": "did:example:abc123#key-2"
    }
  ]
}
```

Figure 7: Example PATCH to Correct a DID and Trigger Re-evaluation of a Rejected Binding

5.3. Pagination

List operations for IdentityBinding resources MUST support SCIM list pagination as defined by [RFC7644]. A service provider MAY additionally support cursor-based pagination as defined by [RFC9865].

6. Error Handling

A conforming server MUST return SCIM error responses as defined in Section 3.12 of [RFC7644] for all error conditions. The following table enumerates error conditions specific to this specification.

Condition	HTTP Status	SCIM scimType
user.value references a non-existent User	400	invalidValue
Request includes more than one dids entry with primary=true	400	invalidValue
A dids entry has primary=true but relationship is not primary	400	invalidValue
Creation of a pairwise binding with a DID already used in another active pairwise binding within the same administrative scope	409	uniqueness
PATCH attempts to modify an immutable attribute (user, correlationModel)	400	mutability
PATCH attempts to modify a readOnly attribute (status, lastValidationAttempt)	400	mutability
Multiple active bindings with primary=true would result from the operation	400	invalidValue
DELETE attempted on an active binding when the server requires non-active status first	409	-

Table 9: Binding-Specific Error Conditions

7. Discovery and Deployment

A service provider implementing this specification MUST publish the User extension schema via /Schemas, MUST publish the IdentityBinding schema via /Schemas, MUST publish the IdentityBinding resource type via /ResourceTypes, and SHOULD publish the ServiceProviderConfig extension when supported. Examples of the expected /Schemas and /ResourceTypes entries are provided in Appendix C.

A SCIM client unaware of this specification will continue to interoperate with base User provisioning because unknown schema extensions and unknown resource types are discoverable rather than mandatory for unrelated operations.

8. Security Considerations

8.1. Threat Model

Relevant attackers include a client that attempts to bind a DID it does not control, a client that replays stale credentials or proofs, an attacker exploiting DID key rotation or deactivation lag, an attacker inducing false correlation by reusing identifiers across relying parties, a malicious or unexpected issuer, and a compromised or confused delegated verifier.

8.2. Required Mitigations

A conforming implementation MUST evaluate DID proof authorization against the relevant DID verification relationship at verification time, MUST re-resolve DID information when revalidating a binding after key rotation or deactivation events are possible, MUST treat credential status as a verifier-side check when status information is available, MUST integrity-protect communication with any delegated verifier and authenticate that verifier, MUST apply normal SCIM authorization controls to IdentityBinding resources, and MUST NOT treat unverified bindings as equivalent to active verified bindings.

8.3. DID Control Proof Delivery

This specification requires that a SCIM server verify DID control before activating a binding, but the mechanism by which a client presents DID control evidence is intentionally out of scope. Implementers are responsible for defining or adopting a proof transport mechanism appropriate for their deployment. Relevant existing specifications include OpenID for Verifiable Presentations (OID4VP) and the DIF Presentation Exchange specification. The SCIM service provider MUST document the proof delivery mechanism it accepts as part of its deployment documentation.

Regardless of the mechanism chosen, a server MUST NOT accept a DID control proof that was generated for a different relying party, binding resource, or session. Servers SHOULD require proof presentations to include a challenge or nonce issued by the server to prevent replay of previously valid proofs.

8.4. Replay Attack Mitigation

A server SHOULD issue a fresh, single-use nonce or challenge when initiating a DID control proof exchange and MUST reject proof presentations that do not bind to the current challenge. Similarly, credential validation results MUST be treated as bound to a specific validation timestamp and MUST NOT be replayed from a prior successful validation to satisfy a new validation request.

8.5. Delegated Verifier Trust Scope

When a SCIM service provider delegates DID or VC validation to an external verifier, the verifier becomes a critical trust anchor for the entire binding model. A compromised or misconfigured delegated verifier that returns status=verified for any DID can activate bindings without genuine cryptographic proof of control. Implementations MUST scope the trust granted to a delegated verifier to the minimum necessary (e.g., specific DID methods or credential types), MUST authenticate the verifier at the transport layer, and SHOULD audit verifier responses for anomalies such as bulk activation of previously-pending bindings.

9. Privacy Considerations

