

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 03, 2014

C. Mortimore, Ed.
Salesforce
P. Harding
P. Madsen
Ping
T. Drake
UnboundID
August 30, 2013

System for Cross-Domain Identity Management: Core Schema
draft-ietf-scim-core-schema-02

Abstract

The System for Cross-Domain Identity Management (SCIM) specification is designed to make managing user identity in cloud based applications and services easier. The specification suite builds upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. Its intent is to reduce the cost and complexity of user management operations by providing a common user schema and extension model, as well as binding documents to provide patterns for exchanging this schema using standard protocols. In essence, make it fast, cheap, and easy to move identity in to, out of, and around the cloud.

This document provides a platform neutral schema and extension model for representing users and groups in JSON format. This schema is intended for exchange and use with cloud service providers. Additional binding documents provide a standard REST API, SAML binding, and use cases.

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 <http://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 March 03, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Notation and Conventions	3
2. Overview	3
2.1. Definitions	4
3. SCIM Schema Structure	4
3.1. Attribute Data Types	5
3.1.1. String	5
3.1.2. Boolean	5
3.1.3. Decimal	5
3.1.4. Integer	5
3.1.5. DateTime	6
3.1.6. Binary	6
3.1.7. Reference	6
3.1.8. Complex	6
3.2. Multi-valued Attributes	6
4. Schema Extension Model	8
5. SCIM Core Schema	8
5.1. Common Schema Attributes	8
5.2. "schemas" Attribute	9
6. SCIM User Schema	10
6.1. Singular Attributes	10
6.2. Multi-valued Attributes	12
7. SCIM Enterprise User Schema Extension	14
8. SCIM Group Schema	15
9. Service Provider Configuration Schema	15
10. Resource Type Schema	17
11. Schema Schema	18
12. JSON Representation	20
12.1. Minimal User Representation	20
12.2. Full User Representation	20

12.3.	Enterprise User Extension Representation	23
12.4.	Group Representation	26
12.5.	Service Provider Configuration Representation	27
12.6.	Resource Type Representation	28
12.7.	Schema Representation	29
13.	Security Considerations	33
14.	Normative References	33
Appendix A.	Contributors	34
Authors' Addresses	34

1. Requirements Notation and Conventions

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] .

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

2. Overview

While there are existing standards for describing and exchanging user information, many of these standards can be difficult to implement and/or use; e.g., their wire protocols do not easily traverse firewalls and/or are not easily layered onto existing web protocols. As a result, many cloud providers implement non-standard APIs for managing users within their services. This increases both the cost and complexity associated with organizations adopting products and services from multiple cloud providers as they must perform redundant integration development. Similarly, cloud services providers seeking to interoperate with multiple application marketplaces or cloud identity providers must be redundantly integrated.

SCIM seeks to simplify this problem through a simple to implement specification suite that provides a common user schema and extension model, as well as binding documents to provide patterns for exchanging this schema via a REST API. It draws inspiration and best practice, building upon existing user APIs and schemas from a wide variety of sources including, but not limited to, existing APIs exposed by cloud providers, PortableContacts, and LDAP directory services.

This document provides a platform neutral schema and extension model for representing users and groups in JSON format. This schema is intended for exchange and use with cloud service providers. Additional binding documents provide a standard REST API, SAML binding, and use cases.

2.1. Definitions

Service Provider: A web application that provides identity information via the SCIM protocol.

Consumer: A website or application that uses the SCIM protocol to manage identity data maintained by the Service Provider.

Resource: The Service Provider managed artifact containing one or more attributes; e.g., User or Group

Resource Type: A type of a Resource that is managed by a Service Provider. The Resource Type defines the Resource name, endpoint URL, Schemas, and other meta-data which indicate where a Resource is managed and how it is composed; e.g., User or Group.

Schema: A collection of Attribute Definitions that describe the contents of an entire or partial Resource; e.g., urn:scim:schemas:core:User:2.0.

Singular Attribute: A Resource attribute that contains 0..1 values; e.g., displayName.

Multi-valued Attribute: A Resource attribute that contains 0..n values; e.g., emails.

Simple Attribute: A Singular or Multi-valued Attribute whose value is a primitive; e.g., String.

Complex Attribute: A Singular or Multi-valued Attribute whose value is a composition of one or more Simple Attributes.

Sub-Attribute: A Simple Attribute contained within a Complex Attribute.

3. SCIM Schema Structure

SCIM schema provides a minimal core schema for representing users and groups (resources), encompassing common attributes found in many existing deployments and schemas.

A resource is a collection of attributes identified by one or more schemas. Minimally, an attribute consists of the attribute name and at least one Simple or Complex value either of which may be Multi-valued. SCIM schema defines the data type, plurality and other distinguishing features of an attribute. Unless otherwise specified all attributes are modifiable by Consumers. Immutable (read-only) attributes SHALL be specified as 'READ-ONLY' within the attribute

definition. Additionally, Service Providers MAY choose to make some or all Resource attributes immutable and SHOULD identify those attributes via the associated Resource's schema endpoint (Section 5.2).

A JSON [1] (JavaScript Object Notation) format is defined. Attribute names SHOULD be camelCased. SCIM resources represented in JSON MUST specify schema via the schemas attribute (Section 5.2).

3.1. Attribute Data Types

Attribute data types are derived from JSON [2] and unless otherwise specified are optional, modifiable by Consumers, and of type String (Section 3.1.1). The JSON format defines a limited set of data types, hence, where appropriate, alternate JSON representations derived from XML schema [3] are defined below. SCIM extensions SHOULD not introduce new data types.

3.1.1. String

A sequence of zero or more Unicode characters. The JSON format is defined in section 2.5 [4] of RFC 4627. A String attribute MAY specify a required data format. Additionally, when Canonical Values are specified Service Providers SHOULD conform to those values if appropriate, but MAY provide alternate String values to represent additional values.

