Opstate Solutions Comparison

Rob Wilton

IETF 94 – Yokohama, NETMOD WG

High level summary of Requirements

The core “opstate requirements” basically come down to:

- To recognize that some systems are configured asynchronously
- To determine what configuration a system is actually running. I.e. programmed everywhere it needs to be
- To be able to determine when some intended configuration change has actually been applied to a system

Proposed Solutions

Three solutions have been presented as drafts that each solve the core problem in different ways:

1. draft-openconfig-netmod-opstate

This solution is based around the **structure of the model**

2. draft-kwatsen-netmod-opstate

This solution is based around **multiple datastores**

5. draft-wilton-netmod-opstate-yang

This solution is based around **schema encoding**

High level comparison of Solutions

Solution	Main benefits
1. Model structure	No enhancement to NETCONF or RESTCONF needed. Does not require datastores.
2. Multiple datastores	No change required to YANG data models. Minimal NETCONF protocol impact.
3. Request encoding	No change required to YANG data models. Does not require datastores.