Unified Properties for ALTO

draft-ietf-alto-unified-props-new-07

Wendy Roome
Sabine Randriamasy
Y. Richard Yang
J. Jensen Zhang
Unified Properties Draft Updates - summary

Goal of revisions -06 and -07

- Clarify the basic concepts and definitions with motivations and examples
- Clarify the issue on resource dependency of property map
- Clarify the approach to specify resource dependencies for property map

Design decisions and discussions

- Two design options to specify resource dependencies for property map:
  - Option 1: domain-specific property design; a property is bound to an entity domain (current revision v07)
  - Many drawbacks with option 1:
    - Option 2: resource-specific design, inspired by relational database; dependencies bind to each individual entity and property
    - Will be specified in a next revision
Overview of Updates in V07

• Major changes in Section 2
• Revisited definitions of the unified properties
  • Property Type = unique domain-specific property identifier
  • Property Name: the name used in the Property Type
  • Dependency Type: media type of the dependent information resource
• Changed some terminology
  • Entity Address -> Entity Identifier
• Proposed an Option 1 to indicate resource dependencies
• Option 1 will be revised
Resource Dependencies in a property map

• Some properties are only applicable for particular entity domains
  • **Example:** the "pid" property is not applicable for entities in the "pid" domain.

• The interpretation of the value of a property may depend on the entity domain
  • **Example**, if some "geo-location" property is defined as the "latitude, longitude" coordinates of a point,
  • when applied to an entity identified by an address in the ipv4 or ipv6 domain, e.g., a host, the property describes the host location and does not depend on any resource
  • when applied to an entity with an identifier in the "pid" domain, e.g., "myPID10"
    • Property may indicate the location of the center of all hosts in entity "myPID10" ➔ different meaning
    • Property depends on the Network Map defining this "pid" entity ➔ specific dependent resource type
Resource dependency ambiguity in a property map: example

• If a Server wants to expose
  • Properties: "cdni-fci-capabilities" and "pid"
  • for entities in the domains: "ipv4", "ipv6", "countrycode" and "asn"

• The IRD entry below does not allow to relate properties and “used” resources because it introduces ambiguities

  • "filtered-cdnifci-property-map" : {⋯
    "uses" : [ "my-default-network-map", "my-default-cdnifci-map" ]
    "capabilities" : {
      "entity-domain-types" : [ "ipv4", "ipv6", "countrycode", "asn" ],
      "prop-types" : [ "cdni-fci-capabilities", "pid" ]
    }
  }
Resource dependency ambiguity in a property map: example

- because "pid" cannot be directly mapped to entities in the "countrycode" and "asn" domains, Server may have to define 2 property maps and compose its IRD information resources as follows:
  - "filtered-cdnifci-property-map" : {
      "uses" : [ "my-default-cdnifci-map" ]
      "capabilities" : {
        "entity-domain-types" : [ "ipv4", "ipv6", "countrycode", "asn" ],
        "prop-types" : [ "cdni-fci-capabilities" ]
      }
    }
  - "filtered-cdnifci-pid-property-map" : {
      "uses" : [ "my-default-network-map", "my-default-cdnifci-map" ]
      "capabilities" : {
        "entity-domain-types" : [ "ipv4", "ipv6" ],
        "prop-types" : [ "cdni-fci-capabilities", "pid" ]
      }
    }

Design Proposal in current version 07: Option 1

• Introduction of Property Type
  • Associates property name with applicable entity domain
  • Each property type has a unique identifier encoded with the following format:
    • PropertyType ::= DomainName : PropertyName
  • DomainName: indicates which entity domain the property type applies to.
  • PropertyName: a JSON string

• Example: property types "ipv4:pid" and "ipv6:pid"
  • have the same Property Name "pid" associated with both "ipv4" and "ipv6" domains.

• Different property types can have the same property name applied to different entity domains, if they have similar semantics.
Option 1: Rules to Specify Dependencies

- Each property type (entity domain : property) MUST indicate a sequence of
dependency types (can be empty), e.g.,
  - "ipv4:geo-location" has no dependency
  - "pid:geo-location" depends on resource type "alto-networkmap"
  - "pid:cdni-fci-capabilities" depends on resource types "alto-networkmap" and "alto-cdnifci"
- Property types with different types of dependent resources MUST NOT be put
  into the same property map, e.g.,
  - Property map for {"entity-domains": ["ipv4", "pid"], "properties": ["geo-location"]} is illegal.
- All entities and property values in a same property map MUST have the same
  dependent resources, e.g.,
  - Property ipv4:pid using networkmap1 cannot be in the same property map as property
    ipv4:pid using networkmap2
Issue with option 1