3.1.2. Boolean

The literal "true" or "false". The JSON format is defined in section 2.1 [5] of RFC 4627.

3.1.3. Decimal

A real number with at least one digit to the left and right of the period. The JSON format is defined in section 2.4 [6] of RFC 4627.

3.1.4. Integer

A Decimal number with no fractional digits. The JSON format is defined in section 2.4 [7] of RFC 4627 with the additional constraint that the value MUST NOT contain fractional or exponent parts.

3.1.5. DateTime

A DateTime value (e.g. 2008-01-23T04:56:22Z). The attribute value MUST be encoded as a valid `xsd:dateTime` as specified in section 3.2.7 [8] of the XML Schema Datatypes Specification.

Values represented in JSON MUST conform to the XML constraints above and are represented as a JSON String [9].

3.1.6. Binary

Arbitrary binary data. The attribute value MUST be encoded as a valid `xsd:base64Binary` as specified in section 3.2.16 [10] of the XML Schema Datatypes Specification.

Values represented in JSON MUST conform to the XML constraints above and are represented as a JSON String [11].

3.1.7. Reference

A reference to a SCIM Resource. The value MUST be the absolute or relative URI of the target Resource. Relative URIs should be resolved as specified in section 5.2 [12] of RFC 3986. The base URI for relative URI resolution MUST include all URI components and path segments up to but not including the Endpoint URI; e.g., the base URI for a request to `https://example.com/v1/Users/2819c223-7f76-453a-919d-413861904646` would be `https://example.com/v1/` and the relative URI for this Resource would be `Users/2819c223-7f76-453a-919d-413861904646`.

Performing a GET operation on a reference URI MUST return the target Resource or an appropriate HTTP response code. The Service Provider MAY optionally choose to enforce referential integrity for references.

By convention, a reference is commonly represented as a "\$ref" sub-attribute in complex or multi-valued attributes, however this is OPTIONAL.

3.1.8. Complex

A Singular or Multi-valued Attribute whose value is a composition of one or more Simple Attributes. The JSON format is defined in section 2.2 [13] of RFC 4627.

3.2. Multi-valued Attributes

Multi-valued attributes are unordered lists of attributes. Each attribute MAY contain Sub-Attributes and therefore multi-valued attributes may contain Complex Attributes. The below Sub-Attributes are considered normative and when specified SHOULD be used as defined.

type A label indicating the attribute's function; e.g., "work" or "home".

primary A Boolean value indicating the 'primary' or preferred attribute value for this attribute, e.g. the preferred mailing address or primary e-mail address. The primary attribute value 'true' MUST appear no more than once.

display A human readable name, primarily used for display purposes. READ-ONLY.

operation The operation to perform on the multi-valued attribute during a PATCH request. The only valid value is "delete", which signifies that this instance should be removed from the Resource.

value The attribute's significant value; e.g., the e-mail address, phone number, etc. Attributes that define a "value" sub-attribute MAY be alternately represented as a collection of primitive types. For example:

```
{
  "emails": [
    {"value": "bjensen@example.com"},
    {"value": "babs@example.com"}
  ]
}
```

May also be represented as:

```
{
  "emails": ["bjensen@example.com", "babs@example.com"]
}
```

\$ref The Reference of the target Resource, if the attribute is a reference.

When returning multi-valued attributes, Service Providers SHOULD canonicalize the value returned, if appropriate (e.g. for e-mail addresses and URLs). Providers MAY return the same value more than

once with different types (e.g. the same e-mail address may be used for work and home), but SHOULD NOT return the same (type, value) combination more than once per Attribute, as this complicates processing by the Consumer.

4. Schema Extension Model

SCIM schema follows an object extension model similar to ObjectClasses used in LDAP. Unlike LDAP there is no inheritance model; all extensions are additive (similar to LDAP Auxiliary Object Classes [14]). Each value indicates additive schema that may exist in a SCIM representation as specified by extensions not defined in this suite. Schema extensions MUST NOT redefine any attributes defined in this specification and SHOULD follow conventions defined in this specification. Each schema extension must identify a URI used to identify the extension. The JSON format MUST use the "schemas" attribute (Section 5.2) to distinguish extended resources and attributes.

5. SCIM Core Schema

5.1. Common Schema Attributes

Each SCIM Resource (Users, Groups, etc.) includes the below common attributes. These attributes MUST be included in all Resources, including any extended Resource types. It is not necessary to specify the schemas attribute if the Resource is fully defined in this document as the core schema is implicitly included.

id Unique identifier for the SCIM Resource as defined by the Service Provider. Each representation of the Resource MUST include a non-empty id value. This identifier MUST be unique across the Service Provider's entire set of Resources. It MUST be a stable, non-reassignable identifier that does not change when the same Resource is returned in subsequent requests. The value of the id attribute is always issued by the Service Provider and MUST never be specified by the Service Consumer. **bulkId:** is a reserved keyword and MUST NOT be used in the unique identifier. REQUIRED and READ-ONLY.

externalId An identifier for the Resource as defined by the Service Consumer. The externalId may simplify identification of the Resource between Service Consumer and Service provider by allowing the Consumer to refer to the Resource with its own identifier, obviating the need to store a local mapping between the local identifier of the Resource and the identifier used by the Service Provider. Each Resource MAY include a non-empty externalId value. The value of the externalId attribute is always issued by the

Service Consumer and can never be specified by the Service Provider. The Service Provider MUST always interpret the externalId as scoped to the Service Consumer's tenant.

meta A complex attribute containing resource metadata. All sub-attributes are OPTIONAL

resourceType The name of the Resource Type of the Resource.
READ-ONLY.

created The DateTime the Resource was added to the Service Provider. The attribute MUST be a DateTime. READ-ONLY.

lastModified The most recent DateTime the details of this Resource were updated at the Service Provider. If this Resource has never been modified since its initial creation, the value MUST be the same as the value of created. The attribute MUST be a DateTime. READ-ONLY.

location The URI of the Resource being returned. This value MUST be the same as the Location HTTP response header. READ-ONLY.

version The version of the Resource being returned. This value must be the same as the ETag HTTP response header. READ-ONLY.

attributes The names of the attributes to remove from the Resource during a PATCH operation.

