I2rs Requirements for NETCONF

Status for

I2RS Requirement on WG LC

- draft-ietf-i2rs-ephemeral-state-00
- draft-ietf-i2rs-pub-sub-requirements/
- draft-ietf-i2rs-traceability/
- draft-ietf-i2rs-protocol-security-requirements-01

Adopted environmental security requirements

- draft-mglt-i2rs-security-environment-reqs-01

Summary

- Ephemeral state from the protocol strawman
- Yang changes
- Identity requirements
- Priority requirements
- Pub/sub requirements
- Security requirements

Ephemeral State

Ephemeral State

- Requirement: Ephemeral state does not persist across reboots
- Comment:
  - Ephemeral state is not a i2rs specific requirement.

Ephemeral State

1. The ephemeral datastore is a datastore holds configuration that is intended to not survive a reboot.

2. The ephemeral datastore is never locked.

3. The ephemeral datastore treated as N client panes where:

- the netconf/restconf server picks how many clients it supports
- multi-head support is optional since max-clients allowed to be

Ephemeral State

4. Each client has a unique priority (see figure 3 for example yang statements)

- If a client is not present in the i2rs-client list, then the worst priority value is assigned.
- The best possible priority needs to be reserved for the system, or the protocol has to make a special case of systems-set data.

5. Each client writes into its own pane so there is no conflict within a pane. The implementation combines the panes into the appropriate image.

- The difference between panes of glass is what the server retains from a partial or failed edit (due to conflicts in the panes (?editor)). It should be a valid operation to save nothing
- It should be a valid operation to save nothing or to save all information (caching) within a pane of glass

Ephemeral (continued)

6. A Partial operation is one where a subset of the written data is not applied because of better priority for that node. A partial operation is only allowed if the error-option is stop-on-error or continue-on-error.

- stop-on-error - means that the configuration process stops when a write to the configuration detects an error due to write conflict.
- continue-on-error - means the configuration process continues when a write to the configuration detects an error due to write process, and error reports are transmitted back to the client writing the error.
- all-or-nothing - means that all of the configuration process is correctly applied or no configuration process is applied.
- Interoperability issues will need to be consider for these three cases

- NETCONF stop-on-error and continue-on-error are not going to work. There is no mandated processing order for edits.
  - For the stop-on-error and the continue-on-error process to work, the I2RS protocol extensions to NETCONF will have to force some processing order in order to support partial edits.
  - NETCONF has no current mechanism for reporting which edits were accepted and which edits were reject for partial operations. The I2RS protocol extensions will have to provide new error handling to the response data.

