draft-iet-i2rs-ephemeral-state-reqs-00

Jeffrey Haas and Susan Hares

(jhaas@pfrc.org and shares@ndzh.com)

Intent of this document

- To attempt to provide concrete examples of desired I2RS protocol behavior.
- To drive discussion about potential implementations of that behavior and their representations in netconf/restconf and yang.

Changes

- Added 10 requirements from list
- Priority on NACM group versus NACM rule list
- Addition discussion on the semantics around storing and managing priority and client id,
- Discussion on transactions

Top 10 requirements (1)

1. The I2RS protocol SHOULD support highly reliable notifications (but not perfectly reliable notifications) from an I2RS agent to an I2RS client.
2. The I2RS protocol SHOULD support a high bandwidth, asynchronous interface, with real-time guarantees on getting data from an I2RS agent by an I2RS client.

3. The I2RS protocol will operate on data models which may be protocol independent or protocol dependent.

Top 10 Requirement (2)

4. I2RS Agent needs to record the client identity when a node is created or modified. The I2RS Agent needs to be able to read the client identity of a node and use the client identity's associated priority to resolve conflicts. The secondary identity is useful for traceability and may also be recorded.

5. Client identity will have only one priority for the client identity. A collision on writes is considered an error, but priority is utilized to compare requests from two different clients in order to modify an existing node entry. Only an entry from a client which is higher priority can modify an existing entry (First entry wins). Priority only has meaning at the time of use.

Top 10 Requirements (3)

6. The Agent identity and the Client identity should be passed outside of the I2RS protocol in a authentication and authorization protocol (AAA). Client priority may be passed in the AAA protocol. The values of identities are originally set by operators, and not standardized.

7. An I2RS Client and I2RS Agent mutually authenticate each other based on pre-established authenticated identities.

8. Secondary identity data is read-only meta-data that is recorded by the I2RS agent associated with a data model's node is written, updated or deleted. Just like the primary identity, the secondary identity is only recorded when the data node is written or updated or deleted

Top 10 Requirements (4)

9. I2RS agent can have a lower priority I2RS client attempting to modify a higher priority client's entry in a data model. The filtering out of lower priority clients attempting to write or modify a higher priority client's entry in a data model SHOULD be effectively handled and not put an undue strain on the I2RS agent.

10. The I2RS protocol MUST support the use of a secure transport. However, certain functions such as notifications MAY use a non-secure transport. Each model or service (notification, logging) must define within the model or service the valid uses of a non-secure transport.

Changes to placement of I2rs priority in NACM

Old
draft-haas-i2rs-ephemeral state-00

```
+--rw rule-list [name]
    +--rw name     string
    +--rw group*   union
    +--rw rule [name]
      +--rw name string
      +--rw module-name?  union
      +--rw (rule-type)?
      |  +--:(protocol-operation)
      |  |  +--rw rpc-name?  union
      |  +--:(notification)
      |  |  +--rw notification-name?  union
      |  +--:(data-node)
      |     +--rw path node-instance-identifier
      +--rw access-operations?  union
      +--rw action action-type
      +--rw comment?  string
      +--rw i2rs:i2rs-priority i2rs-priority-type
```

New
Draft-ietf-i2rs-ephemeral-state-00

```
+--rw nacm
  +--rw enable-nacm?           boolean
  +--rw read-default?          action-type
  +--rw write-default?         action-type
  +--rw exec-default?          action-type
  +--rw enable-external-groups? boolean
  +--ro denied-operations
        yang:zero-based-counter32
  +--ro denied-data-writes
         yang:zero-based-counter32
  +--ro denied-notifications
         yang:zero-based-counter32
  +--rw groups
  |  +--rw group [name]
  |     +--rw name        group-name-type
  |     +--rw user-name*   user-name-type
  |     +--rw i2rs:i2rs-priority i2rs-priority-type
```

No Multi-message atomicity & rollback Multiple Message Handling

I2RS architecture does not include multi-message atomicity and rollback mechanisms, but suggests an I2RS client may inidicate one of the following error handling techniques for a given message sent to the I2RS client:

1. Perform all or none: All operations succeed or none of them will be applied. This useful when there are mutual dependencies.

2. Perform until error: Operations are applied in order, and when error occurs the processing stops. This is useful when dependencies exist between multiple-message operations, and order is important.

3. Perform all storing errors: Perform all actions storing error indications for errors. This method can be used when there are no dependencies between operations, and the client wants to sort it out.

Previous discussed Requirements

5/27, 6/10 I2RS interims

Flagging configuration state as ephemeral

- Proposal: Extend “config” yang keyword to include “ephemeral”.
- Initial discussion: Consider instead a separate keyword “ephemeral true”.
- (Martin Bjorkland also points out we’re potentially hitting much of what is in draft-bjorklund-netmod-operational-00.)

Hierarchy

- Ephemeral configuration may be a child of persistent configuration. The reverse is not permitted.
- Operational state whose parent is ephemeral MUST also be ephemeral.

Netconf Changes

- Announce an ephemeral-config capability.
- Add a new parameter, “filter-ephemeral” to <get-config> and <get>.
  - Consider alternative from draft-bjorklund-netmod-operational-00. Martin suggests we shouldn’t overload <get-config>.

Secondary identity

- A property of I2RS ephemeral state that is stored for each ephemeral configuration state node.
- Made accessible to the user as a read-only piece of meta-data. Note that “read-only” meta-data would be a new construct.
- Carried as a parameter to <edit-config>

Priority

- Similar to secondary-identity, a property of each ephemeral configuration state node.
- User's priority is assigned as a new attribute of NACM.
- For new ephemeral nodes, it is assigned the user's priority for that node. (NACM may vary it by path.)
- For existing ephemeral nodes, the update is only permitted if the user's priority is > the existing node's priority (First holds) The node then has this priority.
- Transaction/Commit will fail if the user has insufficient priority.
- Presented to the user as read-only meta-data.