Kick starting
Capabilities for I2RS

Susan Hares (Huawei)

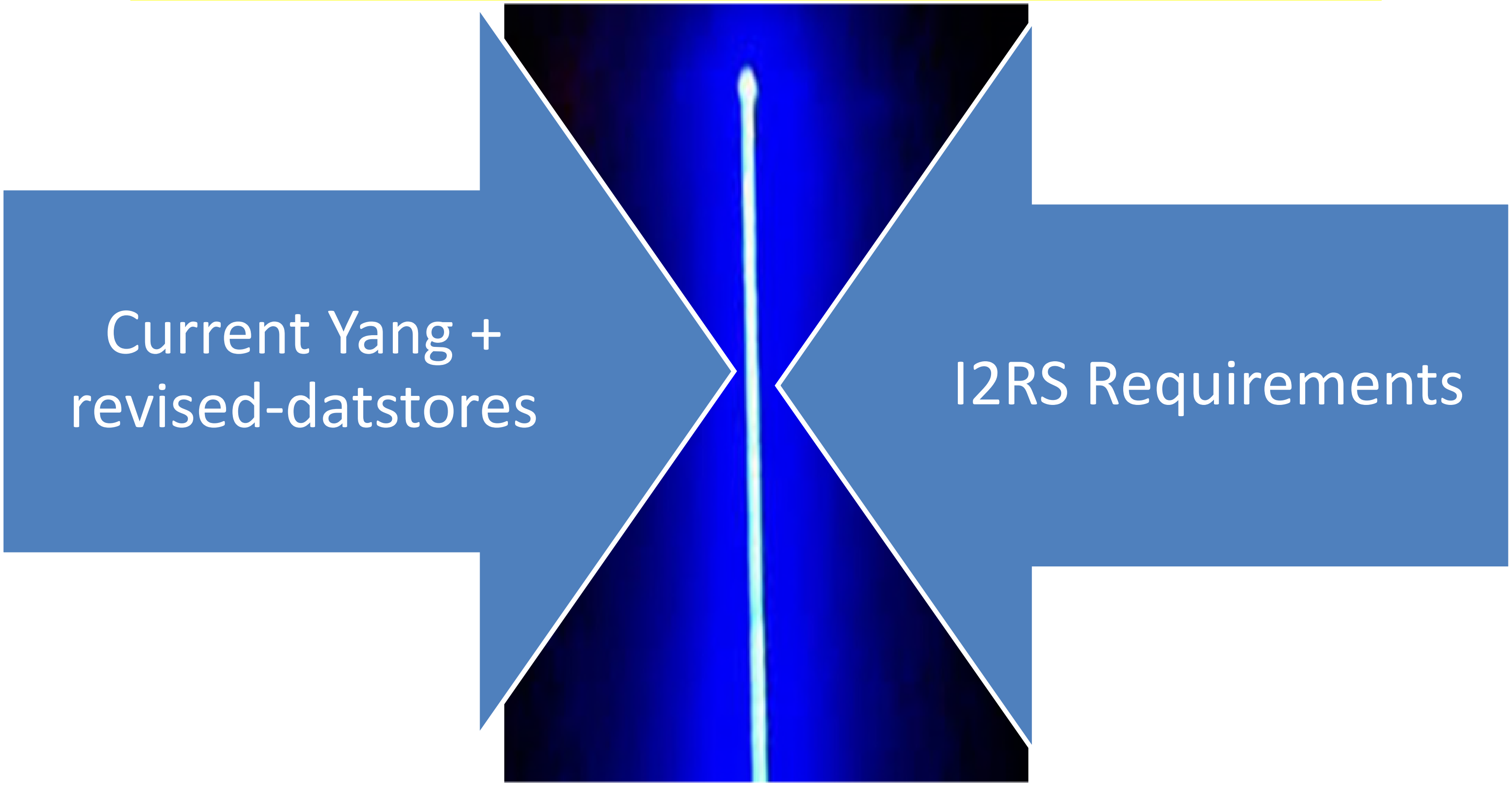
draft-hares-netconf-i2rs-netconf-01

draft-hares-restconf-i2rs-restconf-02

draft-hares-netconf-i2rs-yang-04

Yang + Protocol Discussion

Kick-start from 2 directions



Current Yang + revised-datstores

I2RS Requirements

Kick Start Ideas for I2RS Protocol i2rs-control-plane datastore

RESTCONF (RFC8040+)

- **Support revised-datastores**

  {+restconf}/datastore/ <datastore-name>/data/

- **Write Collision:**

  I2RS client/priority -→ Entity-Tag

- **Ephemeral Capability**
- **Validation** - Internal mechanisms or specified?

