NMDA protocol operation Backwards-Compatibility with Legacy Devices

draft-wu-netconf-nmda-compatibility-01

Qin Wu (bill.wu@huawei.com)

Chong Feng (frank.fengchong@huawei.com)

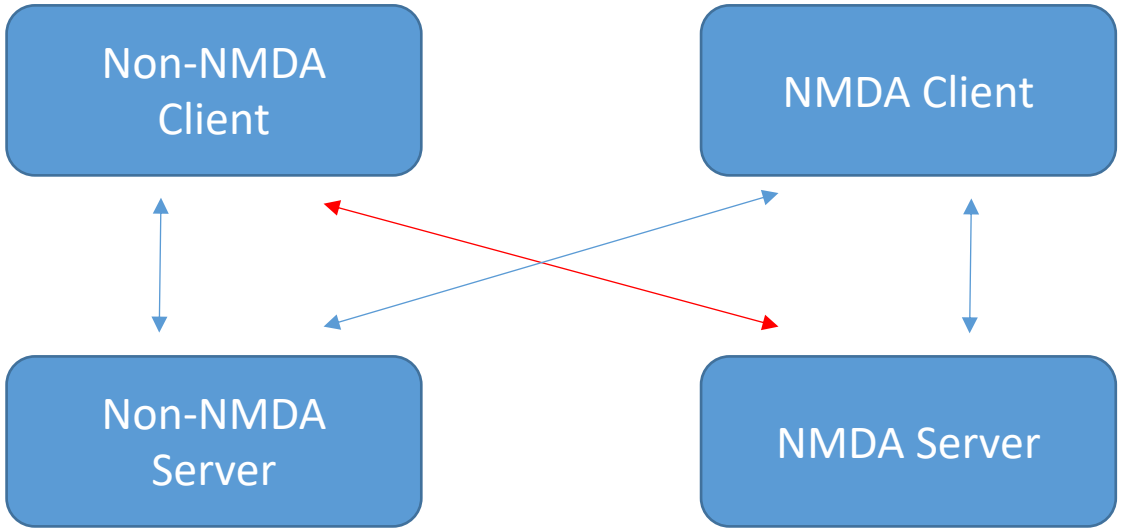
Michael Wang (wangzitao@huawei.com)

Goals

- Objectives & Motivations
  - The NMDA has been published as RFC8342, and the protocol extensions are work in progress;
  - Found some backward compatibility issues when we deployed NMDA
  - Sharing these problems with WG, and solicit comments and suggestions
  - Investigate reasonable solutions if the WG agree with these problems.

Assumptions

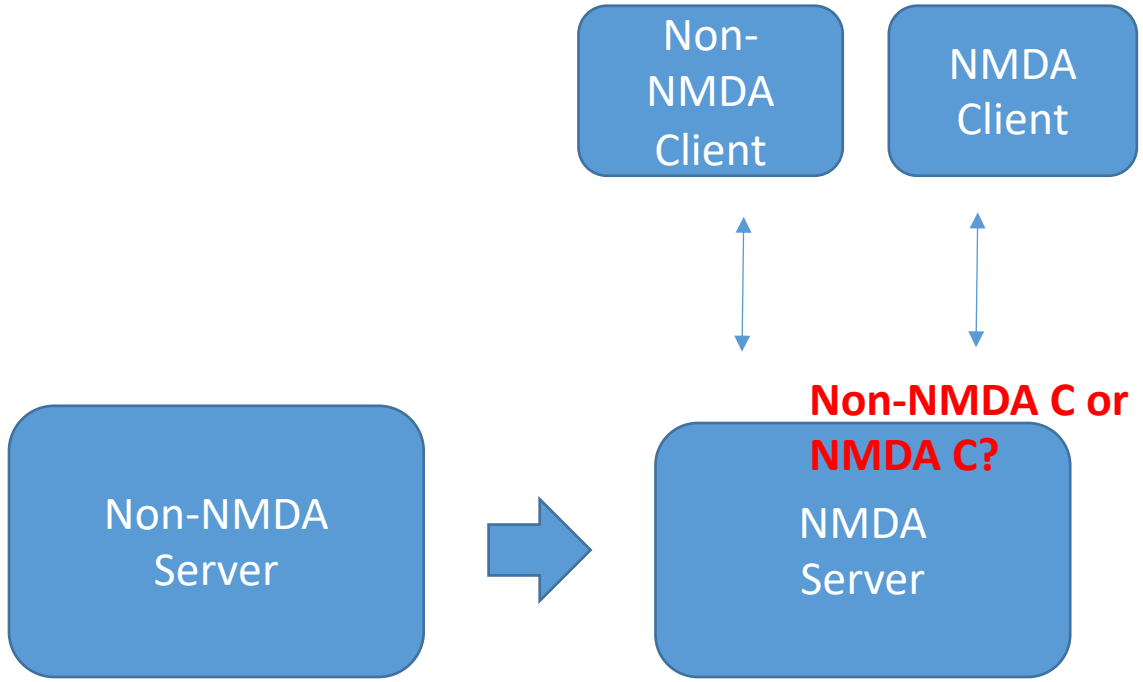
- The NMDA Client, Non-NMDA Client, NMDA Server, and Non-NMDA Server will coexist in period of time. RFC6241 is widely implemented and not obsoleted.
  - The NMDA Client can use conventional operation (i.e. get, get-config) to communicate with Non-NMDA Server;
  - But there are some problems when Non-NMDA Client want to retrieve data from NMDA Server ..