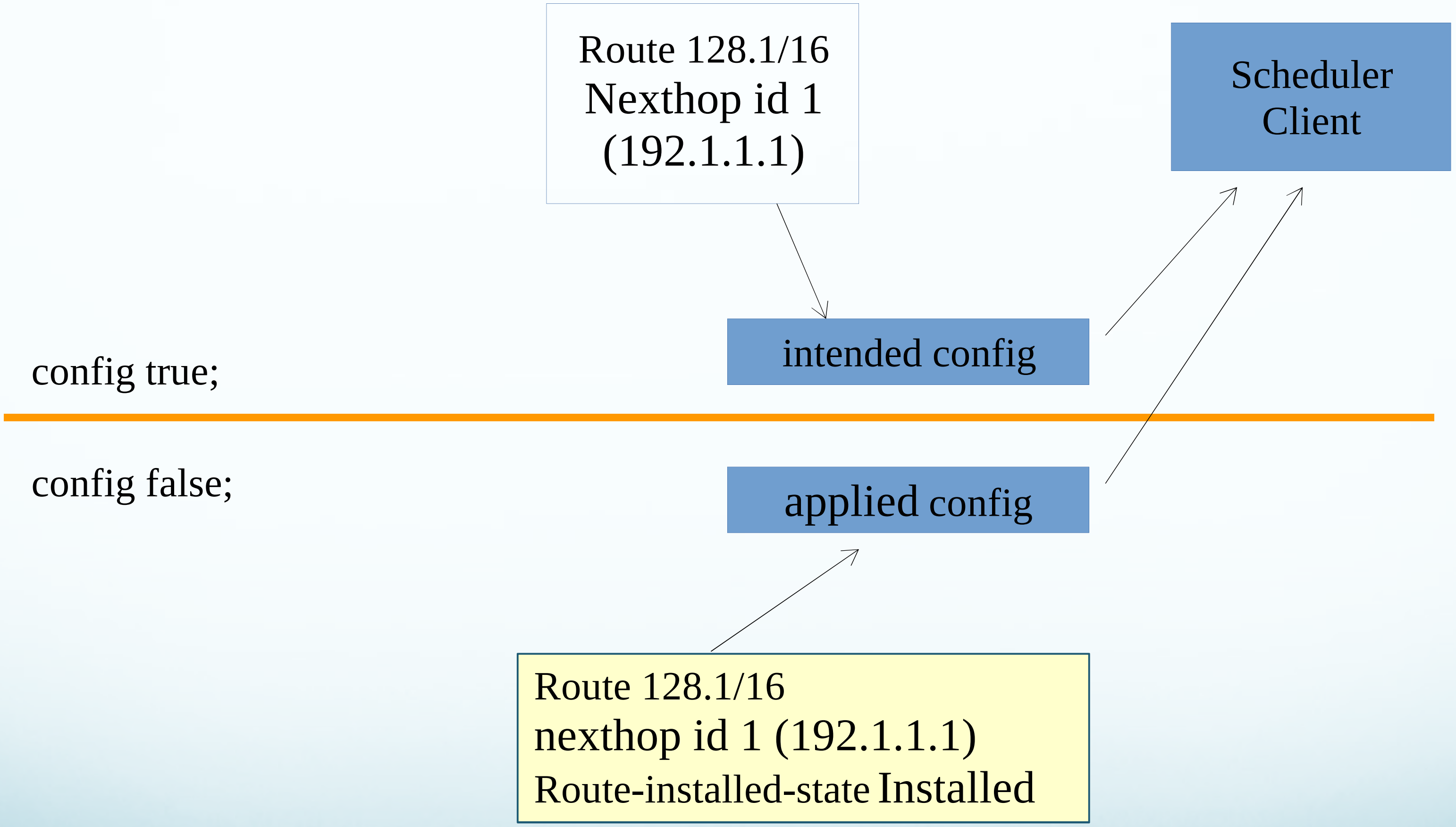
Ephemeral State

- 7. caching is optional and and a server may retain the pain for each client.
  - If caching is not supported then the pane-of-glass never contains unaccepted data. Therefore, the server will return an error and will not retain the edit that caused the error.
  - If caching is supported, then the data is retained in the pane-of-glass, Therefore, if the higher priority data is removed then the lower priority data can be added.
  - Notifications will be provided for the caching process

```
container i2rs-clients {
   leaf max-clients {
     config false;
     mandatory true;
      type uint32 {
        range "1 .. max";
      }
  }

  list i2rs-client {
      key name;
      unique priority;
      leaf name { ... }
     leaf priority { ... }
  }
}
```

Thermostat Model RIB Equivalent



Route 128.1/16
Nexthop id 1
(192.1.1.1)

Scheduler
Client

config true;

