NMDA Protocol Transition Issue Discussion

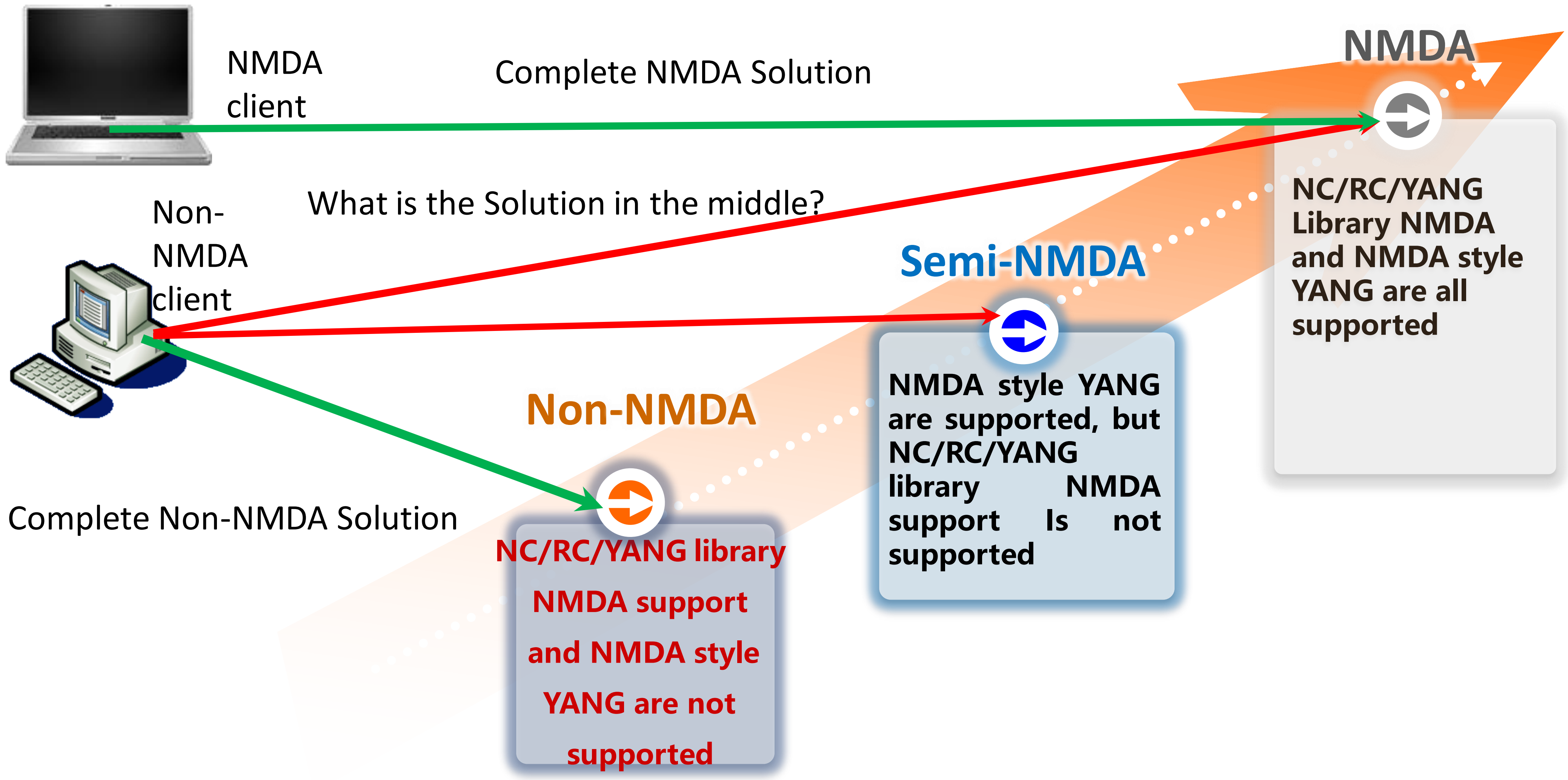
https://mailarchive.ietf.org/arch/msg/netmod/CYMK1cdLp5byiAkwDjaBngcTDQo