5.2. "schemas" Attribute

SCIM supports resources of different types, with extensible schemas. Each resource MUST be indicated using fully qualified URLs.

When a representation does not explicitly provide support for indicating a schema, such as JSON, a schemas attribute is used to indicate the version of SCIM schema as well as any schema extensions.

schemas The schemas attribute is an array of Strings which allows introspection of the supported schema version for a SCIM representation as well any schema extensions supported by that representation. Each String value must be a unique URI. This specification defines URIs for User, Group, and a standard "enterprise" extension. All representations of SCIM schema MUST include a non-zero value array with value(s) of the URIs supported by that representation. The schemas attribute for a Resource MUST only contain values defined as "schema" and "schemaExtensions" for

the Resource's Resource Type. Duplicate values MUST NOT be included. Value order is not specified and MUST not impact behavior. REQUIRED.

6. SCIM User Schema

SCIM provides a schema for representing Users, identified using the following URI: 'urn:scim:schemas:core:2.0:User'. The following attributes are defined in addition to those attributes defined in SCIM Core Schema:

6.1. Singular Attributes

userName Unique identifier for the User, typically used by the user to directly authenticate to the service provider. Often displayed to the user as their unique identifier within the system (as opposed to `id` or `externalId`, which are generally opaque and not user-friendly identifiers). Each User MUST include a non-empty `userName` value. This identifier MUST be unique across the Service Consumer's entire set of Users. REQUIRED.

name The components of the User's real name. Providers MAY return just the full name as a single string in the formatted sub-attribute, or they MAY return just the individual component attributes using the other sub-attributes, or they MAY return both. If both variants are returned, they SHOULD be describing the same name, with the formatted name indicating how the component attributes should be combined.

formatted The full name, including all middle names, titles, and suffixes as appropriate, formatted for display (e.g. Ms. Barbara Jane Jensen, III.).

familyName The family name of the User, or "Last Name" in most Western languages (e.g. Jensen given the full name Ms. Barbara Jane Jensen, III.).

givenName The given name of the User, or "First Name" in most Western languages (e.g. Barbara given the full name Ms. Barbara Jane Jensen, III.).

middleName The middle name(s) of the User (e.g. Jane given the full name Ms. Barbara Jane Jensen, III.).

honorificPrefix The honorific prefix(es) of the User, or "Title" in most Western languages (e.g. Ms. given the full name Ms. Barbara Jane Jensen, III.).

honorificSuffix The honorific suffix(es) of the User, or "Suffix" in most Western languages (e.g. III. given the full name Ms. Barbara Jane Jensen, III.).

displayName The name of the User, suitable for display to end-users. Each User returned MAY include a non-empty displayName value. The name SHOULD be the full name of the User being described if known (e.g. Babs Jensen or Ms. Barbara J Jensen, III), but MAY be a username or handle, if that is all that is available (e.g. bjensen). The value provided SHOULD be the primary textual label by which this User is normally displayed by the Service Provider when presenting it to end-users.

nickName The casual way to address the user in real life, e.g. "Bob" or "Bobby" instead of "Robert". This attribute SHOULD NOT be used to represent a User's username (e.g. bjensen or mpepperidge).

profileUrl A fully qualified URL to a page representing the User's online profile.

title The user's title, such as "Vice President."

userType Used to identify the organization to user relationship. Typical values used might be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown" but any value may be used.

preferredLanguage Indicates the User's preferred written or spoken language. Generally used for selecting a localized User interface. Valid values are concatenation of the ISO 639-1 two letter language code [15], an underscore, and the ISO 3166-1 2 letter country code [16]; e.g., 'en_US' specifies the language English and country US.

locale Used to indicate the User's default location for purposes of localizing items such as currency, date time format, numerical representations, etc. A locale value is a concatenation of the ISO 639-1 two letter language code [17], an underscore, and the ISO 3166-1 2 letter country code [18]; e.g., 'en_US' specifies the language English and country US.

timezone The User's time zone in the "Olson" timezone database format [19]; e.g., 'America/Los_Angeles'.

active A Boolean value indicating the User's administrative status. The definitive meaning of this attribute is determined by the Service Provider though a value of true infers the User is, for example, able to login while a value of false implies the User's account has been suspended.

password The User's clear text password. This attribute is intended to be used as a means to specify an initial password when creating a new User or to reset an existing User's password. No accepted standards exist to convey password policies, hence Consumers should expect Service Providers to reject password values. This value **MUST** never be returned by a Service Provider in any form.

6.2. Multi-valued Attributes

The following multi-valued attributes are defined.

emails E-mail addresses for the User. The value **SHOULD** be canonicalized by the Service Provider, e.g. `bjensen@example.com` instead of `bjensen@EXAMPLE.COM`. Canonical Type values of `work`, `home`, and `other`.

phoneNumbers Phone numbers for the User. The value **SHOULD** be canonicalized by the Service Provider according to format in RFC3966 [20] e.g. `'tel:+1-201-555-0123'`. Canonical Type values of `work`, `home`, `mobile`, `fax`, `pager` and `other`.

ims Instant messaging address for the User. No official canonicalization rules exist for all instant messaging addresses, but Service Providers **SHOULD**, when appropriate, remove all whitespace and convert the address to lowercase. Instead of the standard Canonical Values for `type`, this attribute defines the following Canonical Values to represent currently popular IM services: `aim`, `gtalk`, `icq`, `xmpp`, `msn`, `skype`, `qq`, and `yahoo`.

