APIs and Schema--The Relationship Between TIER and SCIM

For purposes of this document, the term SCIM is a shorthand for RFC's 7642, 7643 and 7644

Principles

1. TIER API specifications follow SCIM conventions on syntax and semantics unless specific and documented alternatives are called out
2. Some TIER APIs may be defined as profiles of SCIM-defined APIs
3. TIER Resource Type specifications follow SCIM conventions on schema definition and extension processes, syntax and semantics
4. Some TIER resource types may be defined as profiles of SCIM resource types
5. In the case of messages that carry a representation of a resource, they should share the data structure and schema of its API counterpart.

Benefits of following these principles:

1. Freedom to define new resource types and schema and/or use existing SCIM resource types and schema
2. TIER APIs and schema can be generated in conformance with SCIM RFCs using SCIM-defined procedures for defining new Resource Types and their core schema.
3. Existing SCIM clients can access SCIM-profiled TIER resources and attributes without modification
4. SCIM SDKs and libraries may be partially reusable for implementing TIER APIs

Further guidance

1. For each TIER-defined resource type, ensure that all required SCIM meta attributes have a direct counterpart in the TIER meta element
2. To add Institution-level schema elements to TIER-defined resource types, follow the SCIM-defined schema extension methods
3. Define Internet2 Trust and Identity processes for self-registration of resource types and their schema: :everage Open API 2.0 and some API Manager, Developer Portal