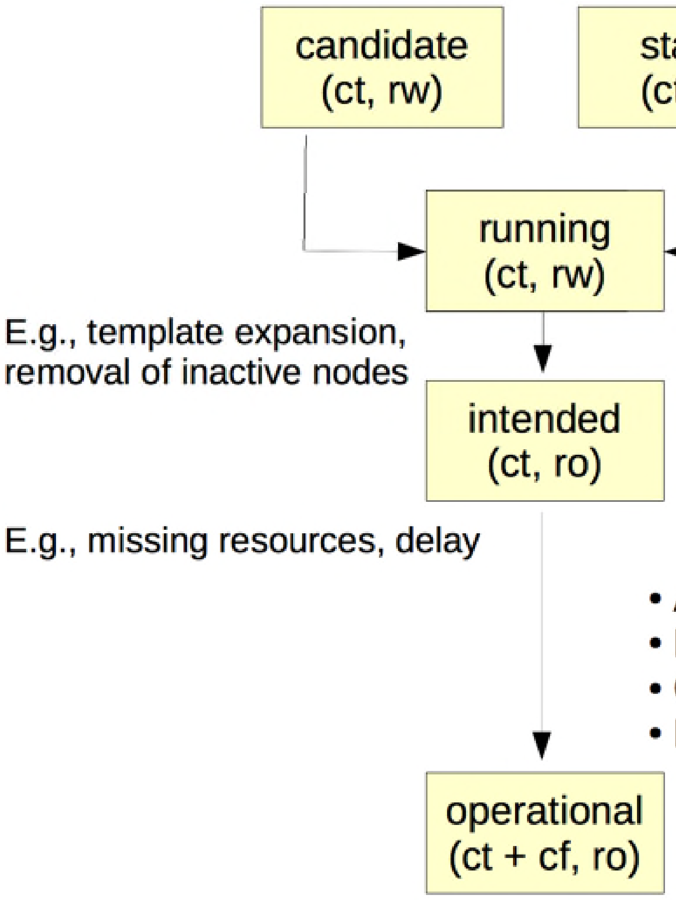
Revised Datastores and I2RS

draft-ietf-netmod-revised-datastores-01

I2RS WG

IETF 98 (Chicago)

Datastore Architecture



candidate
(ct, rw)

startup
(ct, rw)

running
(ct, rw)

E.g., template expansion,
removal of inactive nodes

intended
(ct, ro)

E.g., missing resources, delay

operational
(ct + cf, ro)

- Auto-discovery
- Dynamic config protocols
- Control-plane protocols
- Dynamic datastores (e.g., I2RS)

This is an extensible model; new datastores can be defined, and new mechanisms (e.g., inactive) can be defined by referring to this model.

Guidelines for Defining Datastores

- In a document (e.g., RFC)
  - Define a name for the dynamic datastore
  - Define which YANG modules can be used in the datastore
  - Define which subset of YANG-modeled data applies
  - Define how dynamic data is actualized
  - Define which protocols can be used
  - Define a module for the dynamic datastore

Example

1. Name : ephemeral
2. YANG modules : All (default)
3. YANG statements : config false + ephemeral true
4. How applied : Automatic
5. Protocols : RESTCONF (NETCONF optional)
6. YANG Module : (next slide)

```
module example-ds-ephemeral {
     yang-version 1.1;
     namespace "urn:example:ds-ephemeral";
     prefix eph;
     import ietf-datastores { prefix ds; }
     import ietf-origin { prefix or; }

     // add datastore identity
     identity ds-ephemeral {
       base ds:datastore;
       description "The 'ephemeral' datastore.";
      }

     // add origin identity
     identity or-ephemeral {
       base or:dynamic;
       description "Denotes data from the ephemeral dynamic datastore.";
      }

     // define ephemeral extension
     extension ephemeral {
       argument "value";
       description
         "This extension is mixed into config false YANG nodes to indicate
          that they are writable nodes in the 'ephemeral’ datastore.  This
          statement takes a single argument representing a boolean having
          the values 'true' and 'false'.  The default value is 'false'.";
      }
}
```

Test Module

```
module example-my-module {
    yang-version 1.1;
    namespace "urn:example:ds-ephemeral";
    import example-ds-ephemeral { prefix ex-eph; }  // from prev slide

    container my-config {

      // editable in running and ”ephemeral”
      leaf normal-config {
        type string;
      }

      // editable in “ephemeral” only
      leaf ephemeral-config {
        type string;
        config false;
        ex-eph:ephemeral true;
      }

    }
}
```

Next Steps

- I2RS SHOULD develop an RFC defining the datastore(s) needed, following the guidelines mentioned above.

- The example shown here is very terse and somewhat incomplete. Expectation is that such an RFC would span many pages.

- Thoughts:

  1. Define a stack of dynamic datastores?
     - E.g., one for each user or priority
       - i2rs-p1, i2rs-p2, …, i2rs-pn

  2. Define other I2RS-specific extension statements?
     - E.g. metadata for secondary identity?
     - E.g., a cross-datastore leafref?