Structural Mount

draft-bjorklund-netmod-structural-mount-00

IETF NETMOD Virtual Interim
2016-02-22

Martin Björklund
mbj@tail-f.com

The Problem

Two extension methods in YANG:

- “uses” of “groupings”
  - *Explicit addition of nodes from a source*
- “augment”
  - *Explicit addition of nodes to a target*

Neither is suitable for the “logical network element” problem.

Solution - Structural Mount

Decouple the definition of the relation between the source and target modules from the modules.

1. Define a *mount point* in the data model:

```
container logical-network-elements {
    list logical-network-element {
        key name;
        ...
        yangmnt:mount-point lne-root;
    }
}
```

2. The server lists the models it supports per mount point:

```
 +--ro mount-points
    +--ro mount-point* [module name]
       +--ro module                    yang:yang-identifier
       +--ro name                      yang:yang-identifier
       +--ro (data-model)
          +--:(inline-yang-library)
          +--:(modules)
```

Advantages

- Supports different set of models in different instances of a mount point (different LNEs may have different models).
- Supports recursive mounts (an LNE may implement modules that contain other mount points).
- Supports rpcs and notifications in the mounted modules.

Compare to YSDL

With structural mount, the mount points are explicitly defined in the data model. With YSDL, models can be mounted anywhere.

With structural mount the intention of the data model designer is explicitly expressed, but this means that it is less flexible than YSDL.

YSDL is more flexible, but also more costly to implement, esp. for clients – they have to be prepared that a server “mounts” any model anywhere in the node hierarchy.