

System configuration data handling behavior

netmod

draft-ma-net~~conf~~-with-system-02

Feng Chong(frank.fengchong@huawei.com)

Qiufang Ma(maqiufang1@huawei.com) **Presenter**

Chongfeng Xie(xiechf@chinatelecom.cn)

Recap

- Motivation

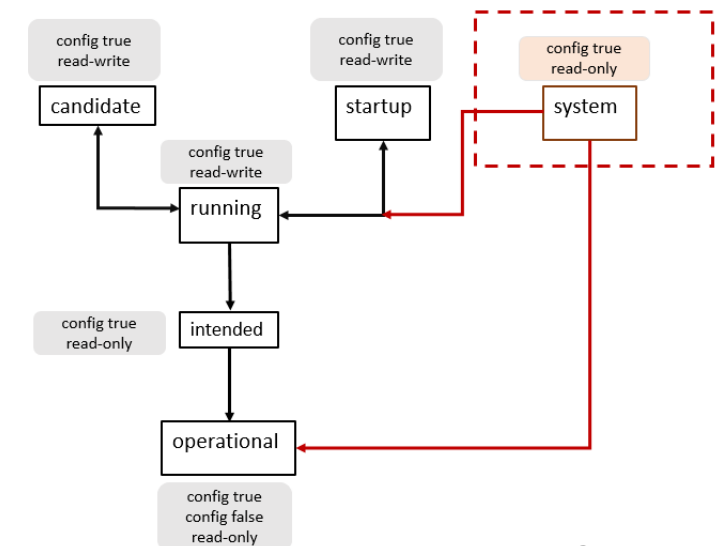
- System configs can be located anywhere & there is no standard system config data handling behavior in Non-NMDA;
- System configs are only present in <operational> and cannot be referenced(e.g., leafref, when, must) directly by the operators in NMDA.
 - There is no way to detect the referenced system configs update and synchronize the update into <running>.
 - Some of the system configs are predefined as a convenience but the operators still need to create the system config data in <running> in this case.

- Goal

- To define a standard system configuration data handling behavior

- Solution Overview

- A new datastore “system” is introduced
- Two system config data handling modes are defined
 - Auto-populate: load <system> into <running> automatically when the device is powered on or the physical resource is present
 - No-populate: will not load <system> into <running> automatically