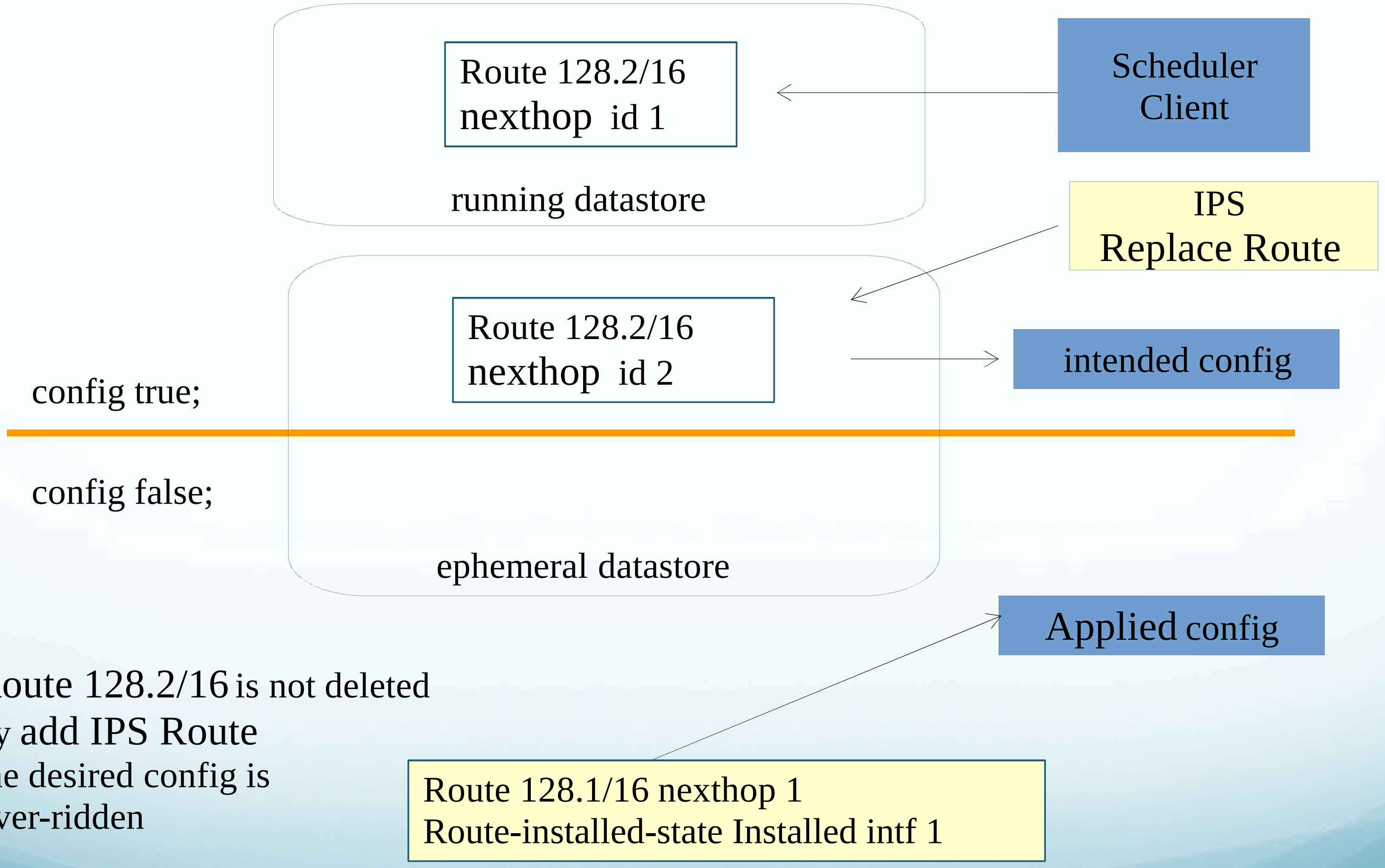
intended config

config false;

applied config

Route 128.1/16
nexthop id 1 (192.1.1.1)
Route-installed-state Installed

Route + Ephemeral Route



Route 128.2/16
nexthop id 1

running datastore

Scheduler
Client

IPS
Replace Route

Route 128.2/16
nexthop id 2

intended config

config true;

config false;

ephemeral datastore

Applied config

Route 128.2/16 is not deleted
by add IPS Route
the desired config is
over-ridden

Route 128.1/16 nexthop 1
Route-installed-state Installed intf 1

YANG changes

- config ephemeral
- Assumes a certain solution

Simple Thermostat + ephemeral

```
module thermostat {
    …
    leaf desired-temp {
       type int32;
       units “degrees Celsius”;
       description “The desired temperature”;
    }

    leaf actual-temp {
       type int32;
       config false;
       ephemeral true;
       units “degrees Celsius”;
       description “The measured temperature”;
    }
}
```

Need to identify
datat in PUT, PATCH or
RPC.
How does this occur?
ephemeral true; ??

Need to identify this
leaf as OK to edit in
the ephemeral datastore

RESTCONF Example

RESTCONF Running Datastore Edit

```
PUT /restconf/data/thermostat:desired-temp

{ “desired-temp”: 18 }
```

RESTCONF Ephemeral Datastore Edit of config=true

```
PUT /restconf/data/thermostat:desired-temp?datastore=ephemeral

{ “desired-temp”: 18 }
```

RESTCONF Ephemeral Datastore Edit of config=false

```
PUT /restconf/data/thermostat:actual-temp?datastore=ephemeral

{ “actual-temp”: 72 }
```

Identity Requirements

Identity Requirement

- Clients shall have identities
- Clients can have secondary identities
  - Carried as part of ??? (RPC, Meta-data)
  - Goes in as part of the notification/trace log

Priority Requirements

Priority Requirements

- Support multi-headed control –
  - Different than ephemeral
  - Supported as part of write-collision
- I2rs attributes may be modeled as meta-data

Pub/Sub Requirements

Pub/Sub

- The I2RS interface should support user subscriptions to data with the following parameters: push of data synchronously or asynchronously via registered subscriptions…
  - Comment: Is security hole concern from NETCONF addressed in the current document?
- The I2RS interface (protocol and IMs) should allow a subscribe to select portions of the data model.
- Real time notification of events e.g. route installation and removal
  - 1-5 seconds

Pub/Sub (cont.)

Requirements related to protocols

- The I2RS agent should be able to notify the client via publish or subscribe mechanism a BGP route change on a specific IP
  - Response: Notification/Subscription per Model and per item
- Can subscribe to the I2RS Agent’s notification of critical node IGP events.
  - Response: Data model defines critical nature
- I2rs must be able to collect large data set from the network with high frequency and resolution with minimal impact to the device's CPU and memory
  - Response (from Alia): 2000/second

Pub/sub and tracing

- I2RS Agents should support publishing I2RS trace log information to that feed

Security Requirements

I2rs Security Requirements

- Requirements 1, 2, 5, 6, 7, 9, 11, 13, 14, 15, 16, 18, 19, 20
- Security review: 8,12, multiple messages, insecure protocol
- Editorial
  - Editorial: Req. 3 /4 – rewritten
  - Req 10 – rewritten
- Security review
  - Req. 8 – is a a security requirement
  - Req 12 –is a protocol security (DDoS a protocol functions)
  - Multiple messages – removed
  - Insecure protocol
    - Only if Data model clearly states it.
    - Call for RIB to determine if can state it.

Action Items

- (will be edited during conference)