Kick Start Ideas for I2RS Protocol i2rs-control-plane datastore

NETCONF:

- **New operations**: **<read-data>, <write-data> <resource-limits>** **(??)**
- **Modified operations:** **<kill-session>, <close-session>**
  - Must be able to shutdown session just for datatstore.
  - Assumes Bundling of revised datastores netconf
- **Modified Capabilities:** rollback-on-error, validate
- **New capabilities:** secondary id, Ephemeral
- **TBD**: edit-write collision

Kick Start Ideas for I2RS Protocol i2rs-control-plane datastore

NETCONF:

- **Unsupported options:** <get-config>, <edit-config>, <copy-config>, <delete-config>, <lock>, <unlock>
- **Unsupported capabilities:** writable-running, candidate-configuration, confirmed commit, distinct start-up
- **Capabilities:** rollback, secondary id, Ephemeral
- **TBD:** edit-write collision

Ephemeral Capability

- Not just for I2RS control-plane
- **ephemeral**
  - Values do not persist on reboot
  - Ephemeral data does not rollback (all or nothing)
  - Filter on meta-data “ephemeral”

Revised Datastore Additions

1. Name : ephemeral

2. YANG modules : all (default)

3. YANG statements :

config false + ephemeral true

4. How applied : automatic

5. Protocols : NC/RC (default)

6. YANG Module : (see below)

1. Name: i2rs-control-plane

2. Yang Modules: default none

3. Yang Module Statements:
config false + ephemeral true

4. How applied: precedence

5. Protocols: RC/NC/…

features:

6. YANG: see below

?? validation

Yang Proposals

Proposal:

```
datastoredef <dsname> {
      substatements
};
```

Yang statement

Alternative in revised-datastores

```
 import ietf-datastores { prefix ds; }
import ietf-origin { prefix or; }
identity ds-ephemeral
  { base ds:datastore;
  description "The 'ephemeral' datastore.";
}
extension ephemeral {
 argument "value";
 description "…"
}
```

Set of Imports

	New	RFC7960	cardinality
Description	No	7.21.3	0…1
dstype	Yes (3.3)	….	1
Ephemeral	Yes (3.4)	….	0…n
Module	No	7.1	0..n
Module-list	Yes (3.5)	….	0..n
Organization	No	7.1.7	0..1
Precedence	Yes(3.6)	…	0..n
Protosup	Yes (3.7)	…	0..n
Reference	No	…	0…1
Revision	No	…	0…n
Revision-date	No	…	0..n
Validation	Yes (3.8)	….	0..n
Version	No	…	0..n

Let’s look at differences

dstype <dsclass> <dsname>;

Class might be different

Protosup <protobase> <protosupport>

protobase: NC/RC (default) /CoAP/gprc …

protosupport: features

Need DS catalog additions

precedence [applied | opstate | <dsname>] { <<precedence-substatements>> };

I2RS needs DS ordering

validation { <validation-flags-statements>
<validation-policy> };

Do we need?

Handling revisions

- Features, Release, Module lists -
  - draft-openconfig-netmod-model-catalog
  - Probably needed as I2RS++ style modules will have lots of revisions
- Do we need something like this for Datastores?

What About Validation

```
validation { <validation-flags-statements>
             <validation-policy> };
```

	New	RFC7960	cardinality
Description	No	7.21.3	0…1
bulkchecks	Yes (3.8.2)	….	1
caching	Yes (3.8.3)	….	0…n
nstransport	Yes (3.8.4)	7.1	0..n