Principles for populating a list of data nodes using a template technique

draft-rajaram-netmod-yang-cfg-template-framework-01

D. Rajaram (NOKIA), R. Peschi (NOKIA), S. Ashraf (NOKIA)

IETF-122 NETMOD

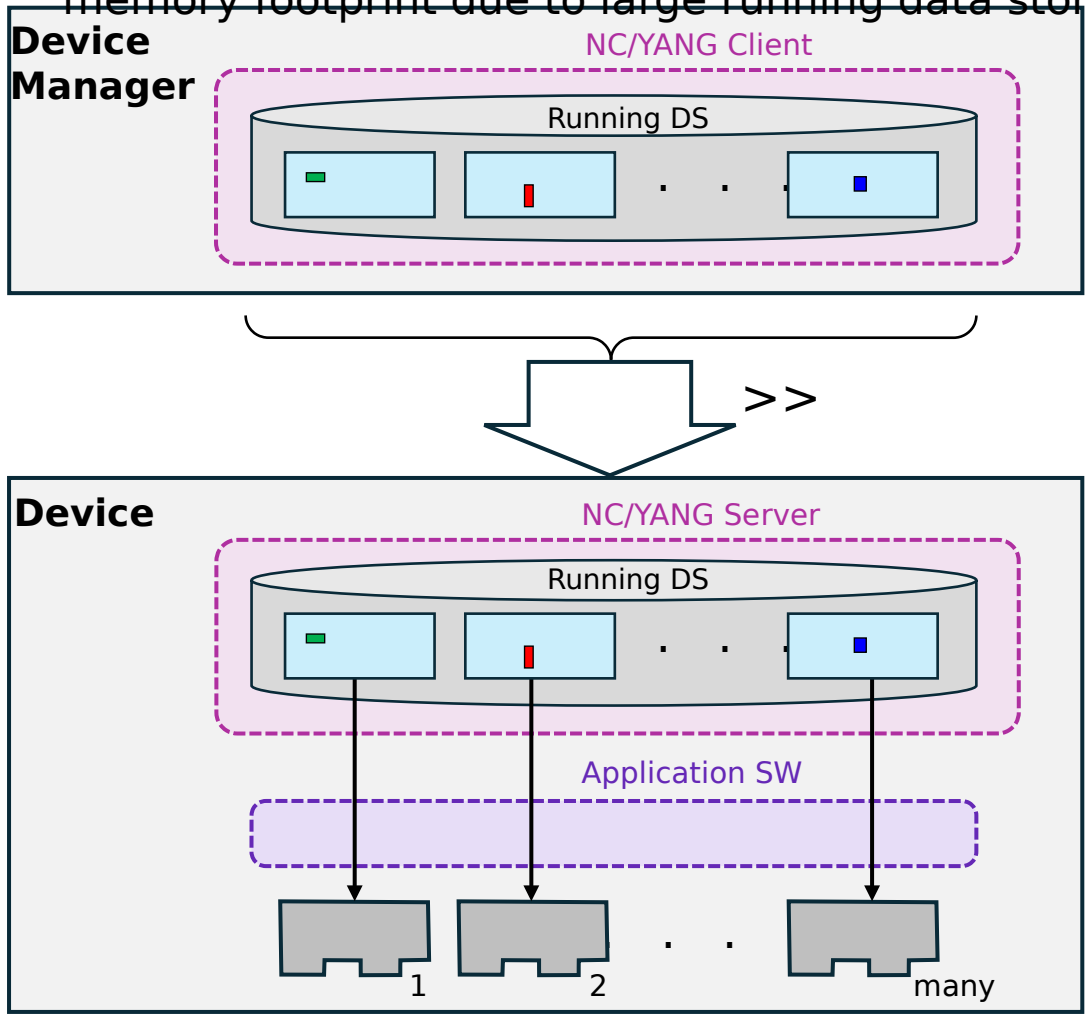
Executive summary

A **template implementation method** to configure YANG managed devices

- This is a short-term implementation method and is *not* an alternative to the template solution being discussed in netmod.
- “YANG managed Device” = entity managed through a YANG model, typically hosting a Netconf YANG Server.
- Suitable when device contains many instances of same function with limited variations in their config. values (standard device contains a HW sub-system replicated many times)
- Useful when a host device also manages lot of peripheral devices using the same data model.
- Relies on existing YANG specification and off-the-shelf tools.
- Explains what it would take to apply this template technique for data nodes from existing IETF models. ietf-interfaces.yang taken as example.