Problem Illustrated # 1: Client NMDA support indication

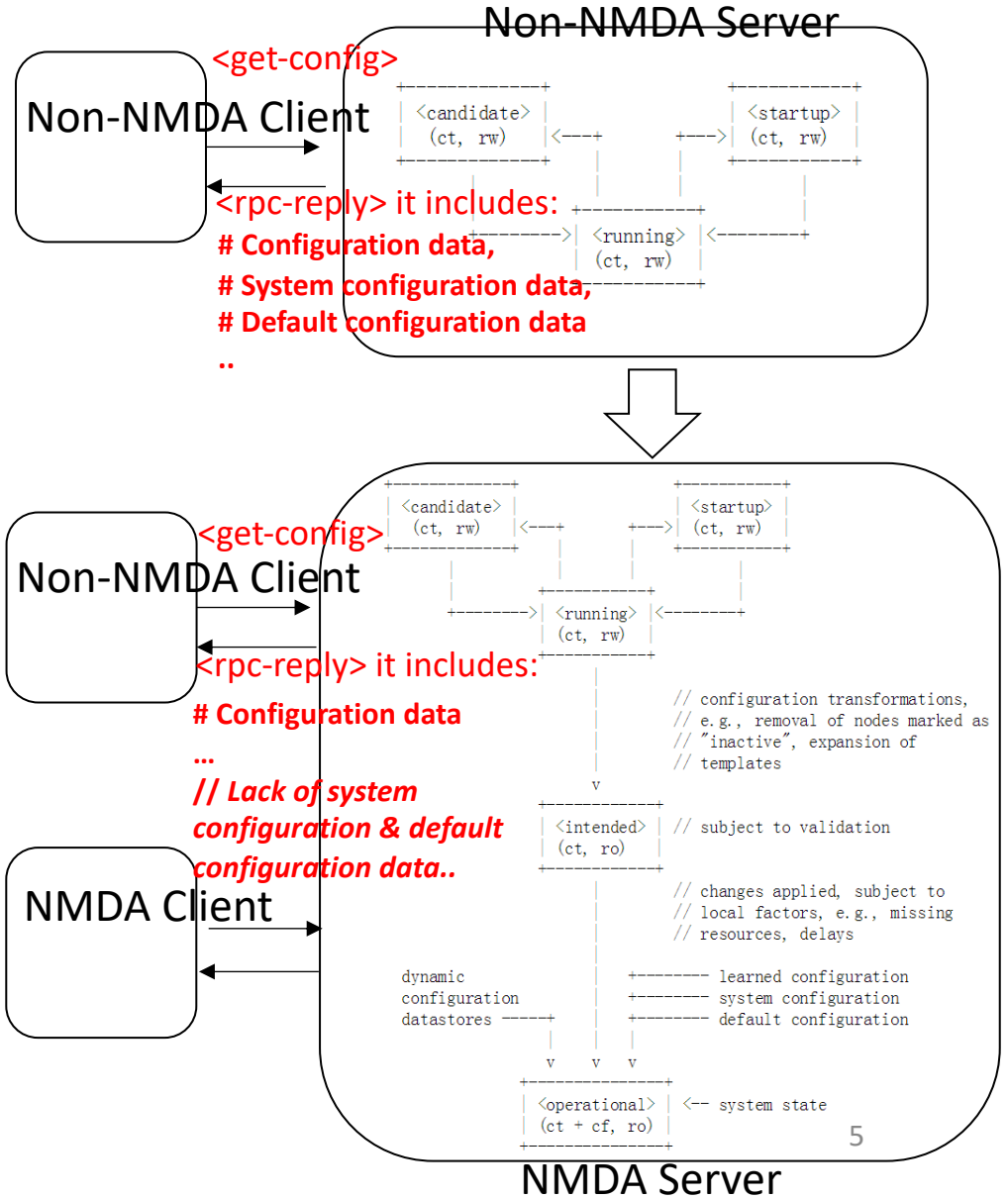
- When a server is upgraded to NMDA aware server and needs to support both NMDA client and non-NMDA clients,

- There is no standards-based way for the server to know whether the client supports NMDA.
  - Suggestions: Client NMDA support should be indicated by protocol operations.
    - If <get>/<get-config>/<edit-config> operation is received from the client, the server should assume the client is Non-NMDA client.
    - If <get-data>/<edit-data> operation is received from the client, the server should assume the client is NMDA client.