DID Core warns that globally unambiguous identifiers create correlation risk and recommends pairwise DIDs where correlation is not desired. DID Core also warns that even pairwise DIDs can be re-correlated if DID documents reuse identical verification methods or correlating service endpoints. VC Data Model 2.0 separately warns about identifier-based, signature-based, and metadata-based correlation and recommends selective disclosure or unlinkable disclosure when strong anti-correlation properties are needed [DID-CORE] [VC-DATA-MODEL-2.0].

Accordingly, an implementation conforming to this specification SHOULD prefer correlationModel=pairwise unless public reuse is intentionally desired, MUST NOT require storage of full VC payloads for routine SCIM provisioning behavior, SHOULD store only references and normalized validation outcomes, SHOULD avoid exposing DID values to clients that do not need them, SHOULD support privacy-preserving credential ecosystems that use selective disclosure or unlinkable disclosure, and SHOULD ensure that pairwise bindings do not accidentally reuse correlating DID material within the same administrative domain.

The credentials.holder and credentials.credentialSubjectId sub-attributes, while optional, present a specific privacy risk: both typically contain DID URIs that directly identify the subject, and

storing them in a SCIM resource makes the SCIM server itself a correlation point linking subject DIDs to SCIM account identifiers. This partially defeats the anti-correlation goals of pairwise DID usage. Implementations SHOULD omit these fields unless there is a specific operational requirement for them, and SHOULD apply appropriate access controls to limit which clients can read them when they are stored.

Even with pairwise DIDs and minimal storage, the SCIM service provider itself becomes a correlation point because it holds the mapping between all of a user's pairwise DIDs and their single SCIM account identifier. Access to IdentityBinding resources MUST be restricted to authorized clients, and implementations SHOULD apply data-minimization principles to SCIM query responses (e.g., using SCIM attribute projection via the attributes or excludedAttributes query parameters).

10. IANA Considerations

This document requests registration of the following URIs in the "System for Cross-domain Identity Management (SCIM) Schema URIs" registry defined by [RFC7643]:

* urn:ietf:params:scim:schemas:extension:didvc:2.0:User

* urn:ietf:params:scim:schemas:extension:didvc:2.0:IdentityBinding

* urn:ietf:params:scim:schemas:extension:didvc:2.0:ServiceProviderConfig

The registration templates appear in Appendix A.

11. References

11.1. Normative References

[DID-CORE] World Wide Web Consortium, "Decentralized Identifiers (DIDs) v1.0", W3C Recommendation did-1.0, 19 July 2022, <https://www.w3.org/TR/did-1.0/>.

[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>.

[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>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[VC-DATA-MODEL-2.0] World Wide Web Consortium, "Verifiable Credentials Data Model v2.0", W3C Recommendation vc-data-model-2.0, 15 May 2025, <https://www.w3.org/TR/vc-data-model-2.0/>.

11.2. Informative References

[RFC9865] Peterson, M., Ed., Zollner, D., and A. Sehgal, "Cursor-Based Pagination of System of Cross-domain Identity Management (SCIM) Resources", RFC 9865, DOI 10.17487/RFC9865, October 2025, <https://www.rfc-editor.org/info/rfc9865>.

Appendix A. SCIM Schema URI Registration Templates

A.1. User Extension

Schema URI: urn:ietf:params:scim:schemas:extension:didvc:2.0:User

Schema Name: DID and VC User Extension

Intended or Associated Resource Type: User

Purpose: Read-only discovery of DID and VC binding state for a SCIM user.

Single-value Attributes: primaryDid, bindingState.

Multi-valued Attributes: bindingRefs.

A.2. IdentityBinding

Schema URI: urn:ietf:params:scim:schemas:extension:didvc:2.0:IdentityBinding

Schema Name: IdentityBinding

Intended or Associated Resource Type: IdentityBinding

Purpose: Mutable binding resource linking a SCIM User to one or more DIDs and optional credential references.

Single-value Attributes: user, correlationModel, status, lastValidationAttempt, and statusReason.

Multi-valued Attributes: dids and credentials.

A.3. ServiceProviderConfig Extension

Schema URI: urn:ietf:params:scim:schemas:extension:didvc:2.0:ServiceProviderConfig

Schema Name: DID and VC ServiceProviderConfig Extension

Intended or Associated Resource Type: ServiceProviderConfig

Purpose: Capability discovery for DID and VC binding support.

Single-value Attributes: enabled and verificationDelegation.

Multi-valued Attributes: supportedDidMethods, supportedCredentialChecks, and supportedCorrelationModels.

Appendix B. Examples

B.1. Example User Representation

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:didvc:2.0:User"
  ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "alice@example.com",
  "name": {
    "givenName": "Alice",
    "familyName": "Ng"
  },
  "urn:ietf:params:scim:schemas:extension:didvc:2.0:User": {
    "primaryDid": "did:example:abc123",
    "bindingState": "active",
    "bindingRefs": [
      {
        "value": "c3a801b0-0b28-4c9c-a7fb-9964a6d916c1",
        "$ref": "/IdentityBindings/c3a801b0-0b28-4c9c-a7fb-9964a6d916c1",
        "display": "Primary workforce binding",
        "primary": true,
        "status": "active"
      }
    ]
  }
}
```