photos URL of a photo of the User. The value **SHOULD** be a canonicalized URL, and **MUST** point to an image file (e.g. a GIF, JPEG, or PNG image file) rather than to a web page containing an image. Service Providers **MAY** return the same image at different sizes, though it is recognized that no standard for describing images of various sizes currently exists. Note that this attribute **SHOULD NOT** be used to send down arbitrary photos taken by this User, but specifically profile photos of the User suitable for display when describing the User. Instead of the standard Canonical Values for `type`, this attribute defines the following Canonical Values to represent popular photo sizes: `photo`, `thumbnail`.

addresses A physical mailing address for this User. Canonical Type Values of `work`, `home`, and `other`. The value attribute is a complex type with the following sub-attributes. All Sub-Attributes are **OPTIONAL**.

formatted The full mailing address, formatted for display or use with a mailing label. This attribute MAY contain newlines.

streetAddress The full street address component, which may include house number, street name, P.O. box, and multi-line extended street address information. This attribute MAY contain newlines.

locality The city or locality component.

region The state or region component.

postalCode The zipcode or postal code component.

country The country name component. When specified the value MUST be in ISO 3166-1 alpha 2 "short" code format [21]; e.g., the United States and Sweden are "US" and "SE", respectively.

groups A list of groups that the user belongs to, either thorough direct membership, nested groups, or dynamically calculated. The values are meant to enable expression of common group or role based access control models, although no explicit authorization model is defined. It is intended that the semantics of group membership and any behavior or authorization granted as a result of membership are defined by the Service Provider. The Canonical types "direct" and "indirect" are defined to describe how the group membership was derived. Direct group membership indicates the User is directly associated with the group and SHOULD indicate that Consumers may modify membership through the Group Resource. Indirect membership indicates User membership is transitive or dynamic and implies that Consumers cannot modify indirect group membership through the Group resource but MAY modify direct group membership through the Group resource which MAY influence indirect memberships. If the SCIM Service Provider exposes a Group resource, the "value" sub-attribute MUST be the "id" and the "\$ref" sub-attribute must be the URI of the corresponding Group resources to which the user belongs. Since this attribute is read-only, group membership changes MUST be applied via the Group Resource (Section 8). READ-ONLY.

entitlements A list of entitlements for the User that represent a thing the User has. That is, an entitlement is an additional right to a thing, object or service. No vocabulary or syntax is specified and Service Providers/Consumers are expected to encode sufficient information in the value so as to accurately and without ambiguity determine what the User has access to. This value has NO canonical types though type may be useful as a means to scope entitlements.

roles A list of roles for the User that collectively represent who the User is; e.g., 'Student', "Faculty". No vocabulary or syntax is specified though it is expected that a role value is a String or label representing a collection of entitlements. This value has NO canonical types.

x509Certificates A list of certificates issued to the User. Values are Binary (Section 3.1.6) and DER encoded x509. This value has NO canonical types.

7. SCIM Enterprise User Schema Extension

The following SCIM extension defines attributes commonly used in representing users that belong to, or act on behalf of a business or enterprise. The enterprise user extension is identified using the following URI: 'urn:scim:schemas:extension:enterprise:2.0:User'.

The following Singular Attributes are defined:

employeeNumber Numeric or alphanumeric identifier assigned to a person, typically based on order of hire or association with an organization.

costCenter Identifies the name of a cost center.

organization Identifies the name of an organization.

division Identifies the name of a division.

department Identifies the name of a department.

manager The User's manager. A complex type that optionally allows Service Providers to represent organizational hierarchy by referencing the "id" attribute of another User.

managerId The id of the SCIM resource representing the User's manager. REQUIRED.

`$ref` The URI of the SCIM resource representing the User's manager. REQUIRED.

`displayName` The `displayName` of the User's manager. OPTIONAL and READ-ONLY.

8. SCIM Group Schema

SCIM provides a schema for representing groups, identified using the following URI: `'urn:scim:schemas:core:2.0:Group'`.

Group resources are meant to enable expression of common Group or role based access control models, although no explicit authorization model is defined. It is intended that the semantics of group membership and any behavior or authorization granted as a result of membership are defined by the Service Provider and are considered out of scope for this specification.

The following Singular Attribute is defined in addition to the common attributes defined in SCIM Core Schema:

`displayName` A human readable name for the Group. REQUIRED.

The following multi-valued attribute is defined in addition to the common attributes defined in SCIM Core Schema:

`members` A list of members of the Group. Canonical Types "User" and "Group" are READ-ONLY. The "value" sub-attribute must be the "id" and the "\$ref" sub-attribute must be the URI of a SCIM resource, either a User, or a Group. The intention of the Group type is to allow the Service Provider to support nested Groups. Service Providers MAY require Consumers to provide a non-empty members value based on the "required" sub attribute of the "members" attribute in Group Resource Schema.

9. Service Provider Configuration Schema

SCIM provides a schema for representing the Service Provider's configuration identified using the following URI:
`'urn:scim:schemas:core:2.0:ServiceProviderConfig'`

The Service Provider Configuration Resource enables a Service Provider to expose its compliance with the SCIM specification in a standardized form as well as provide additional implementation details to Consumers. All attributes are READ-ONLY. Unlike other core Resources, the "id" attribute is not required for the Service Provider Configuration Resource.

The following Singular Attributes are defined in addition to the common attributes defined in Core Schema:

`documentationUrl` An HTTP addressable URL pointing to the Service Provider's human consumable help documentation.

`patch` A complex type that specifies PATCH configuration options. REQUIRED.

`supported` Boolean value specifying whether the operation is supported. REQUIRED.

`bulk` A complex type that specifies BULK configuration options. REQUIRED

`supported` Boolean value specifying whether the operation is supported. REQUIRED.

`maxOperations` An integer value specifying the maximum number of operations. REQUIRED.

`maxPayloadSize` An integer value specifying the maximum payload size in bytes. REQUIRED.

`filter` A complex type that specifies FILTER options. REQUIRED.

`supported` Boolean value specifying whether the operation is supported. REQUIRED.

`maxResults` Integer value specifying the maximum number of Resources returned in a response. REQUIRED.

