Quantum Service API extensions

This document describes a proposal of OpenStack Quantum Service core API extensions. Based on earlier Quantum Service Core API document[1], we proposed extensible structures "OpenStack Port Profile" and "OpenStack Stats" to describe a port's configuration and stats. The usage of these structures are illustrated in the port details examples.

Resources

The following graph shows what resources are included in the Quantum service and how they connect with each other. As the graph shows, there are mainly 3 resources: network, port and attachment.

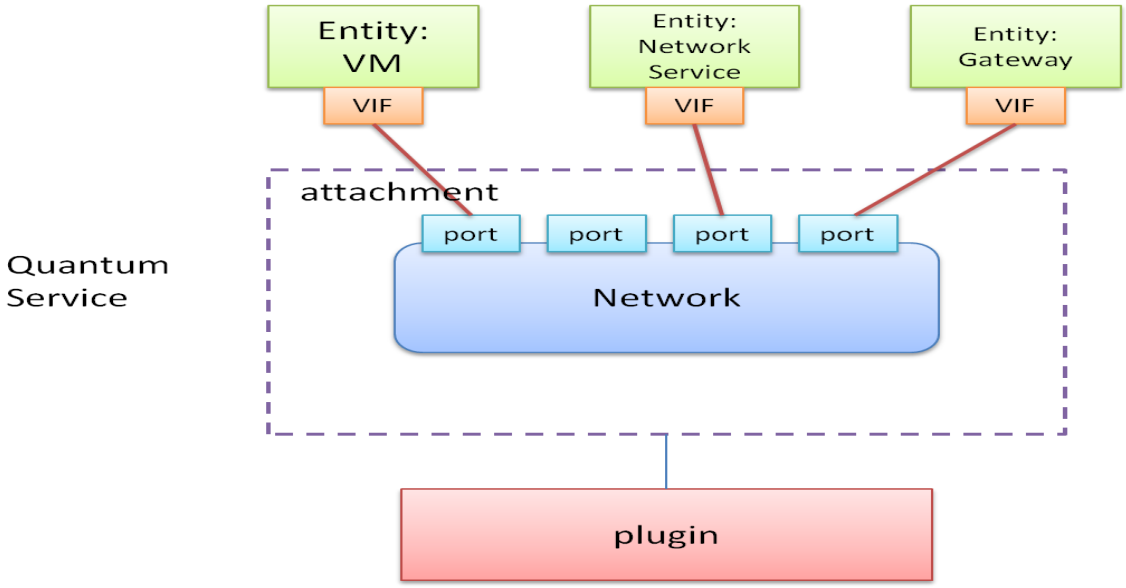

Figure 1. Quantum service's resources

Brief description for each of them. (Some of them are from community API document.)

Network: A virtual network providing connectivity between entities, i.e.: collection of virtual ports sharing network connectivity. In the Quantum terminology, a network is always a Layer-2 network.

Logical Port: A port on the virtual network switch represented by a virtual Layer-2 network.

Attachment: The relationship between the logical port and the Entity(VIF). If an entity is plugged into a port, the attachment relationship is created between them. At this stage, such relationship is 1:1.

Data Model

Before diving into the API details, we first show the Data Model of Quantum Service. This can give you the idea on what attributes/properties are associated with each resource. Meanwhile, it elaborates how the resources are related with each other.