Qin Wu （Presenter）

Recap

- NETCONF/RESTCONF/YANG library NMDA support has been published as RFCs. Most of IETF developed models are NMDA compliant and new model under development by IETF will be NMDA complaint. But temporary non-NMDA version of modules still exist to bridge the gap of time period until NMDA implementations are available.
- There is a transition period before NMDA solutions are universally available and most of NETCONF clients in existing deployment doesn’t support NMDA.
- **NMDA YANG transition guideline has been provided in RFC8407, However it doesn’t provide clear guideline for NMDA protocol transition and how NMDA protocol work with NMDA module and non-NMDA module.**

The road from non-NMDA to NMDA



NMDA client

Complete NMDA Solution

NMDA

NC/RC/YANG Library NMDA and NMDA style YANG are all supported

Non-NMDA client

What is the Solution in the middle?

Semi-NMDA

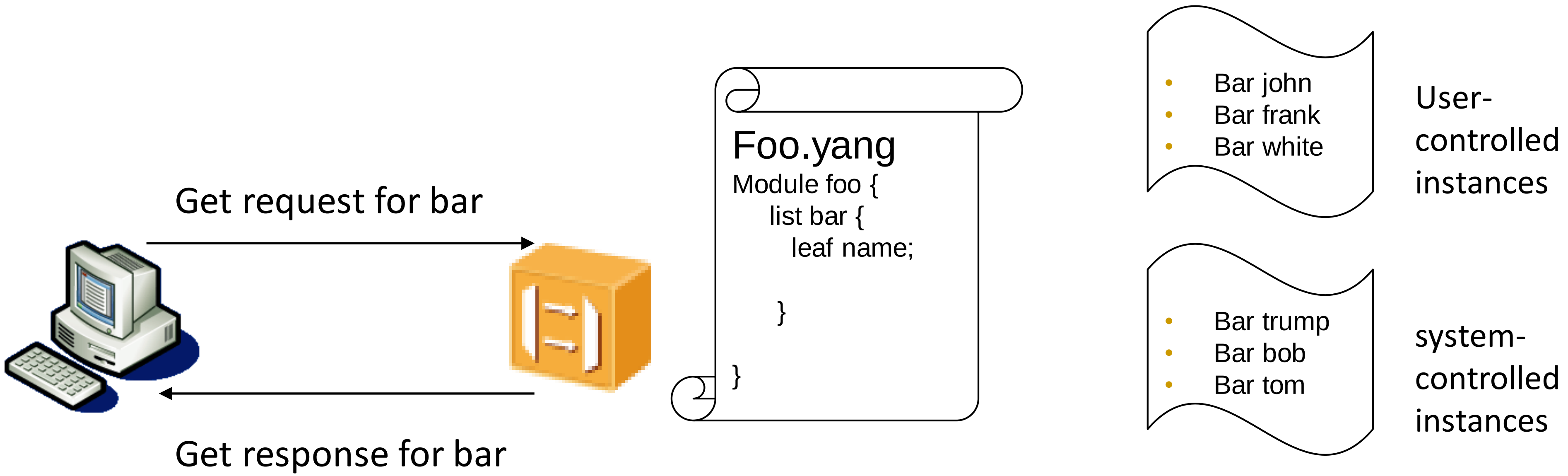
NMDA style YANG are supported, but NC/RC/YANG library NMDA support Is not supported

Non-NMDA

Complete Non-NMDA Solution

NC/RC/YANG library NMDA support and NMDA style YANG are not supported

Non-NMDA client vs. semi/full NMDA server



Get request for bar

Get response for bar

```
Foo.yang
Module foo {
    list bar {
        leaf name;

     }
}
```