`changePassword` A complex type that specifies Change Password configuration options. REQUIRED.

`supported` Boolean value specifying whether the operation is supported. REQUIRED.

`sort` A complex type that specifies Sort configuration options. REQUIRED.

`supported` Boolean value specifying whether sorting is supported. REQUIRED.

`etag` A complex type that specifies Etag configuration options. REQUIRED.

supported Boolean value specifying whether the operation is supported. REQUIRED.

The following multi-valued attribute is defined in addition to the common attributes defined in Core Schema:

authenticationSchemes A complex type that specifies supported Authentication Scheme properties. Instead of the standard Canonical Values for type, this attribute defines the following Canonical Values to represent common schemes: oauth, oauth2, oauthbearer token, httpbasic, and httpdigest. To enable seamless discovery of configuration, the Service Provider SHOULD, with the appropriate security considerations, make the authenticationSchemes attribute publicly accessible without prior authentication. REQUIRED.

name The common authentication scheme name; e.g., HTTP Basic. REQUIRED.

description A description of the Authentication Scheme. REQUIRED.

specUrl A HTTP addressable URL pointing to the Authentication Scheme's specification. OPTIONAL.

documentationUrl A HTTP addressable URL pointing to the Authentication Scheme's usage documentation. OPTIONAL.

10. Resource Type Schema

The Resource Type schema specifies the meta-data about a Resource Type. Resource Type Resources are READ-ONLY and identified using the following URI: 'urn:scim:schemas:core:2.0:ResourceType'. Unlike other core Resources, all Attributes are REQUIRED unless otherwise specified, and the "id" attribute is not required for the Resource Type Resource.

The following Singular Attributes are defined:

name The Resource Type name. When applicable Service Providers MUST specify the name specified in the core schema specification; e.g., "User" or "Group". This name is referenced by the meta.resourceType attribute in all resources.

description The Resource Type's human readable description. When applicable Service Providers MUST specify the description specified in the core schema specification.

endpoint The Resource Type's HTTP addressable endpoint relative to the Base URL; e.g., /Users.

schema The Resource Type's primary schema URI; e.g., urn:scim:schemas:core:2.0:User. This MUST be equal to the "id" attribute of the Schema.

schemaExtensions A list of URIs of the Resource Type's schema extensions. OPTIONAL.

schema The URI of an extended schema; e.g., urn:edu:2.0:Staff. This MUST be equal to the "id" attribute of a Schema. REQUIRED.

required A Boolean value that specifies whether the schema extension is required for the Resource Type. If true, a Resource of this type MUST include this schema extension and include any attributes declared as required in this schema extension. If false, a Resource of this type MAY omit this schema extension. REQUIRED.

11. Schema Schema

The Schema schema specifies the Attribute(s) and meta-data that constitute a Schema. Schema Resources are READ-ONLY and identified using the following URI: 'urn:scim:schemas:core:2.0:Schema'. Unlike other core Resources the Schema Resource MAY contain a complex object within a Sub-Attribute and all Attributes are REQUIRED unless other specified.

The following Singular Attributes are defined:

id The unique URI of the schema. When applicable Service Providers MUST specify the URI specified in the core schema specification; e.g., "urn:scim:core:2.0:User". Unlike most other schemas, which use some sort of a GUID for the "id", the Schema "id" is a URI so that it can be registered and is portable between different Service Providers and Clients.

name The Schema's human readable name. When applicable Service Providers MUST specify the name specified in the core schema specification; e.g., "User" or "Group". OPTIONAL.

description The Schema's human readable description. When applicable Service Providers MUST specify the description specified in the core schema specification. OPTIONAL.

The following multi-valued attribute is defined:

attributes A complex type that specifies the set of Resource attributes.

name The attribute's name.

type The attribute's data type; e.g., String.

multiValued Boolean value indicating the attribute's plurality.

description The attribute's human readable description. When applicable Service Providers MUST specify the description specified in the core schema specification.

readOnly A Boolean value that specifies if the attribute is mutable.

required A Boolean value that specifies if the attribute is required.

caseExact A Boolean value that specifies if the String attribute is case sensitive.

referenceTypes The names of the Resource Types that may be referenced; e.g., User. This is only applicable for attributes that are of the "reference" data type.

The following multi-valued attributes are defined. There are no canonical type values defined and the primary value serves no useful purpose.

subAttributes A list specifying the contained attributes.
OPTIONAL.

name The attribute's name.

type The attribute's data type; e.g., String.

description The attribute's human readable description. When applicable Service Providers MUST specify the description specified in the core schema specification.

readOnly A Boolean value that specifies if the attribute is mutable.

required	A Boolean value that specifies if the attribute is required.
caseExact	A Boolean value that specifies if the String attribute is case sensitive.
referenceTypes	The names of the Resource Types that may be referenced; e.g., User. This is only applicable for attributes that are of the "reference" data type.
canonicalValues	A collection of canonical values. When applicable Service Providers MUST specify the canonical types specified in the core schema specification; e.g., "work", "home". OPTIONAL.

12. JSON Representation

12.1. Minimal User Representation

The following is a non-normative example of the minimal required SCIM representation in JSON format.

```
{
  "schemas": ["urn:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen@example.com"
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/"3694e05e9dff590\"",
    "location": "https://example.com/v1/Users/2819c223-7f76-453a-919d-413
861904646"
  }
}
```

