Ephemeral State for I2RS
IETF 91 - Honolulu

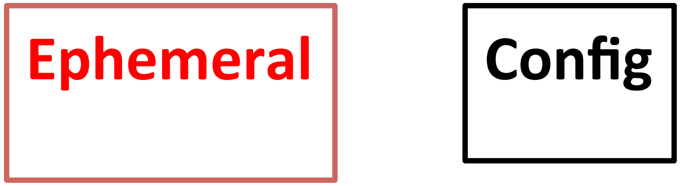
Jeffrey Haas – jhaas@pfrc.org

Ephemeral State

- The I2RS architecture draft (draft-ietf-i2rs-architecture) refers to I2RS state as ephemeral.
- A goal of “simplicity” is also stated in the same architecture draft.
- I2RS chose netconf/restconf and Yang as its mechanisms to interact with a routing element.
- Given the above, what does “ephemeral state” *mean?*

Use cases for ephemeral state: Disjoint

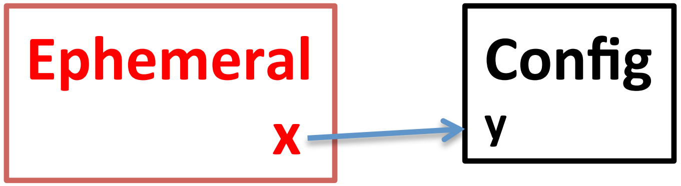
- Ephemeral state and configuration state do not interact with each other; common protocol operations may retrieve either.



Example use case: A topology data model that doesn’t use state from other IGP data models.

Use cases for ephemeral state: Ephemeral refers to config

- Yang constraints such as "must/when" refer from ephemeral state to config state.
- The reverse is probably never safe.



Ephemeral

x

Config

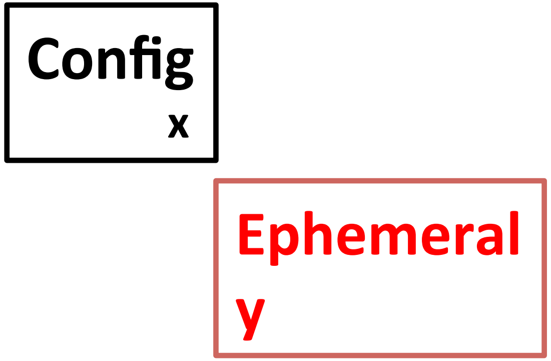
y

leaf x has a must relationship on y in the config. State is otherwise disjoint.

Example use case: A dynamically created BGP neighbor in the Ephemeral datastore uses the Config datastore's Autonomous System value.

Use cases for ephemeral state: Augmenting

- Rather than “copy and paste” some bit of related config into an i2rs schema, i2rs provides an augmentation on configuration state to provide the i2rs related feature:



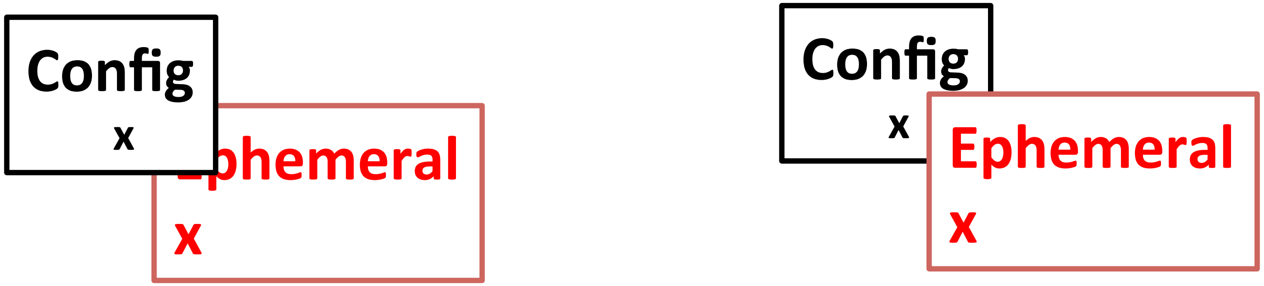
Config
x

Ephemeral
y

Ephemeral node y is a child of x. However, what happens if x is deleted? What about consistency between the two separate datastores?

Example use case: An IGP interface has I2RS state adding a “color” Traffic Engineering Attribute

Use cases for ephemeral state: Overriding/occluding



x could overlap in different datastores in the same place in the schema. Depending on "priority", a read operation on x may return either the config datastore copy or the ephemeral datastore copy.

Example use case: A static route in the Config's datastore could have its nexthop overridden by dynamic state.

NYC netmod interim, September 2014

- Based on list discussion in I2RS, draft-haas-i2rs-netmod-netconf-requirements was assembled to try to capture some possibilities for ephemeral state.
- Originally proposed 3 possibilities were:
  1. Separate ephemeral datastore.
  2. Configuration state in running datastore "tagged" as ephemeral.
  3. Permitting existing configuration to be configured as ephemeral.

Option 4

- During discussion of the various pros and cons of the three proposed discussion points, a fourth option reached some level of room consensus as a useful possibility:
  - Any config for the supported schema in the system could be written to the ephemeral datastore.
  - For nodes present in the config datastore and not occluded by ephemeral state, it is possible to read the merged/union form of config+ephemeral based on ephemeral configuration priority.
  - Support would be required for this in netconf.

Open issues

- Discussion on the I2RS list after the interim did not produce consensus about what we should do about the ephemeral state. This will be the primary topic for Thursday's I2RS session.
- If we do choose something like option 4, our belief is that these changes are now primarily netconf.
- We do have other things that require resolution, like changes for targeting notifications to a session different than the connected one, but will save this for until the harder problem is addressed.

Interesting Observation

- The peer-mount drafts (draft-clemm-netmod-mount, et al.), while not I2RS focused, have a number of similarities of the *problems* they'd have for cross-repository consistency.
- However, this problem space is something people innately understand better due to its similarity to a file system.
- Solve problem for peer-mount, some of the issues are addressed for i2rs.