YANG configuration of the device: traditional way

- Traditionally a device manager (e.g. NetConf Client) configures the device with each data node of each instance, one by one
- This is inconvenient: it can be error prone, can lead to long device configuration time, causes large device memory footprint due to large running data store



Example of traditional YANG tree:

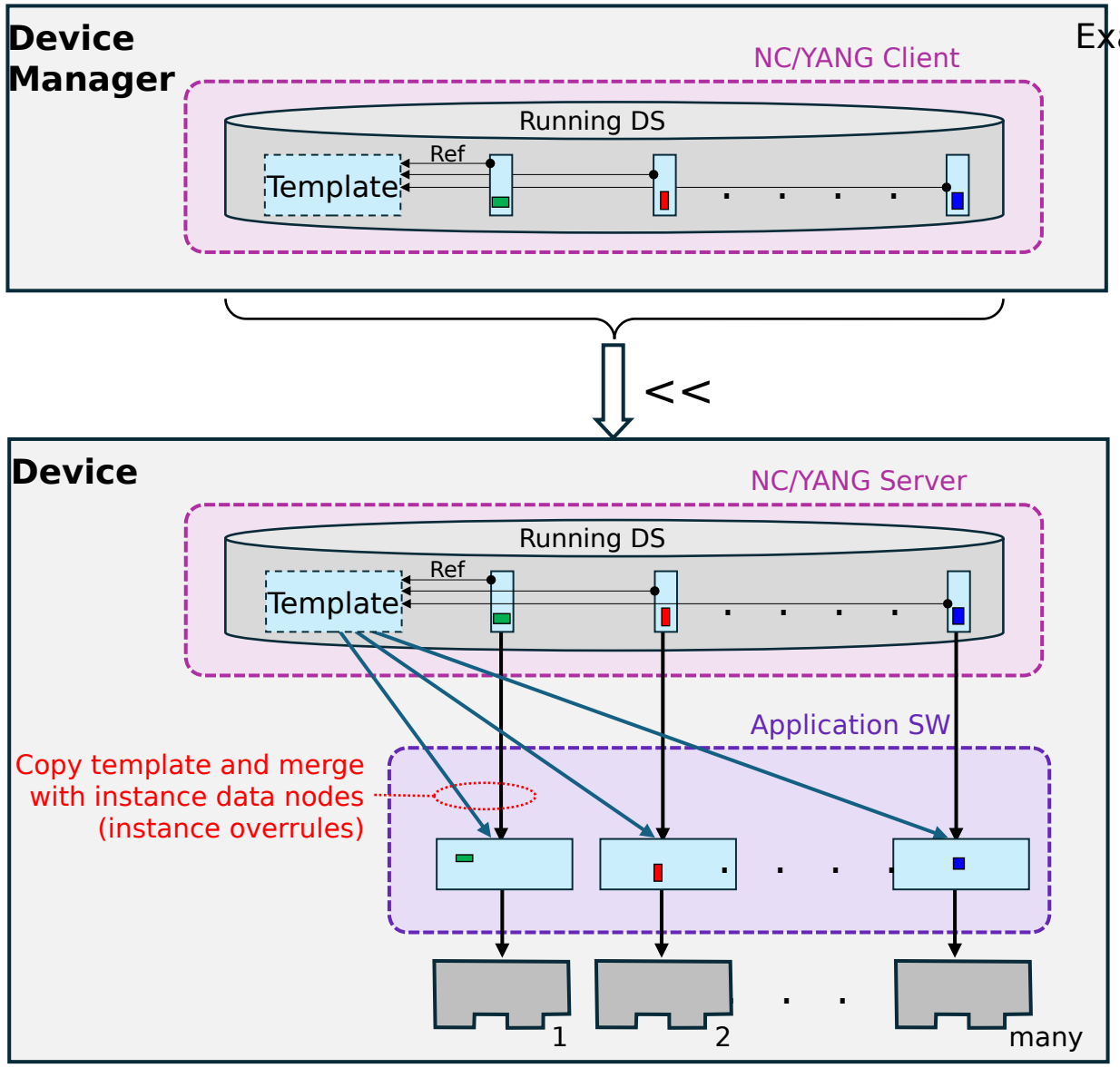
```
+--rw instance* [name]    // large list of functional instances
    +--rw name             string-ascii64
    +--rw description?     string-ascii128
    +--rw data
       +--rw list-a [name] //e.g. similar to a list of interf.
       |  +--rw name
       |  +--rw parm-x
       |  +--rw parm-y
       |
       +--rw list-b [name] //e.g. similar to a list of hw comp.
          +--rw name
          +--rw parm-t
          +--rw parm-u
```