12.2. Full User Representation

The following is a non-normative example of the fully populated SCIM representation in JSON format.

```
{
  "schemas": ["urn:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
```

```
"externalId": "701984",
"userName": "bjensen@example.com",
"name": {
  "formatted": "Ms. Barbara J Jensen III",
  "familyName": "Jensen",
  "givenName": "Barbara",
  "middleName": "Jane",
  "honorificPrefix": "Ms.",
  "honorificSuffix": "III"
},
"displayname": "Babs Jensen",
"nickname": "Babs",
"profileUrl": "https://login.example.com/bjensen",
"emails": [
  {
    "value": "bjensen@example.com",
    "type": "work",
    "primary": true
  },
  {
    "value": "babs@jensen.org",
    "type": "home"
  }
],
"addresses": [
  {
    "type": "work",
    "streetAddress": "100 Universal City Plaza",
    "locality": "Hollywood",
    "region": "CA",
    "postalCode": "91608",
    "country": "USA",
    "formatted": "100 Universal City Plaza\nHollywood, CA 91608 USA",
    "primary": true
  },
  {
    "type": "home",
    "streetAddress": "456 Hollywood Blvd",
    "locality": "Hollywood",
    "region": "CA",
    "postalCode": "91608",
    "country": "USA",
    "formatted": "456 Hollywood Blvd\nHollywood, CA 91608 USA"
  }
],
"phoneNumbers": [
  {
    "value": "555-555-5555",
```

```
    "type": "work"
  },
  {
    "value": "555-555-4444",
    "type": "mobile"
  }
],
"ims": [
  {
    "value": "someaimhandle",
    "type": "aim"
  }
],
"photos": [
  {
    "value": "https://photos.example.com/profilephoto/7293000000Ccne/F",
    "type": "photo"
  },
  {
    "value": "https://photos.example.com/profilephoto/7293000000Ccne/T",
    "type": "thumbnail"
  }
],
"userType": "Employee",
"title": "Tour Guide",
"preferredLanguage": "en_US",
"locale": "en_US",
"timezone": "America/Los_Angeles",
"active": true,
"password": "tlmeMa$heen",
"groups": [
  {
    "value": "e9e30dba-f08f-4109-8486-d5c6a331660a",
    "$ref": "https://example.com/v1/Groups/e9e30dba-f08f-4109-8486-d5c6a331660a",
    "display": "Tour Guides"
  },
  {
    "value": "fc348aa8-3835-40eb-a20b-c726e15c55b5",
    "$ref": "https://example.com/v1/Groups/fc348aa8-3835-40eb-a20b-c726e15c55b5",
    "display": "Employees"
  },
  {
    "value": "71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
    "$ref": "https://example.com/v1/Groups/71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
    "display": "US Employees"
  }
],
"x509Certificates": [
```



```

    {
      "value": "MIIDQzCCAqygAwIBAgICEAAwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEB
hMCVVMx
RQwEgYD
jI0MzFa
QQKDatl
UlJMSIw
kiG9w0B
6AK02bc
PKpzz5i
WFTVfFZ
likSQG3
zJQBmDr
TAJBgNV
XJ0aWZp
BgwFoAU
FnOdYJt
fTiDz1R
RUeDovl
      +GFIBZ+GNF/cAYKcMtGcrs2i97ZkJMo="
    }
  ],
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\\\\"a330bc54f0671c9\\",
    "location": "https://example.com/v1/Users/2819c223-7f76-453a-919d-413
861904646"
  }
}

```

12.3. Enterprise User Extension Representation

The following is a non-normative example of the fully populated User using the enterprise User extension in JSON format.

```
{
  "schemas": ["urn:scim:schemas:core:2.0:User", "urn:scim:schemas:extension:enterprise:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "externalId": "701984",
  "userName": "bjensen@example.com",
  "name": {
    "formatted": "Ms. Barbara J Jensen III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
  }
}
```

```
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "nickName": "Babs",
  "profileUrl": "https://login.example.com/bjensen",
  "emails": [
    {
      "value": "bjensen@example.com",
      "type": "work",
      "primary": true
    },
    {
      "value": "babs@jensen.org",
      "type": "home"
    }
  ],
  "addresses": [
    {
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza\nHollywood, CA 91608 USA",
      "type": "work",
      "primary": true
    },
    {
      "streetAddress": "456 Hollywood Blvd",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "456 Hollywood Blvd\nHollywood, CA 91608 USA",
      "type": "home"
    }
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    },
    {
      "value": "555-555-4444",
      "type": "mobile"
    }
  ],
  "ims": [
```


kiG9w0B

AQEFAAOCAQ8AMIIBCgKCAQEA7Kr+DcDs/JQ5GwejJFcBIP682X3xpjis5

6AK02bc

Mortimore, et al.

Expires March 03, 2014

[Page 25]

```

1FLgzdLI8auoR+cC9/Vrh5t66HkQIOdA4unHh0AaZ4xL5PhVbXIPMB5vA
PKpzz5i
PSi8x08SL7I7SDhcBVJhqVqr3Hg1lEG6UC1DdHO7nkLuwXq8HcISKkbT5
WFTVfFZ
zidPl8HZ7DhXkZIRtJwBweq4bvm3hM1Os7UQH05ZS6cVDgweKNwdLLrT5
1iksSQG3
DYrl+ft781UQRIqxgwgqCfXEuDiinPh0kkvIi5jivVu1Z9QiwlyEdRbLJ4
zJQBmDr
SGTMYn4lRc2HgHO4DqB/bnMVorHB0CC6AV1QoFK4GPe1LwIDAQABo3swe
TAJBgNV
HRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZ
XJ0aWZp
Y2F0ZTAdBgNVHQ4EFgQU8pD0U0vsZIsaA16lL8En8bx0F/gwHwYDVR0jB
BgwFoAU
dGeKitcaF7gnzsNwDx708kqaVt0wDQYJKoZIhvcNAQEFBQADgYEAA81Ss
FnOdYJt
Ng5Tcq+/ByEDrBgnusx0jloUhByPMEVkoMZ3J7j1ZgI8rAbOkNngX8+pK
fTiDz1R
C4+dx8oU6Za+4NJXUjllL5CvV6BEYb1+QAEJwitTVvxB/A67g42/vzgAto
RUEdov1
+GFibZ+GNF/cAYKcMtGcrs2i97ZkJMo="
}
],
"urn:scim:schemas:extension:enterprise:2.0:User": {
  "employeeNumber": "701984",
  "costCenter": "4130",
  "organization": "Universal Studios",
  "division": "Theme Park",
  "department": "Tour Operations",
  "manager": {
    "managerId": "26118915-6090-4610-87e4-49d8ca9f808d",
    "$ref": "/Users/26118915-6090-4610-87e4-49d8ca9f808d",
    "displayName": "John Smith"
  }
},
"meta": {
  "resourceType": "User",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\\\\"3694e05e9dff591\\",
  "location": "https://example.com/v1/Users/2819c223-7f76-453a-919d-413
861904646"
}
}

