

Alternative Proposals on the Relationship Between TIER and SCIM APIs and Schema

~~Proposal 1) TIER APIs and Schema have some overlap with SCIM APIs and Schema. The development of TIER defined API and Schema is not bound by SCIM specifications~~

~~Pros:-~~

- ~~1. Maximum freedom in defining TIER specific APIs and schema~~
- ~~2. SCIM compliance for simple API calls can be achieved by proper configuration of the parties.~~
 - ~~a. Comment: Not in a SCIM compliant way~~
- ~~3. Minimal development effort is required to provide simple SCIM responses. Simple SCIM responses will be enough to meet the majority of real-world SCIM use cases.~~
- ~~4. a. Considering all that goes into a service 'Minimal effort' for the SCIM response seems less necessary.~~

~~Gens:~~

- ~~1. Less SCIM compliance overall than other proposals on the table~~
- ~~2. Ongoing challenge to explain to developers and others where and how TIER diverges from SCIM~~
- ~~3. Resources are littered with 'tier' prefixes.~~

Proposal 2)

SCIM APIs are a strict subset of TIER APIs. Rather than relying solely on SCIM User and Group Resource Types, TIER defines and uses new Resource Types: TierGroup and TierUser using SCIM-defined procedures.

Pros:

1. Full SCIM compliance: Can be accomplished in conformance with SCIM RFCs using SCIM-defined procedures for defining new Resource Types and their core schema.
2. Freedom to define our own resource types and schema without having to prefix each attribute name with 'tier'
3. Avoids use of the choppy SCIM schema extension mechanism for Users and Groups (SCIM mandates all core attributes together followed by sections of SCIM extensions and TIER extensions).
4. Existing SCIM clients can access TIER resources and attributes without modification—in theory anyway.
5. SCIM SDKs and Libraries would work (by and large)

Cons and further discussion:

1. TIER User and Group Resource Types would have "Tier" in their names while other resource type names would not (or would they?)
 - a. Comment: For clarity and to avoid conflict, probably they would. But the important bit is the schema registration.
2. To support SCIM User and Group objects, TIER would need to provide bi-directional, automated mapping between the TierUser and TierGroup schema and the corresponding SCIM schema.
 - a. Q: Does that mean we still do prefixing for these objects?
 - b. A: No, I'd say when converting objects to SCIM, we'd ditch all the TIER ResourceType attributes that don't have a direct counterpart in SCIM. That way SCIM endpoints would see pure SCIM.
 - c. Q: Can we fix issues with SCIM e.g. that we want to offer a better way to page through large resultsets?
 - d. A: Since we declare TIER a superset of SCIM, we could define points on which TIER APIs diverge from SCIM APIs, but I think we should set a pretty high bar for deciding it's necessary to do that. If there's something wrong with SCIM, we should first really try to get it fixed in the next release of SCIM. In general, better lines of communication with the SCIM community is a good thing.
 - e. Q: Does this mean we can't add things to the Meta object? We are back to having a Meta and a TierMeta for Tier resource types?
 - f. A: If we just make sure all the SCIM meta attributes have a direct counterpart in the meta element we define for each TIER Resource Type, then we just need one meta object. Mapping from a TIER Resource Type to a SCIM Resource Type is then a straight mapping of corresponding attributes, plus filtering out all the TIER-specific meta elements.
 - g. Comment: We don't *need* to do the mapping. We can if we think there is value in doing so.
3. This solves the problem of TIER wanted to pepper the object model with new attributes, but what about when institutions want to add new attributes, they will have to either define institution specific resource types which is a lot of work or follow the scim extension model which is clunky.
 - a. Response: Worth discussing. I'm inclined to say that for TIER-defined Resource Types we should allow local implementations to supplement the TIER schema at will with locally-defined attributes but insist that they prefix those attributes in a way that minimizes the chance of clashing definitions. That can be accomplished by requiring the 2nd-level component of the local domain name (the part just to the left of .edu) as a prefix. This isn't guaranteed to work for non-edu sites, so there are corner cases to keep an eye on. Then mapping from a local resource representation down to 'Pure TIER' or 'Pure SCIM' can be accomplished with a well-defined and simple filtering rule.
 - b. Local implementations can do what they want, but adding attributes would break SCIM compliance. The new attributes would not be in the resource schema specification. One of the advantages of adopting a standard, in addition to having little documentation to do ourselves, is that we can take advantage of already written applications and libraries that work with the standard. A quick search found several libraries for SCIM and Python. One of which defines a "EnterpriseUser" resource. So we are not the first to propose this. Whether or not undocumented attributes would break anything is unknown.
 - c. Since it is we who control the schema, allowing institutions to propose formal additions should not be out of the question.
 - d. Comment: What exactly is "clunky" about the extension mechanism in this use case? For TIER API as a whole it is problematic because we need to "redefine" a bunch of attributes, but for campus extensions this shouldn't happen. (Or else the campus should do the same

thing and create a CampusPerson or whatever.) So is it just that the attribute label is lengthy? Perhaps we need a convention for extending complex attributes?

4. Some call this a con, some say pro, but you would have /TierUser and /User, /TierGroup, and /Group
 - a. There is duplication here, might be more to implement, might have drift between the object models
 - b. My contention is that we do not have /TierUser and /User, /TierGroup, and /Group. In fact, in our API we have only EduPeople and EduGroups, and EduMembers (to use the new names). Our API does not have Users or Groups. It may be a service would add support for those SCIM core resources, but it wouldn't have to. The two are separate APIs. We might try to keep our resources as supersets of the scim core ones, to help sites that want to support both, but we wouldn't have to.
 - i. Response: +1, The 'Edu' prefix has the advantage of giving TIER complete control over its resource schema. The 'superset of SCIM' idea minimizes the effort needed to map Tier resource representations to SCIM and vice versa.