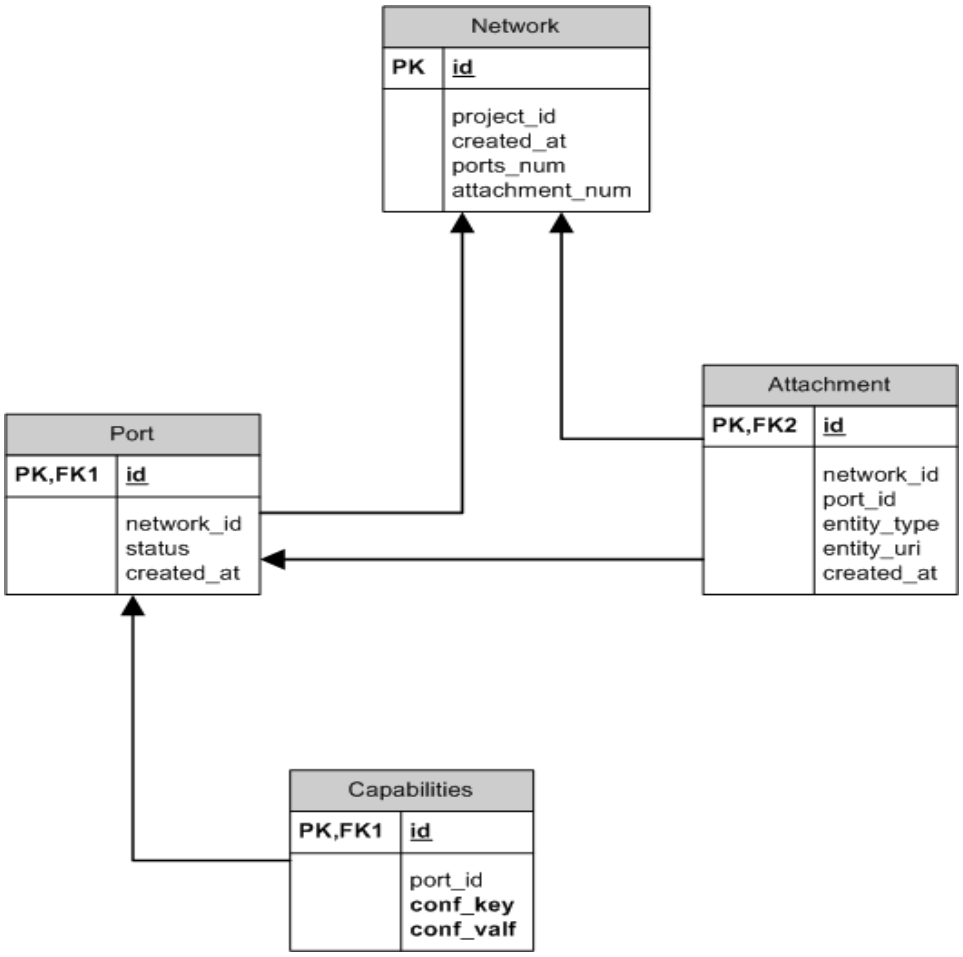

Figure 2. Data Model

API and extensions:

1. *Network related API*

Verb	URI	Request	Response	Comment
POST	/networks	Network Name[1]	Network Id	Create network
PUT	/networks/{net_id}	New Name		Update network
GET	/networks/{ all|net_id}	n/a	Network details or all available network	List all available network or network details
DELETE	/networks/{all|net_id}	n/a		Delete all or a specific network

Example response:

1.1 Create a network

XML:

```
<network>
        <id>1</id>
        <name>"My L2 Network"</name>
</network>
```

JSON:

```
{
    "network" : {
        "id" : 1,
        "name" : "My L2 Network"
        }
}
```

1.2 List a network details

XML

```
<network>
        <id> 1 </id>
        <name> "My L2 Network" </name>
```

[1] Discussion 1: Do we need to associate a network profile to a network? Or just leave network with one attribute “Name”.

```
        <created_at> 2011-05-18 18:30:40 </created_at>
        <num_available_ports> 6 </num_available_ports>
        <num_attached_ports> 4 </num_attached_ports>
</network>
```

JSON

```
{
    "network" : {
        "id" : 1,
        "name" : "My L2 Network",
        "created_at" : "2011-05-18 18:30:40",
        "num_available_ports" : 6,
        "num_attached_ports" : 4
        }
}
```

2. *Port related API:*

Verb	URI	Request	Response	Comment
POST	/networks/{net_id}/ports	OpenStack Port Profile	Port Id	Create port
PUT	/networks/{net_id}/ports/{port_id}	OpenStack Port Profile		Update port
GET	/networks/{net_id}/ports/{all | port_id}	n/a	Port list or port details	List all available ports or a port details
DELETE	/networks/{net_id}/ports/{all | port_id}			Delete all ports or a single port