- Bar john
- Bar frank
- Bar white

User-controlled instances

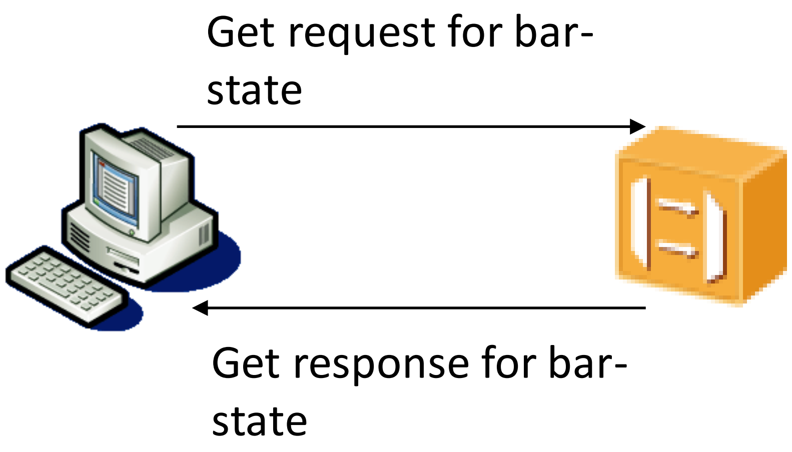
- Bar trump
- Bar bob
- Bar tom

system-controlled instances

```
<data>
  <foo:bar>
     <name>john</name>
  </foo:bar>
 <foo:bar>
     <name>frank</name>
  </foo:bar>
 <foo:bar>
     <name>white</name>
  </foo:bar>
</data>
```

ISSUE:Get operation can not retrieve system-controlled instances from data based NMDA style YANG modules, because NMDA style YANG module has no state branch copy.

Solution1:reserve/add state copy nodes



Foo.yang

```
Module foo {
    list bar {
      leaf name;
     }
    list bar-state {
       config false;
       leaf name;
    }
}
```

- Bar john
- Bar frank
- Bar white

User-controlled instances

- Bar trump
- Bar bob
- Bar tom

system-controlled instances

```
<data>
  <foo:bar-state>
     <name>john</name>
  </foo:bar-state>
 <foo:bar-state>
     <name>frank</name>
  </foo:bar-state>
 <foo:bar-state>
     <name>white</name>
  </foo:bar-state>
```

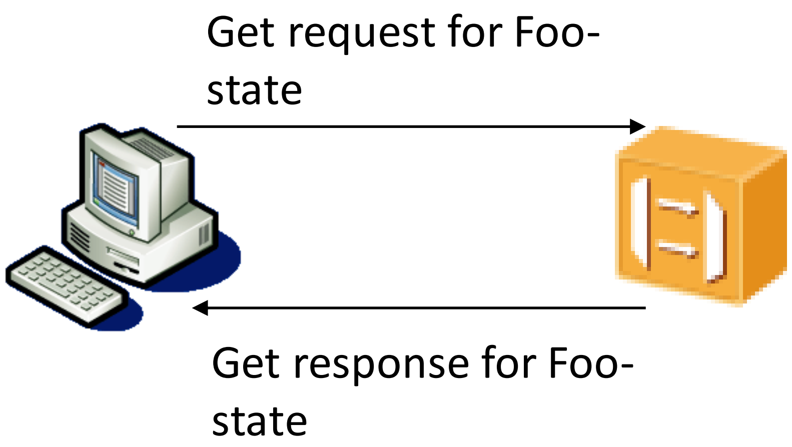
```
<foo:bar-state>
     <name>trump</name>
  </foo:bar-state>
 <foo:bar-state>
     <name>bob</name>
  </foo:bar-state>
 <foo:bar-state>
     <name>tom</name>
  </foo:bar-state>
</data>
```

Solution: If NMDA style YANG module has deprecated state copy, keep it. If NMDA style YANG module has no state copy, add it (only object who has system-controlled instances should be added as state copy).

