SCIM Issue #19
Adopt the vCard Schema?

Findings, Pro and Con
IETF-85 Atlanta
Background

• Adoption of vCard as the SCIM Schema has been a long-standing suggestion
• Making such a change is a major decision for the SCIM spec, as it represents a significant amount of work and affects other schema changes also under consideration
• Therefore, making a decision sooner is better.
Key Sources

• SCIM and vCard Mapping

• vCard RFC6350
  • S. Perreault
  • http://tools.ietf.org/html/rfc6350

• vCard KIND spec for applications
  – Peter Saint-Andre, et al.

• SCIM Use Cases
  – P. Hunt, et al.
  – https://datatracker.ietf.org/doc/draft-zeltsan-scim-use-cases/

• vCard JSON format
  – R. Bhat & P. Saint-Andre
Goals for a SCIM Schema

It is granted that the standard needs to provide a schema to encourage adoption...

• Don’t try to incorporate every possible attribute or support every use case
• Instead, support the essential core attributes and provide for extensibility for the rest
• Define the interchange format, NOT the storage format
• Don’t reinvent wheels that already exist
A nod to other alternatives

- inetOrgPerson
- POCO (was input to vCard 4 and to SCIM)
- XFN (considered as input to vCard 4)
- OMA, W3C (fragmentation)
- Other social networking standards
The Questions

Is a JSON representation of the attributes and semantics defined by vCard a viable schema format for SCIM?

Are there enough advantages inherent in vCard to justify a change for SCIM?
The Findings

Is a JSON representation of the attributes and semantics defined by vCard a viable schema format for SCIM?

YES.

Are there enough advantages inherent in vCard to justify a change for SCIM?

NO.
Considerations in evaluating use of vCard for SCIM

- Extensibility (forms & formality)
- Support for meta-data, on attributes and values
- Backward compatibility & the adoption of the current SCIM schema
- Alignment with SCIM goals (a philosophical question)
- Ease of Integration
- Existence of a JSON format spec
- Support for complex objects & relationships
  - application types
  - devices

- Notation style was NOT considered.
<table>
<thead>
<tr>
<th>Aspect/Consideration</th>
<th>Importance</th>
<th>Comment / Finding</th>
<th>Pro or Con?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema match?</td>
<td>High</td>
<td>There is a close overlap in attributes. In addition, vCard already defines additional resource types, but these can be incorporated into SCIM if desired</td>
<td>Neutral</td>
</tr>
<tr>
<td>Extensibility</td>
<td>High</td>
<td>The vCard use of prefixes for extensions requires more parsing &amp; assumptions by the client than the current SCIM approach which use URN namespaces.</td>
<td>Con</td>
</tr>
<tr>
<td>Complex Objects</td>
<td>Med</td>
<td>Yes, vCard supports nested objects. Inheritance does not exist in either</td>
<td>Neutral</td>
</tr>
<tr>
<td>Meta-Data</td>
<td>High</td>
<td>vCard expressions of meta-data are equivalent to current SCIM. vCard would not add expressions that do not exist yet in SCIM, like meta-data annotations of multi-values.</td>
<td>Neutral</td>
</tr>
<tr>
<td>JSON Representation</td>
<td>High</td>
<td>There is a draft defining a natural mapping to of vCard to JSON</td>
<td>Neutral</td>
</tr>
<tr>
<td>Backward Compatibility/Adoption</td>
<td>High</td>
<td>vCard is not yet adopted by any/many identity providers; SCIM is getting traction. A switch means losing that traction.</td>
<td>Con</td>
</tr>
<tr>
<td>Lowering ‘SCIM barrier’ for existing vCard services</td>
<td>Low</td>
<td>It is possible that services which already support vCard would have an easier time using SCIM if vCard were adopted. This is reduced if the JSON representation is required.</td>
<td>Pro</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Med</td>
<td>Many x-values are in common usage, but are not formalized</td>
<td>Con</td>
</tr>
</tbody>
</table>
Recommendation

• Our conclusion is the vCard should not be adopted into the SCIM specification

• draft-greevenbosch-scim-vcard-mapping is worth pursuing

• However, if vCard is adopted
  – Recommend supporting ONLY the JSON representation, not the text vCard form
  – Recommend defining IANA names, not relying on common usage of “x-” names
Backup material follows
Breadth of Adoption

• Many libraries exist for vCard, but NOT for the JSON representation
• Many X-terms are in common usage, and are not formalized in the spec or IANA
• vCard is not yet adopted by any/many Identity Providers
SCIM attributes not in vCard

• externalId
• userName, displayName,
• meta/* ( except meta/lastModified )
• locale
• active
• password
• costCenter, division, department, manager/
  managerId, manager/displayName
• entitlements
Required vCard attributes

• FN (formatted name)
  – Equivalent to SCIM name.formatted for Users
  – Equivalent to SCIM displayname for Groups

• VERSION (of the vCard Spec)
  – In SCIM this is implied in the URN of the schemas element
Schema Findings

• A very good match from SCIM into vCard, except:
  – Need conversion from SCIM “id” to vCard “UID” (Greevenbosch)
  – Addresses will need to be split into components

“...mapping between attributes and properties with similar, but not equal, semantics.”

-Bert Greevenbosch
Extensibility?

• “x-” prefixed names are used for KIND, attributes, and parameters
  – Caution: “[...] reserved for experimental use, not intended for released products, or for use in bilateral agreements.” RFC6350

• “vnd-{PEN}” prefixes support vendor namespaces

• New names can be IANA registered.
  – New KINDs specify attributes.

• No need for a vCard spec revision
Complex Objects?

• No hard & fast rules, but can do nested structures
• vCards can contain pointers to other vCards
• The “related” parameter supports relationship types, currently:
  
  – related-type-value = "contact" / "acquaintance" / "friend" / "met" / "co-worker" / "colleague" / "co-resident" / "neighbor" / "child" / "parent" / "sibling" / "spouse" / "kin" / "muse" / "crush" / "date" / "sweetheart" / "me" / "agent" / "emergency"
vCard meta-data

• Some properties already have some meta-data
  – Email (work, home)
  – Home (work, home, fax)

– Example: Expiration Date
  • “This parameter can be applied to these properties”
  • Will need to work with the Vcard experts to get advice
Comparison of Data Types

• Parameters
  "tel": { "type": [ "voice", "video" ],
  "uri": "tel:+1-555-555-555" }

• Multi-Values
  – FIXME
Findings re: JSON format

• Is in the first stages of development (IETF draft)
• Maintains the extensibility of vCard
• Defines a name conversion to lower-case
• Parameters are contained objects
• Multi-values are JSON arrays
• Structured properties are nested object trees
• Others are name-value pairs
• Format conversions are TBD
• Schema is TBD (FIXME ask Peter Saint-Andre about this)
About vCard

• FIXME more here, quick summary of:
  – History
  – Current Version
  – Scope
    • Business Cards (Individuals), Groups, Locations, Orgs
    • Extensions for Applications, Systems, “Things”, Calendar Events
  – RFC Status
  – Adoption
    • Which existing systems that might use SCIM already support vCard?
      – Active Directory, Jabber
      – Are there extant libraries?

• Terms:
  – KIND = SCIM resource
  – Property = SCIM attribute
  – Parameter = e.g. {work phone, home address} = SCIM multi-valued ‘type’

• Subjective impressions:
  – Mostly about human people, not so much for accounts
  – Best for semi-permanent data, not dynamic data