

# Needing an extensible Mount syntax across Schema, Alias, & Peers

## IETF 95

Eric Voit, Alex Clemm  
evoit@cisco.com, alex@cisco.com  
April 4<sup>th</sup> 2016

# Different types of YANG Mount

`+-> schema (tree) mount (a.k.a. structural mount)`

`|`

`yang mount ->`

`|`

`+-> local data tree (alias) mount`

`+-> data (tree) mount ->`

`|`

`+-> remote data tree (peer) mount`

# Purposes

- YANG Mount** Abstract concept of incorporating a YANG-defined data tree or schema tree (the mounted data or schema tree) into an existing YANG-defined data tree or schema tree (the parent data tree)
- Schema Mount** Enables reuse of existing model definitions to facilitate implementation of alternative model structures in multiple contexts
- Alias Mount** Enables multiple concurrent context specific object models without replicating instance data. This allows programmers access common objects via the hierarchies that best make sense to them. (E.g. Symbolic link in a file system.)
- Peer Mount** Provides an on-demand virtual federated datastore that provides visibility and location transparency of YANG-data across device boundaries. Retains clear authoritative ownership. Abstracts away tasks needed to synchronize, reconcile aggregate, validate information replicated across these devices. (E.g. File share to remote host in a file system.)

# Differences & Commonalities

- Schema-mount (Structural Mount and YSDL) are about mounting definitions
  - Reuse an existing module definition by a new module
  - Analogy: “grouping” (Yang module) – “uses” (schema mount)
  - Requires a YANG language extension
- Data-mount is about mounting instances
  - Alias-Mount: Define an alternative path to instance information already elsewhere in the tree
  - Peer-Mount: Allow to access instance information authoritatively owned by a different server
  - Requires YANG language and Server infrastructure extensions
- Commonality between Data-Mount and Schema-Mount → the YANG mountpoint extension
  - Possible to have an extensible YANG language construct for YANG mountpoints
  - Differences in terms of additional parameters (to identify target node and target system)

# Peer Mount & Alias Mount drafts

Requirements **draft-voit-netmod-yang-mount-requirements-03**

(with Sander Mertens, recently refreshed)

Technical spec **draft-clemm-netmod-mount-04**

(with Jan Medved, recently refreshed)

- What's new since IETF94

Alias-mount Definition: inserting a subtree under a mountpoint, to provide an alternative path within a device

Peer-mount Deployment: OpenDaylight mounting of subtrees resides on a remote system

795 google results for "mount" on site:opendaylight.org (as of 4-Apr-2016)



- We continue to update drafts

WG should adopt peer-mount if there looks to be interest/proliferation beyond OpenDaylight

WG can adopt alias-mount if there is interest, even if WG remains divided on if/when to adopt peer-mount

# FYI: Peer mount example

```
rw controller-network
  +-- rw network-elements
    +-- rw network-element [element-id]
      +-- rw element-id
      +-- rw element-address
      |   +-- ...
      +-- M interfaces
```

**Module structure**

```
...
list network-element {
  key "element-id";
  leaf element-id {
    type element-ID;
  }
  container element-address {
    ...
  }
  mnt:mountpoint "interfaces" {
    mnt:target "./element-address";
    mnt:subtree "/if:interfaces";
  }
}
...
```

**Mountpoint declaration**

- YANG module defines YANG mount extensions + data model for mountpoint management

- YANG extensions:

Mountpoint: Defined under a containing data node (e.g. container, list)

Target: References data node that identifies remote server [peer-mount only]

Subtree: Defines root of remote subtree to be attached

```
<network-element>
  <element-id>NE1</element-id>
  <element-address> .... </element-address>
  <interfaces>
    <if:interface>
      <if:name>fastethernet-1/0</if:name>
      <if:type>ethernetCsmacd</if:type>
      <if:location>1/0</if:location>
      ...
    </if:interface>
    ...
  </network-element>
```

**Instance information**

# FYI: Alias mount example

```
rw my-container
  +-- rw sub-container
    +-- M interfaces
  ...
```

**Module  
structure**

```
container my-container {
  container subcontainer {
    mnt:mountpoint "interfaces" {
      mnt:subtree "/if:interfaces";
    }
  }
  ...
}
```