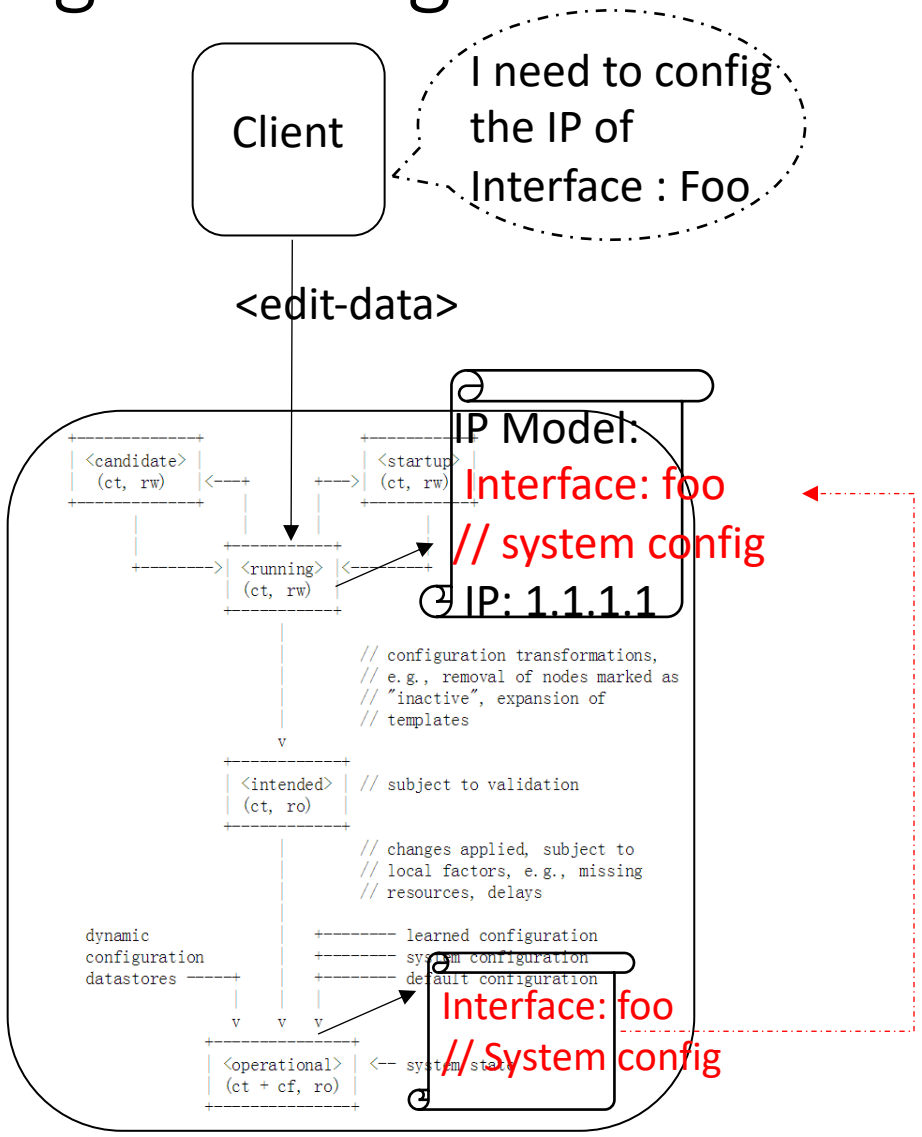
Problem Illustrated # 2: Default data handling

- **Assumption:** System configuration & default configuration originally part of conventional configuration datastores have been explicitly separated and moved to <operational> datastore under NMDA.
- It is not clear whether the NMDA aware server can return the same results to non-NMDA clients as non-NMDA-aware server does..
  - If yes, impact on <get-config>, <get>, <get-data> and default handling behavior:
    - <get>: almost no impact, return content of <running> and system config, default config from <operational>
    - <get-data>: If target datastore is <running>, the default config in case of report all retrieval should not be reported unless explicitly set by the client.
    - <get-config>: in report-all mode ,the data retrieved on the <running> datastore will be reduced without including default configuration unless explicitly set by the client.
    - **Default handling behavior**: Report all mode can not report default configuration unless explicitly set by the client
    - <edit-config>:no impact and follow default handling behavior in RFC6243
    - <edit-data> on <running>: 'create' default config succeed while 'delete' default config fails.



Problem Illustrated # 3: System config handling

- In some case, further configuration is needed within the system configuration
  - e.g., configure IP address of the interface after such interface configuration is auto-created (i.e.,system configuration), such auto-created configuration needs to set by the client
  - since int config as system confi doesn't exist in the conventional configuration datastore after NMDA is introduced.
- The effect is the same as feed auto-created interface configuration into running datastore and make it become client set configuration.
- After the interface configuration is applied, it will be merged with the other existing system configuration in the <operational> datastore.
- **It is still not clear whether the NMDA aware server can return the same results to non-NMDA clients as non-NMDA-aware server does..**
  - If yes, impact on <edit-config>, <edit-data>
    - <edit-config> on <running>: return error to indicate system config exist or return ok to indicate system exist doesn't exist???
    - <edit-data> on <running>: return ok to indicate system config does'n exist.



Q & A

Or talk to us on the mailing list

Thank You !