- Design option 1 binds the resource dependencies to the whole property map.
  - **Benefit:** each property map is very simple and easy to handle.
  - **Drawback:**
    - the ALTO server may need to export too many property maps.
    - Design is too complex
Option 2 Motivation: Revisit Endpoint Property Map in RFC7285: A Database View

```
"netmap-1": {
  "PID1": {
    "ipv4": ["1.1.1.0/24", "1.1.2.0/24"]
  },
  "PID2": {
    "ipv4": ["1.1.3.0/24"]
  }
},
"netmap-2": {
  "PID2-1": {
    "ipv4": ["1.1.1.0/24"]
  },
  "PID2": {
    "ipv4": ["1.1.2.0/24", "1.1.3.0/24"]
  }
}
```

```
"endpoint-property-map": {
  "ipv4:1.1.1.0/24": {
    "netmap-1.pid": "PID1",
    "netmap-2.pid": "PID1"
  },
  "ipv4:1.1.2.0/24": {
    "netmap-1.pid": "PID1",
    "netmap-2.pid": "PID2"
  },
  "ipv4:1.1.3.0/24": {
    "netmap-1.pid": "PID2",
    "netmap-2.pid": "PID2"
  }
}
```

```
<table>
<thead>
<tr>
<th>ipv4</th>
<th>netmap-1.pid</th>
<th>netmap-2.pid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>PID1</td>
<td>PID1</td>
</tr>
<tr>
<td>1.1.2.0/24</td>
<td>PID1</td>
<td>PID2</td>
</tr>
<tr>
<td>1.1.3.0/24</td>
<td>PID2</td>
<td>PID2</td>
</tr>
</tbody>
</table>
```
Option 2 Motivation: Revisit Endpoint Property Map in RFC7285: A Database View

- Each resource defines a table in a database
- A column of a table defines an attribute
- The projection of a column defines its domain

- Each table defines one or multiple mappings
  - Table1 (netmap-1)
    - ipv4 -> pid
    - [pid -> ipv4]
  - Table2 (netmap-2)
    - ipv4 -> pid
    - [pid -> ipv4]
- A property map defines a view
  - attr -> property
- UP as a SQL query (for each input)
  - select ipv4, netmap-1.pid, netmap-2.pid from netmap-1, netmap-2 where netmap-1.ipv4=ipv4.input netmap-2.ipv4=ipv4.input

<table>
<thead>
<tr>
<th>Table1: netmap-1</th>
<th>Table2: netmap-2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pid</strong></td>
<td><strong>ipv4</strong></td>
</tr>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
</tr>
<tr>
<td>PID1</td>
<td>1.1.2.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
</tr>
</tbody>
</table>
Design Space for Option 2: Entity

• Entity Domain Type Name Space: ipv4, ipv6, pid, asn, ane, ...
  • Analogy to DB: a type and name of a table column

• Entity Domain:
  • A set of entities from the attribute of a specific resource
    • Analogy to DB: a projection of a column/attribute
  • Identification
    <domain_id> ::= <resource-id>.<domain_type>

• Entity:
  • An element in an entity domain, identified by
    <entity_id> ::= <domain_id>::<domain_specific_entity_id>

Table 1: netmap-1

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2/24</td>
</tr>
<tr>
<td>PID3</td>
<td>1.1.3/24</td>
</tr>
</tbody>
</table>

Table 2: netmap-2

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2/24</td>
</tr>
<tr>
<td>PID3</td>
<td>1.1.3/24</td>
</tr>
</tbody>
</table>
Design Space for Option 2: Property

• Property Type Name Space: pid, ⋯, geo-location

• Goal: Identify a property for an entity in an entity domain

• Identified by
  • `<resource-id>..<prop>` (same as Sec 10.8 of RFC7285)

<table>
<thead>
<tr>
<th>Table1: netmap-1</th>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
<td></td>
</tr>
<tr>
<td>PID1</td>
<td>1.1.2.0/24</td>
<td></td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table2: netmap-2</th>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
<td></td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2.0/24</td>
<td></td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
<td></td>
</tr>
</tbody>
</table>
Design Space for Option 2: Resource Type -> Property Map

• Each resource type (network map, endpoint property map) defines
  • Entities: a list of
    • entity domain type in Entity Domain Name Space
    • properties in Property Name Space of entities in the entity domain type

• Example:
  • For the network map (alto-netmap) resource type
    • Entity domain type: ipv4
      • Property of this domain type -> pid property
    • Entity domain type: ipv6
      • Property of this domain type -> pid property

Table1: netmap-1

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
</tr>
<tr>
<td>PID1</td>
<td>1.1.2.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
</tr>
</tbody>
</table>

Table2: netmap-2

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
</tr>
</tbody>
</table>
Design Space for Option 2: Property Map Resource

• A unified property map resource itself includes
  • An entity domain type
  • Properties provided for the entities (refer to as this.prop)
  • Properties imported from other resources (refer to as resource-id.prop)

• Note: this can be a recursive process

Table1: netmap-1

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
</tr>
</tbody>
</table>

Table2: netmap-2

<table>
<thead>
<tr>
<th>pid</th>
<th>ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID1</td>
<td>1.1.1.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.2.0/24</td>
</tr>
<tr>
<td>PID2</td>
<td>1.1.3.0/24</td>
</tr>
</tbody>
</table>
How to Proceed with IANA Registry

• Registry for each entity domain type
• Registry for each property type (just call property)
• Registry for the mapping provided by each resource type
  • Entity domain types
  • For each entity domain type
  • Properties
Next Steps

• Process
  • Discuss option 1 and option 2, will
    • post on mailing list
    • arrange at least two meetings in the weekly alto design meeting (all are welcome to join)
  • Finish the new revision of the draft for WGLC (in 2 weeks)
    • Draft of option 2 exists, but have not checked in
      • The main text will not change, mostly focus on only two subsections and example notitions

• Authors goal:
  • Submit draft with the final design to WG by end of April