```

12.4. Group Representation

The following is a non-normative example of SCIM Group representation in JSON format.

```
{
```

```
"schemas": ["urn:scim:schemas:core:2.0:Group"],  
"id": "e9e30dba-f08f-4109-8486-d5c6a331660a",  
"displayName": "Tour Guides",  
"members": [  
  {
```

```

    "value": "2819c223-7f76-453a-919d-413861904646",
    "$ref": "https://example.com/v1/Users/2819c223-7f76-453a-919d-41386
1904646",
    "display": "Babs Jensen"
  },
  {
    "value": "902c246b-6245-4190-8e05-00816be7344a",
    "$ref": "https://example.com/v1/Users/902c246b-6245-4190-8e05-00816
be7344a",
    "display": "Mandy Pepperidge"
  }
]
"meta": {
  "resourceType": "Group",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\/"3694e05e9dff592\"",
  "location": "https://example.com/v1/Groups/e9e30dba-f08f-4109-8486-d5
c6a331660a"
}
}

```

12.5. Service Provider Configuration Representation

The following is a non-normative example of the SCIM Service Provider Configuration representation in JSON format.

```

{
  "schemas": ["urn:scim:schemas:core:2.0:ServiceProviderConfig"],
  "documentationUrl": "http://example.com/help/scim.html",
  "patch": {
    "supported": true
  },
  "bulk": {
    "supported": true,
    "maxOperations": 1000,
    "maxPayloadSize": 1048576
  },
  "filter": {
    "supported": true,
    "maxResults": 200
  },
  "changePassword": {
    "supported": true
  },
  "sort": {
    "supported": true
  },
  "etag": {

```



```

    "supported":true
  },
  "authenticationSchemes": [
    {
      "name": "OAuth Bearer Token",
      "description": "Authentication Scheme using the OAuth Bearer Token
Standard",
      "specUrl": "http://tools.ietf.org/html/draft-ietf-oauth-v2-bearer-01",
      "documentationUrl": "http://example.com/help/oauth.html",
      "type": "oauthbearertoken",
      "primary": true
    },
    {
      "name": "HTTP Basic",
      "description": "Authentication Scheme using the Http Basic Standard",
      "specUrl": "http://www.ietf.org/rfc/rfc2617.txt",
      "documentationUrl": "http://example.com/help/httpBasic.html",
      "type": "httpbasic"
    }
  ],
  "meta": {
    "resourceType": "ServiceProviderConfig",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/\\"3694e05e9dff594\" "
  }
}

```

12.6. Resource Type Representation

The following is a normative example of the SCIM Resource Type representation in JSON format.

```

{
  "schemas": ["urn:scim:schemas:core:2.0:ResourceType"],
  "name": "User",
  "endpoint": "/Users",
  "description": "Core User",
  "schema": "urn:scim:schemas:core:2.0:User",
  "schemaExtensions": [
    {
      "schema": "urn:scim:schemas:extension:enterprise:2.0:EnterpriseUser",
      "required": true
    }
  ],
  "meta": {
    "resourceType": "ResourceType",

```



```

    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/\\"3694e05e9dff595\"
  }
}

```

12.7. Schema Representation

The following is a normative example of the SCIM Schema representation in JSON format.

```

{
  "id": "urn:scim:schemas:core:2.0:User",
  "name": "User",
  "description": "Core User",
  "attributes":[
    {
      "name":"id",
      "type":"string",
      "multiValued":false,
      "description":"Unique identifier for the SCIM resource as defined by the Service Provider. Each representation of the resource MUST include a non-empty id value. This identifier MUST be unique across the Service Provider's entire set of resources. It MUST be a stable, non-reassignable identifier that does not change when the same resource is returned in subsequent requests. The value of the id attribute is always issued by the Service Provider and MUST never be specified by the Service Consumer. REQUIRED.",
      "readOnly":true,
      "required":true,
      "caseExact":false
    },
    {
      "name":"name",
      "type":"complex",
      "multiValued":false,
      "description":"The components of the user's real name. Providers MAY return just the full name as a single string in the formatted sub-attribute, or they MAY return just the individual component attributes using the other sub-attributes, or they MAY return both. If both variants are returned, they SHOULD be describing the same name, with the formatted name indicating how the component attributes should be combined.",
      "readOnly":false,
      "required":false,
      "caseExact":false,
      "subAttributes":[
        {
          "name":"formatted",
          "type":"string",
          "multiValued":false,
          "description":"The full name, including all middle names, titles, and suffixes as appropriate, formatted for display (e.g. Ms. Barbara J Jensen, III.)." ,