The solution realization: how it works

The template technique consists in the following,

- configure a *model of a typical instance*, call it "*template*"
- configure the list of instances but without data nodes, except a reference to the configured template that they should be cloned from. This reference instructs the device software to populate the data nodes configuration of the instance with a copy from the template configuration
- Because an instance may need some variation from the template configuration, the technique still allows to configure any data nodes on a per instance basis with values that will overrule the values coming from the template (naturally the template method is most efficient when deviations from the template are limited).
- To allow any data node to be customized, the template YANG model and the instance YANG model contain the same data nodes.

Configuring the device by means of the template technique:



Example of YANG tree using the template technique:

```
+--rw template* [name]    // small list of templates
|  +--rw name             string-ascii64
|  +--rw description?     string-ascii128
|  +--rw data
|     +--rw list-a [name] //e.g. similar to a list of itfs.
|     |  +--rw name
|     |  +--rw parm-x
|     |  +--rw parm-y
|     |
|     +--rw list-b [name] //e.g. similar to a list of hw comp.
|        +--rw name
|        +--rw parm-t
|        +--rw parm-u
|
+--rw instance* [name]    // large list of functional instances
   +--rw name             string-ascii64
   +--rw description?     string-ascii128
   +--rw template?        -> /data-nodes-pattern/template/name
   +--rw data
      +--rw list-a [name] //e.g. similar to a list of itfs.
      |  +--rw name
      |  +--rw parm-x
      |  +--rw parm-y
      |
      +--rw list-b [name] //e.g. similar to a list of hw comp.
         +--rw name
         +--rw parm-t
         +--rw parm-u
```

Template Realization Using YANG Groupings

- To allow any data node to be customized or overridden, the template YANG model and the instance YANG model contain the same data nodes.
- An easy way to do so is when data nodes are defined in a grouping. The same grouping can then be used to define data nodes inside the template and the instance.
- When this method is used for existing standards, a copy of existing nodes is made into a grouping in a new module. which is then imported into applicable template model and the instance model to avoid duplication.
- This draft explores the possibility of templatizing the existing ietf-interfaces model by grouping sample nodes into groupings thereby allowing the possibility to use that in template as well as the instance.

Summary

A generic **template implementation** to configure YANG managed devices

- Suitable when device contains many instances of same function with limited variations in their config. values (a device contains a HW sub-system replicated many times)

Benefits:

- Improves scalability, reduces running Data Store -> less memory footprint & faster configuration
- Reusability & consistency: common structure configured only once and applied multiple times in the device
- Simplifies maintenance: if a template is modified, changes will be automatically reflected where it is used
- Relies on existing YANG 1.1 and off-the-shelf tools.

This draft aims at being informational to convey a possible method of implementation by different organizations using existing tools which cover basic functionalities of an overall template solution.

This of course does not cover a wider set of template requirements which was discussed in IETF 121/interims(https://github.com/netmod-wg/template-reqs/issues)