**Mountpoint declaration**

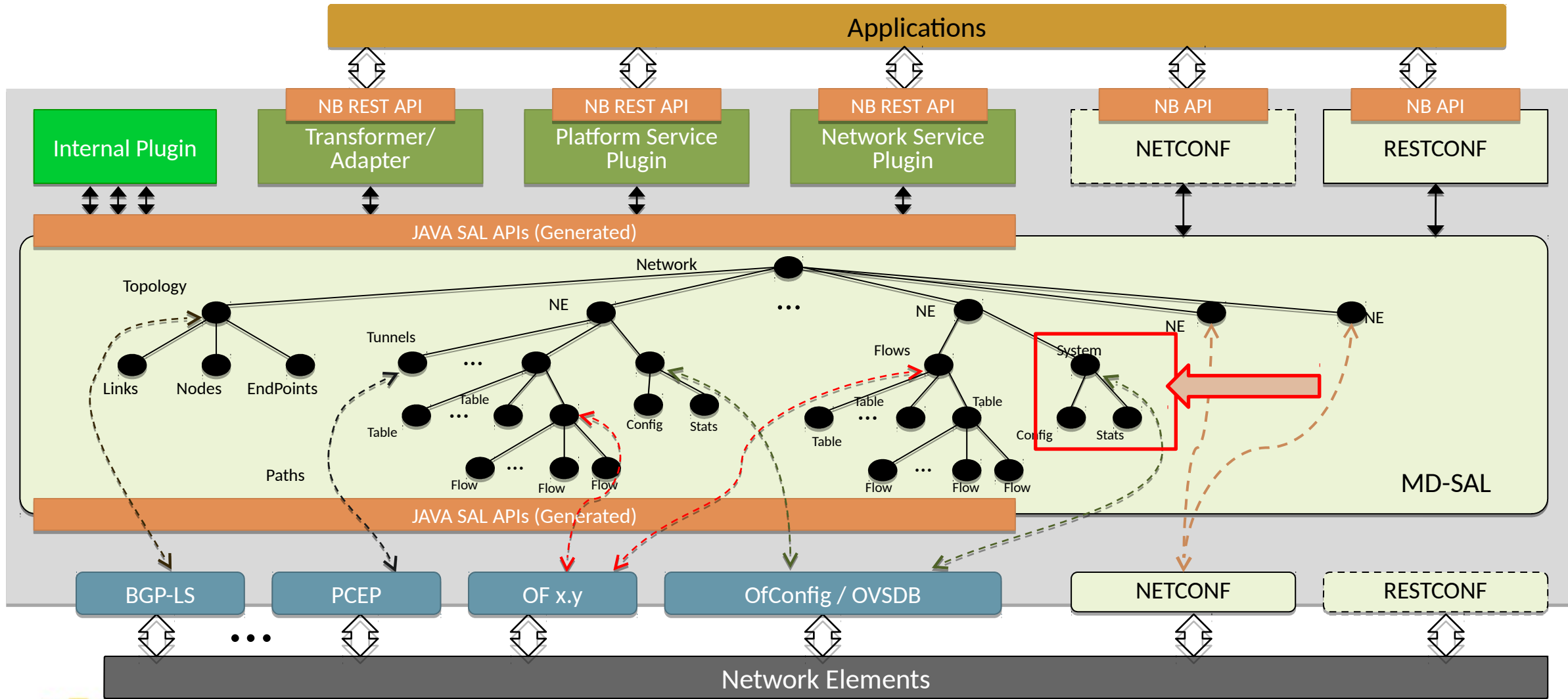
Mount point is local



```
<my-container>
  <sub-container>
    <interfaces>
      <if:interface>
        <if:name>fastethernet-1/0</if:name>
        <if:type>ethernetCsmacd</if:type>
        <if:location>1/0</if:location>
        ...
      </if:interface>
    ...
  ...
```