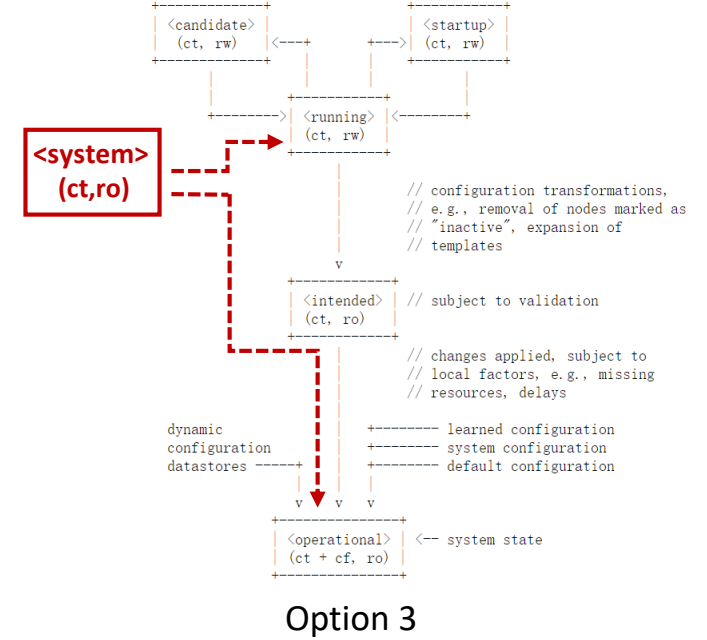
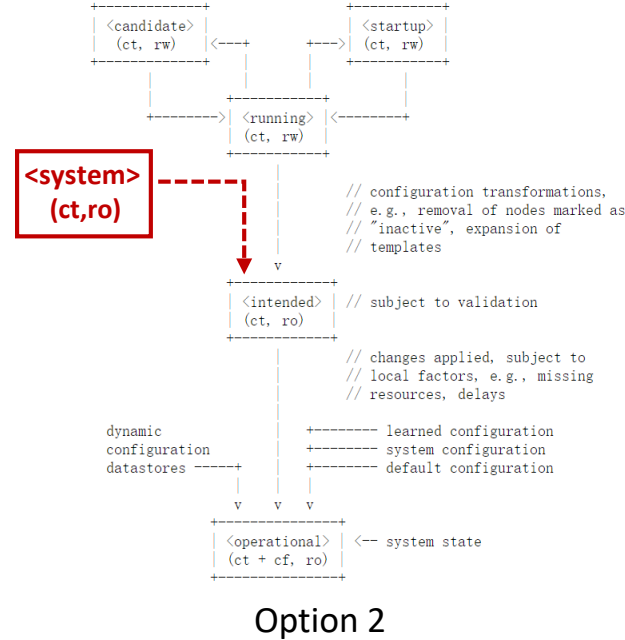
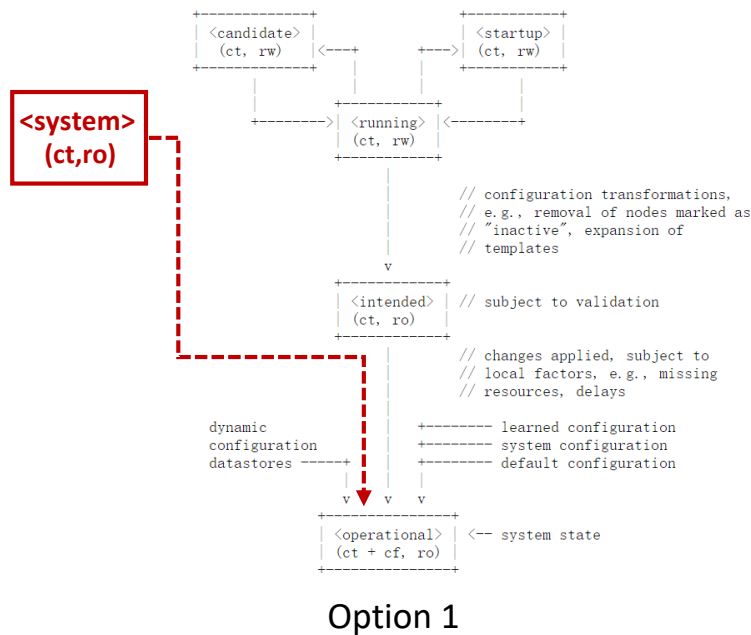


Document Status

- This draft was first presented in the IETF 110 NETCONF session and has also been discussed on the NETMOD list recently. We have received so many valuable comments and suggestions, thanks a lot!

- High-level updates since IETF110:
 - Remove system config data retrieval behavior in the main body and examples in the appendix
 - Change the basic-mode values : report-all->**auto-populate**, explicit->**no-populate**
 - Consider <factory-default> to work together with <system> and the impact of the “factory-reset” RPC operation on <system>

Which datastore should <system> be merged into?



- cannot go through the system-defined template expansion;
- Neither <intended> nor <running> can pass validation due to missing references.
- System-defined templates in <system> can be expanded and present in <intended>;
- Enables <intended> to pass validation (Please note that <running> itself cannot be valid here).
- Enables <running> itself to be valid as the presence of referenced system configs;
- Unreferenced system configs may also be merged into <operational> directly.

If <running> MUST also hold the referenced system configs to be valid...

- How to synchronize the system config into <running> if it must be copied firstly to make sure a successful validation for <running>?
 - 1). Auto-populate
 - PRO(s)
 - Efficient, as there is no need to retrieve the referenced system config from <system>/<operational> and then copy it into <running> manually
 - Ready to be referenced all the time
 - CON
 - Might violate the definition of <running> and NMDA since <running> should be controlled by the operators
 - 2). An RPC operation to synchronize the entire <system> into <running>
 - PRO
 - The operator-driven, few- or one-shot
 - CON
 - Some unnecessary system config may also exist in <running>
 - 3). Existing mechanism, e.g., edit-config
 - PRO
 - Only the most necessary system configs are present in <running>
 - CON(s)
 - Cumbersome operations
 - May lose the initial intention of system predefinition and bringing convenience for the operators

Follow up

- More examples about system configurations will be worked on.
- Contributions/coauthors are welcome! :-)