```

```
"readOnly":false,  
"required":false,  
"caseExact":false  
},  
{  
  "name":"familyName",  
  "type":"string",
```

```
        "multiValued":false,
        "description":"The family name of the User, or Last Name in most
Western languages (e.g. Jensen given the full name Ms. Barbara J Jensen, I
II.).",
        "readOnly":false,
        "required":false,
        "caseExact":false
    },
    {
        "name":"givenName",
        "type":"string",
        "multiValued":false,
        "description":"The given name of the User, or First Name in most
Western languages (e.g. Barbara given the full name Ms. Barbara J Jensen,
III.).",
        "readOnly":false,
        "required":false,
        "caseExact":false
    },
    {
        "name":"middleName",
        "type":"string",
        "multiValued":false,
        "description":"The middle name(s) of the User (e.g. Robert given
the full name Ms. Barbara J Jensen, III.).",
        "readOnly":false,
        "required":false,
        "caseExact":false
    },
    {
        "name":"honorificPrefix",
        "type":"string",
        "multiValued":false,
        "description":"The honorific prefix(es) of the User, or Title i
n most Western languages (e.g. Ms. given the full name Ms. Barbara J Jensen,
III.).",
        "readOnly":false,
        "required":false,
        "caseExact":false
    },
    {
        "name":"honorificSuffix",
        "type":"string",
        "multiValued":false,
        "description":"The honorific suffix(es) of the User, or Suffix
in most Western languages (e.g. III. given the full name Ms. Barbara J Jense
n, III.).",
        "readOnly":false,
        "required":false,
        "caseExact":false
    }
]
},
{
```

```
"name": "emails",  
"type": "complex",  
"multiValued": true,
```

```

    "description": "E-mail addresses for the user. The value SHOULD be
canonicalized by the Service Provider, e.g. bjensen@example.com instead of b
jensen@EXAMPLE.COM. Canonical Type values of work, home, and other.",
    "readOnly": false,
    "required": false,
    "caseExact": false,
    "subAttributes": [
      {
        "name": "value",
        "type": "string",
        "multiValued": false,
        "description": "E-mail addresses for the user. The value SHOULD
be canonicalized by the Service Provider, e.g. bjensen@example.com instead
of bjensen@EXAMPLE.COM. Canonical Type values of work, home, and other.",
        "readOnly": false,
        "required": false,
        "caseExact": false
      },
      {
        "name": "display",
        "type": "string",
        "multiValued": false,
        "description": "A human readable name, primarily used for displ
ay purposes. READ-ONLY.",
        "readOnly": true,
        "required": false,
        "caseExact": false
      },
      {
        "name": "type",
        "type": "string",
        "multiValued": false,
        "description": "A label indicating the attribute's function; e.
g., 'work' or 'home'.",
        "readOnly": false,
        "required": false,
        "caseExact": false,
        "canonicalValues": ["work", "home", "other"]
      },
      {
        "name": "primary",
        "type": "boolean",
        "multiValued": false,
        "description": "A Boolean value indicating the 'primary' or pre
ferred attribute value for this attribute, e.g. the preferred mailing addres
s or primary e-mail address. The primary attribute value 'true' MUST appear
no more than once.",
        "readOnly": false,
        "required": false,
        "caseExact": false
      }
    ],
    "name": "addresses",

```

```
"type": "complex",  
  "multiValued": true,  
  "description": "A physical mailing address for this User, as described in (address Element). Canonical Type Values of work, home, and other. The value attribute is a complex type with the following sub-attributes.",
```

```
"readOnly":false,
"required":false,
"caseExact":false,
"subAttributes":[
  {
    "name":"formatted",
    "type":"string",
    "multiValued":false,
    "description":"The full mailing address, formatted for display
or use with a mailing label. This attribute MAY contain newlines.",
    "readOnly":false,
    "required":false,
    "caseExact":false
  },
  {
    "name":"streetAddress",
    "type":"string",
    "multiValued":false,
    "description":"The full street address component, which may in
clude house number, street name, PO BOX, and multi-line extended street addr
ess information. This attribute MAY contain newlines.",
    "readOnly":false,
    "required":false,
    "caseExact":false
  },
  {
    "name":"locality",
    "type":"string",
    "multiValued":false,
    "description":"The city or locality component.",
    "readOnly":false,
    "required":false,
    "caseExact":false
  },
  {
    "name":"region",
    "type":"string",
    "multiValued":false,
    "description":"The state or region component.",
    "readOnly":false,
    "required":false,
    "caseExact":false
  },
  {
    "name":"postalCode",
    "type":"string",
    "multiValued":false,
    "description":"The zipcode or postal code component.",
    "readOnly":false,
    "required":false,
    "caseExact":false
  }
]
```



```

    },
    {
      "name": "country",
      "type": "string",
      "multiValued": false,
      "description": "The country name component.",
      "readOnly": false,
      "required": false,
      "caseExact": false
    },
    {
      "name": "type",
      "type": "string",
      "multiValued": false,
      "description": "A label indicating the attribute's function; e.
g., 'work' or 'home'.",
      "readOnly": false,
      "required": false,
      "caseExact": false,
      "canonicalValues": ["work", "home", "other"]
    }
  ]
}
],
"meta": {
  "resourceType": "Schema",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\\"3694e05e9dff596\"",
  "location": "https://example.com/v1/Schemas/urn:scim:schemas:core:2.0
:User"
}
}

```

13. Security Considerations

The SCIM Core schema contains personally identifiable information as well as other sensitive data. Aside from prohibiting password values in a SCIM response this specification does not provide any means or guarantee of confidentiality.

14. Normative References

[PortableContacts]

Smarr, J., "Portable Contacts 1.0 Draft C - Schema Only", August 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Appendix A. Contributors

The SCIM Community would like to thank the following people for the work they've done in the research, formulation, drafting, editing, and support of this specification.

Morteza Ansari (morteza.ansari@cisco.com)

Sidharth Choudhury (schoudhury@salesforce.com)

Samuel Erdtman (samuel@erdman.se)

Kelly Grizzle (kelly.grizzle@sailpoint.com)

Chris Phillips (cjphillips@gmail.com)

Erik Wahlstroem (erik.wahlstrom@nexussafe.com)

Special thanks to Joeseeph Smarr, who's excellent work on the Portable Contacts Specification [PortableContacts] provided a basis for the SCIM schema structure and text.

Authors' Addresses

Chuck Mortimore (editor)
Salesforce.com

Email: cmortimore@salesforce.com

Patrick Harding
Ping Identity

Email: pharding@pingidentity.com

Paul Madsen
Ping Identity

Email: pmadsen@pingidentity.com

Trey Drake
UnboundID

Email: trey.drake@unboundid.com