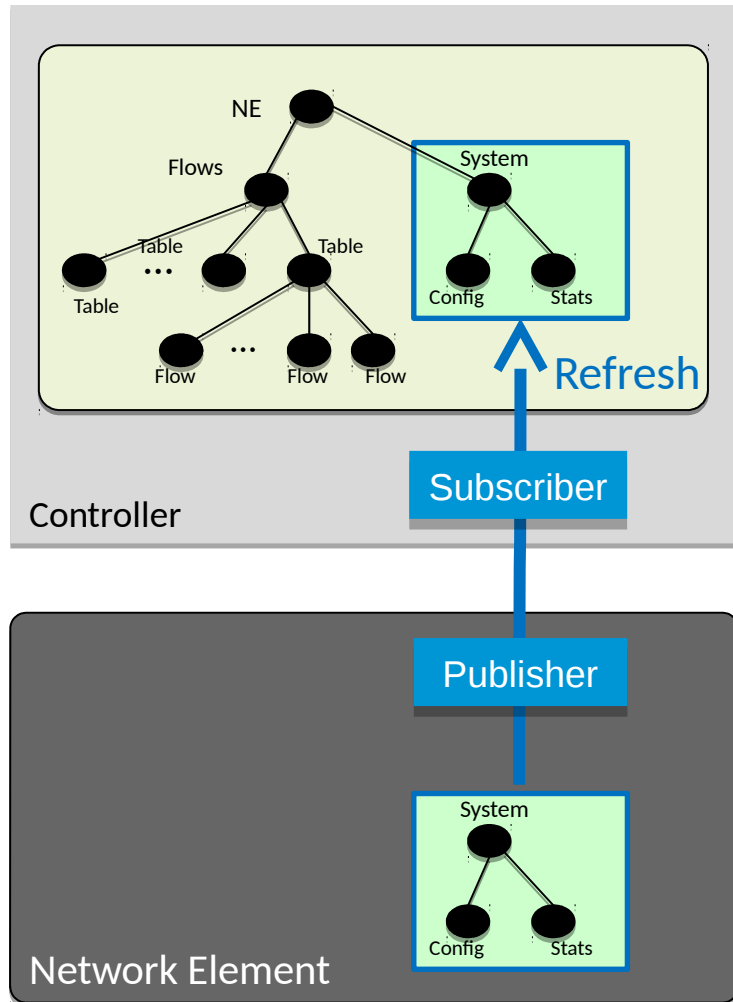
**Instance information**

# FYI: Peer mount in OpenDaylight: Model-Driven SAL





# FYI: Peer Mount can optionally have YANG Push underneath to improve scale/response time



- Transparent caching mechanisms + YANG subscriptions speed performance if needed
  - *Periodic*: Object update passed every 'X' seconds
  - *On Change*: as Mounted Objects change
- For more info see drafts
  - draft-ietf-i2rs-pub-sub-requirements
  - draft-ietf-netconf-yang-push
  - draft-voit-netconf-restconf-yang-push

# Next steps (asking for your feedback)

- Explore possible alignment between Peer, Alias, & Schema-Mount

Pursue independently, or consider extensible YANG Mount syntax?

# Backup

# Mount Definitions & Purposes

	<b>Definition</b>	<b>Purpose</b>
YANG Mount	The abstract concept of incorporating a YANG-defined data tree or schema tree (the mounted data or schema tree) into an existing YANG-defined data tree or schema tree (the parent data tree)	Provides model flexibility by enabling the growth of YANG trees via an explicit reference to other YANG information and structures.
Schema Mount	A type of YANG Mount where a new YANG Schema is constructed by inserting any existing YANG schema under a parent model within a local datastore. Objects populated into the mounted schema are only instantiated as part of the parent's hierarchy.	This allows reuse of existing model definitions to facilitate implementation of alternative model structures in multiple contexts. In effect, it allows to define models which reuse other model definitions as if they had been defined as a special kind of grouping.
Alias Mount	A type of YANG Mount which allows to provide an alternative path to local objects of YANG data.	Enables multiple concurrent context specific object models without replicating instance data. This allows programmers access common objects via the hierarchies that best make sense to them. (E.g. Symbolic link in a file system.)
Peer Mount	A type of YANG Mount which enables access to remote objects as if they were contained within a local datastore dynamically.	Provides an on-demand virtual federated datastore that provides visibility and location transparency of YANG-data across device boundaries. Retains clear authoritative ownership. Eliminates many tasks otherwise needed to synchronize, reconcile aggregate, validate information replicated across these devices. (E.g. File share to remote host in a file system.)