Response Example:

2.1 OpenStack port profile:

XML

```
<port_profile>

        <conf key="acl"> permit ip any 209.165.201.2 255.255.255.255
        </conf>
        <conf key="vlan_segment"> 5 </conf>

</port_profile>
```

JSON

```
{
    "port_profile" : {
        "acl" : "permit ip any 209.165.201.2 255.255.255.255",
        "vlan_segment" : "5"
        }
}
```

2.2 List all the ports

XML

```
<port_list>
        <port_id> 001 </port_id>
        <port_id> 003 </port_id>
        <port_id> 004 </port_id>
        <port_id> 005 </port_id>
        <port_id> 006 </port_id>
        <port_id> 007 </port_id>
        <port_id> 008 </port_id>
</port_list>
```

JSON

```
{
    "port_list" : ["001","003","004","005","006","007","008"]
}
```

2.3 List a port details

XML

```
<port>
        <id> 8 </id>
        <created_at> 2011-05-18 18:35:40 </created_at>
        <status> Active </status>
        <configurations>
               <conf key=”acl”> permit ip any 209.165.201.2
255.255.255.255
               </conf>
               <conf key=”vlan_segment”> 5 </conf>
        </configurations>
        <statistics>
               <stat key=”max_bandwidth”> 10GB </stat>
               <stat key=”available_bandwidth”> 6GB </stat>
        </statistics>
</port>
```

JSON

```
{
    "port" : {
        "id" : 8,
        "name" : "My L2 Network",
        "created_at" : "2011-05-18 18:30:40",
        "status" : "Active",
        "configureations" : {
                "acl" : "permit ip any 209.165.201.2 255.255.255.255",
                "vlan_segment" : "5"
                }

        "statistics" : {
                "max_bandwidth" : "10GB",
                "available_bandwidth" : "6GB"
                }
        }
}
```

3. *Attachment related API:*

Verb	URI	Request	Response	Note
POST	/networks/{net_id}/ports/{port_id}/attachments	VIF URI	Attachment Id	Create an attachment on the specific port
POST	/networks/{net_id}/attachments	VIF URI	Attachment Id and attached port id	Create an attachment on a free port
PUT	/networks/{net_id}/ports/{port_id}/attachments	New VIF URI		
GET	/networks/{net_id}/ports/{port_id}/attachments/{att_id}		Attachment detail	Attachment details.
GET	/networks/{net_id}/attachments	n/a	Attachment list	List of attachment
DELETE	/networks/{net_id}/ports/{port_id}/attachments/{att_id}			Delete an attachment
DELETE	/networks/{net_id}/attachments			Delete all attachm

				ents associated with a network

Response Example

3.1 List an attachment detail

XML

```
<attachment>
        <id> 1 </id>
        <net_id> 1 </net_id>
        <port_id> 8 </port_id>
        <created_at> 2011-05-18 18:30:40 </created_at>
        <entity_type> Virtual Machine </entity_type>
        <entity_uri> URI to locate the VM[2] </entity_uri>
</attachment>
```

JSON

```
{
    "attachment" : {
        "id" : 1,
        "net_id" : 1,
        "port_id" : 8,
        "created_at" : "2011-05-18 18:30:40",
        "entity_type" : "Virtual Machine",
        "entity_uri" : "URI to locate the VM"
        }
}
```

Discussion

1. Do we need a network profile?
   Currently, there is one attribute “Name” associated with a network. All the configurations are associated with the port. Do we need to define a network profile, which includes configuration and ports to be created?

2. We introduce OpenStack port profile. It includes all the configurations associated with a port. The configurations are list of <key, value> pairs. When a port is

[2] Need to discuss: how to locate the VM and what’s the URI format we should use?

created, an OpenStack port profile can be specified. If not, the default configurations are created for the port.

3. Port profile is extensible. A new key can be defined. However, a set of “Reserved Key” should be defined, for example “acl” or “vlan_segment”

Reference

Quantum core API document
http://wiki.openstack.org/QuantumAPIBase?action=AttachFile&do=view&target=Quantum_API_spec-draft-v0.11.pdf