Figure 8

B.2. Example IdentityBinding Representation

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:extension:didvc:2.0:IdentityBinding"
  ],
  "id": "c3a801b0-0b28-4c9c-a7fb-9964a6d916c1",
  "meta": {
    "resourceType": "IdentityBinding",
    "created": "2026-04-20T09:00:00Z",
    "lastModified": "2026-04-21T14:37:00Z",
    "location": "/IdentityBindings/c3a801b0-0b28-4c9c-a7fb-9964a6d916c1",
    "version": "W/\"3694e05afa83191e\""
  },
  "user": {
    "value": "2819c223-7f76-453a-919d-413861904646",
    "$ref": "/Users/2819c223-7f76-453a-919d-413861904646",
    "display": "Alice Ng"
  },
  "correlationModel": "pairwise",
  "dids": [
    {
      "value": "did:example:abc123",
      "relationship": "primary",
      "verificationMethod": "did:example:abc123#key-1",
      "proofPurpose": "authentication",
      "controller": "did:example:abc123",
      "primary": true,
      "status": "verified"
    }
  ],
  "credentials": [
    {
      "credentialId": "urn:uuid:4dd2dc49-9c0f-43f8-b96e-4f4d1f578a2d",
      "types": ["VerifiableCredential", "EmploymentCredential"],
      "issuer": "did:example:issuer:acme",
      "statusRef": "https://issuer.example/status/94567",
      "schemaRef": "https://issuer.example/schema/employment-v1",
      "validFrom": "2026-04-01T00:00:00Z",
      "validUntil": "2027-04-01T00:00:00Z",
      "status": "active"
    }
  ],
  "status": "active",
  "lastValidationAttempt": "2026-04-21T14:37:00Z",
  "statusReason": "DID control and required credential checks succeeded"
}
```


Figure 9

B.3. Example ServiceProviderConfig Representation

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig",
    "urn:ietf:params:scim:schemas:extension:didvc:2.0:ServiceProviderConfig"
  ],
  "patch": { "supported": true },
  "filter": { "supported": true, "maxResults": 200 },
  "sort": { "supported": false },
  "etag": { "supported": true },
  "urn:ietf:params:scim:schemas:extension:didvc:2.0:ServiceProviderConfig": {
    "enabled": true,
    "verificationDelegation": "mixed",
    "supportedDidMethods": ["did:web", "did:key", "did:example"],
    "supportedCredentialChecks": ["status", "schema", "validity", "holderBinding"],
    "supportedCorrelationModels": ["pairwise", "shared", "public"]
  }
}
```

Figure 10

Appendix C. Discovery Examples

C.1. Example /ResourceTypes Entry for IdentityBinding

The following illustrates the ResourceType entry a conforming server MUST publish at /ResourceTypes for the IdentityBinding resource.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
  "id": "IdentityBinding",
  "name": "IdentityBinding",
  "endpoint": "/IdentityBindings",
  "description": "Auditable linkage between a SCIM User and one or more DIDs and crede
ntial references.",
  "schema": "urn:ietf:params:scim:schemas:extension:didvc:2.0:IdentityBinding",
  "meta": {
    "resourceType": "ResourceType",
    "location": "/ResourceTypes/IdentityBinding"
  }
}
```

Figure 11

C.2. Example /Schemas Entry for the User Extension

The following illustrates the Schema entry a conforming server MUST publish at /Schemas for the User extension defined by this specification. The entry for the IdentityBinding schema and the ServiceProviderConfig extension schema follow the same pattern.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Schema"],
  "id": "urn:ietf:params:scim:schemas:extension:didvc:2.0:User",
  "name": "DID and VC User Extension",
  "description": "Read-only discovery of DID and VC binding state for a SCIM user.",
  "attributes": [
    {
      "name": "primaryDid",
      "type": "string",
      "multiValued": false,
      "required": false,
      "mutability": "readOnly",
      "returned": "default"
    },
    {
      "name": "bindingState",
      "type": "string",
      "multiValued": false,
      "required": true,
      "canonicalValues": ["none", "pending", "active", "suspended", "revoked", "reject
ed"],
      "mutability": "readOnly",
      "returned": "default"
    },
    {
      "name": "bindingRefs",
      "type": "complex",
      "multiValued": true,
      "required": false,
      "mutability": "readOnly",
      "returned": "default",
      "subAttributes": [
        { "name": "value", "type": "string", "required": true, "mutability": "readOnly
" },
        { "name": "$ref", "type": "reference", "required": true, "mutability": "readOn
ly" },
        { "name": "display", "type": "string", "required": false, "mutability": "readO
nly" },
        { "name": "primary", "type": "boolean", "required": false, "mutability": "read
Only" },
        {
          "name": "status",
          "type": "string",
          "required": true,
          "canonicalValues": ["pending", "active", "suspended", "revoked", "rejected"]
,
          "mutability": "readOnly"
```

```
            }
          ]
        }
      ],
      "meta": {
        "resourceType": "Schema",
        "location": "/Schemas/urn:ietf:params:scim:schemas:extension:didvc:2.0:User"
      }
    }
```

Figure 12

Acknowledgments

The author thanks contributors and reviewers whose feedback will be acknowledged in future revisions of this document.

Author’s Address

Saurabh Kushwaha
Oracle Corporation
Pleasanton, CA
United States of America
Email: saurabh.kushwaha@oracle.com