

# Report from the DataStore Design Team

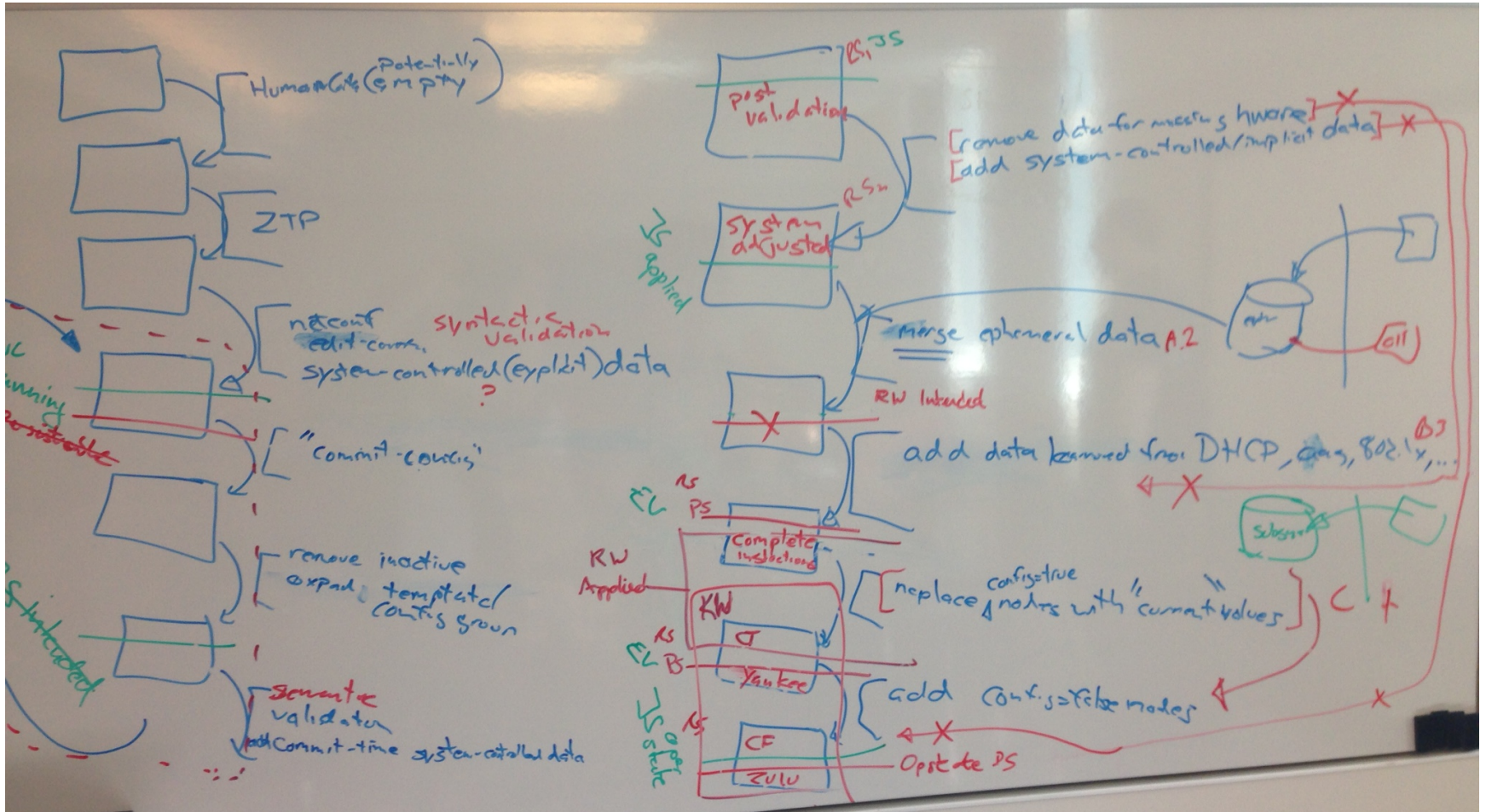
draft-nmdsdt-netmod-revised-datastores-00.xml

IETF97

# Tale of Two Drafts

- Two similar but different drafts:
  - draft-schoenw-netmod-revised-datastores
  - draft-wilton-netmod-refined-datastores
- Minor but important differences
- Discussions seemed to make little progress
  - Terminology was fuzzy
- Wednesday break out at the last IETF

# Discussion Outcome



# Design Team Task

The NETMOD Working Group has formed the Updated YANG Datastore Design Team. The mandate for the DT is to build on the drafts [1, 2] and discussions related to using a conceptual datastore-based approach to support <applied> vs <intended> configuration, and deliver a baseline individual draft for discussion by the Working Group prior to the next IETF (IETF 97, Seoul). The proposed solution should also take into consideration support for ephemeral state as documented by the I2RS Working Group. The published individual draft will be discussed and progressed per normal WG process.

# In other words....

- Small group discussion
- Outcome is an individual draft with proposed solution
- Not binding on the WG
- "Certainty of death, small chance of success? What are we waiting for?"
- And we actually got it done!

Thanks to the Design Team Members:

Martin Bjorklund

Rob Shakir

Rob Wilton

Juergen Schoenwaelder

# Our Solution

- Use three new, well-defined datastores:
  - <intended>
  - <applied>
  - <operational-state>
- All three are read-only
- Went past task to consider operational data
  - An issue we've left unsolved ([rfc6244#section-4.3](#))

# <intended> (#5.1)

- Contains the validated configuration
  - "post validation"
  - As seen by the rest of the system
- Any templates/scripts/etc expanded
- Any inactive nodes are removed
- Content driven from <running>
  - May be identical (implementation choice)
- Only "config true" nodes

# <applied> (#5.2)

- Currently active in-use configuration data
- Complete view of "config true" nodes
  - Where origin is static or dynamic (no defaults)
- Data may be removed:
  - Missing resources (aka ephemeral interfaces)
- Data may be added:
  - Non-"traditional" configuration sources:
    - DHCP, Dynamic Datastores, 802.1x, etc



# Origin Attribute

- "origin" attribute describes the source of data
  - Appears on each node
  - Value comes from YANG identity
    - static – data comes from <intended>
    - dynamic – data from dynamic datastore
    - data-model – value comes from data model
    - system – system-controlled data
  - Allows extensibility
    - dhcp based on dynamic, etc

# <operational-state> (#5.3)

- "The whole enchilada"
  - All nodes, "config true" and "config false"
- Currently active in-use values
- "config true" nodes are marked with the origin attribute
- Constraints from data models do not apply
- <applied> is subset of <operational-state>
  - Where @origin is "static" or "dynamic"

# Implications (#6)

- Define new DSs
- Device advertising support for DSs
  - NETCONF: capability exchange
  - RESTCONF: ??
- `<get/>` is deprecated  
(And there was much rejoicing!)  
Also `{+restconf}/data`
  - Needs parameter for `<operational-state>`
- Clarification
  - YANG constraints apply to `<intended>`

# Open Issues (#B)

- Add <active> datastore?
  - Between <running> and <intended>
- Semantic constraints in <operational-state>?
  - Are they just ignored?
  - Do we need a new YANG statement to define if a "must" constraints applies to the <operational-state>?
- Support for <applied> in RESTCONF?
- Better name for "static configuration"?
- Better name for "intended"?