

# I2RS Architecture

Alia Atlas, Joel Halpern, Sue Hares, David Ward, and Tom Nadeau

IETF 88, Vancouver, B.C.

# Issue 1: Indirect Interactions

“In addition it must be noted that there may be indirect interactions between write operations. Detection and avoidance of such interactions is outside the scope of the I2RS work and is left to agent design and implementation for now.”

- No responses to discussion on list.
- So... what's there is good (enough)

## Issue 2: Persistence (ephemeral vs. permanent)

“The I2RS Agent will not attempt to retain or reapply state across routing element reboot. Determination of whether state still applies depends heavily on the causes of reboots, and reapplication is at least as likely to cause problems as it is to provide for correct operation.”

- LOTS of discussion – but mostly about I2RS Agent failure scenarios (next slide)
- On this – agreement for just ephemeral
  - If I2RS-installed state causes a problem being able to reboot to just the local config is good

# Side-Topic: I2RS Agent Failure

- What happens if an I2RS Agent fails independently of the associated routing element?
  - Lively discussion, branching into thoughts on pub/sub brokers, heartbeats (or lack of), etc.
  - Apparent consensus was:
    - Graceful failure (shutdown/disabled): I2RS Agent can (implementation-dependent) optionally notify all its clients that their state is being torn down. Then the I2RS Agent must notify all its clients that the agent is going down.
    - Agent crash (unexpected failure): Cache each known I2RS Client. When an I2RS Agent starts, it notifies each saved I2RS Client that it is up AND includes an agent-boot-count, indicating how many times the I2RS Agent has restarted since the associated routing element.
  - Pointing towards need for an I2RS Agent info-model and associated notifications.
  - Capture this concern and consensus in next version.

# Issue 3: Operations are Immediate & Continuing

“Given that the complexity of possible conditions is very large, and that some conditions may even cross network element boundaries, clearly some degree of handling must be provided on the I2RS client. As such, in this architecture it is assumed that all the complexity associated with this should be left to the I2RS client. This architectural view does mean that reliability of the communication path between the I2RS client and I2RS agent is critical.”

- Some support on list and no disagreements

# Issue 4: Templates for QoS & Policy

" Many network elements have separate policy and QoS mechanisms, including knobs which affect local path computation and queue control capabilities. These capabilities vary widely across implementations, and I2RS cannot model the full range of information collection or manipulation of these attributes. A core set does need to be included in the I2RS data models and in the expected interfaces between the I2RS Agent and the network element, in order to provide basic capabilities and the hooks for future extensibility."

- Interest in templates for QoS coming from RIB info model (Sri)
- Common mechanism desired across I2RS for common reuse.
- Named object (template) could have a type, name, and even a data-model associated with it.
- If consensus, will add a sentence on the named object to the architecture.

# Issue 5: Client Redundancy

- Lively discussion at last IETF – added text:

“I2RS must support client redundancy. At the simplest, this can be handled by having a primary and a backup network application that both use the same client identity and can successfully authenticate as such. Since I2RS does not require a continuous transport connection and supports multiple transport sessions, this can provide some basic redundancy. However, it does not address concerns for troubleshooting and accountability about knowing which network application is actually active. At a minimum, basic transport information about each connection and time can be logged with the identity. Further discussion is necessary to determine whether additional client identification information is necessary.”
- List Discussion:
  - Considering redundancy is good.
  - Leave network application or I2RS client redundancy outside of I2RS
- Consensus: What’s there is good enough...

# Planned Changes

- Discuss handling of I2RS Agent shutdown and failure.
- Add minor details on templates
  - **Any others suggested?**
  - **What's missing?**
  - **After this, ready for attempting WGLC?**