Pros: provide a way to access system-controlled instances.

Cons: YANG modules which has no state copy must be modified, but a published standard YANG MUST NOT be modified.

Solution2:add state copy Modules



Get request for Foo-state

Get response for Foo-state

Foo-state.yang

```
Module foo-state {
    list bar {
        config false;
        leaf name;
    }
}
```

- Bar john
- Bar frank
- Bar white

User-controlled instances

- Bar trump
- Bar bob
- Bar tom

system-controlled instances

```
<data>
  <foo:bar>
     <name>john</name>
  </foo:bar>
 <foo:bar>
     <name>frank</name>
  </foo:bar>
 <foo:bar>
     <name>white</name>
  </foo:bar>
```

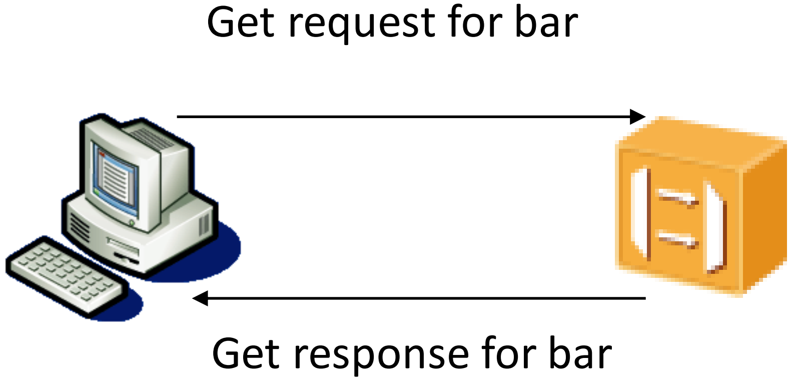
```
<foo:bar>
     <name>trump</name>
  </foo:bar>
 <foo:bar>
     <name>bob</name>
  </foo:bar>
 <foo:bar>
     <name>tom</name>
  </foo:bar>
</data>
```

Solution: If a NMDA style YANG module has no state copy, make a state copy YANG module(only YANG modules which has system-controlled instances should be made it's state copy).

Pros: provide a way to access system-controlled instances. Original YANG modules MAY not be modified.

Cons: state copy YANG modules are private, while it's original YANG module is standard. It cause some difficulties when client interact with server. For example, client MUST recognize, load and process private state copy YANG modules.

Solution3: change get operation’s definition



Get request for bar

Get response for bar

Foo.yang

```
Module foo {
    list bar {
       leaf name;
    }
}
```

- Bar john
- Bar frank
- Bar white

User-controlled instances

- Bar trump
- Bar bob
- Bar tom

system-controlled instances

```
<data>
  <foo:bar>
      <name>john</name>
  </foo:bar>
 <foo:bar>
      <name>frank</name>
  </foo:bar>
 <foo:bar>
      <name>white</name>
  </foo:bar>
```

```
<foo:bar>
       <name>trump</name>
   </foo:bar>
 <foo:bar>
       <name>bob</name>
   </foo:bar>
 <foo:bar>
       <name>tom</name>
   </foo:bar>
</data>
```

Solution: change get operation’s definition:
OLD: Retrieve running configuration and device state information.
NEW: Retrieve running configuration including system configuration and device state information.

Pros: get operation can retrieve system-controlled instances from config data node(list), original YANG modules NEED NOT be modified.

Cons: This change maybe cause unknown influence.

Open Issue: NMDA transition needed?

- Question: Shall we support NMDA module first and then support NC/RC/YANG library NMDA support management protocol? Or Support NMDA module and NMDA management protocol at the same time?
- NMDA transition Options:
  - Option 1: Agree there is a NMDA transition period but not need complete standard solution
  - Option 2: Agree there is a NMDA transition period but need complete standard solution
  - Option 3: NMDA transition period is not required. Be Bold to move toward complete standard NMDA solution?