SIP PUBLISH

draft-ietf-simple-publish-01

Aki Niemi

aki.niemi@nokia.com

Changes since last version

- Editorship changed
- Merged with “draft-olson-simple-publish-02”
- Versioning based on HTTP validation model
  - Version information (entity-tag)
  - Request preconditions
- Other changes aimed at improved readability
  - Added text
  - Changed title
  - Editorial work

Open Issues: Atomicity of Publication

- Presence state is special (and problematic)
  - Can be split into segments (tuples)
  - Also contains other components (presentity-level elements)
  - Only a single container (PIDF)
    - Can have 0…n tuples, and/or presentity-level elements
- Option 1: each PUBLISH carries PIDF with full state for a particular PUA
  - Does a failed refresh invalidate all of it?
  - Each update (not a refresh) needs to carry full state
- Option 2: each PUBLISH carries PIDF with exactly one segment
  - Each atomic segment (<tuple>, <note>) has its own container
    - Publishing a <note> means PIDF without any tuples
    - Publishing a <tuple> means PIDF without presentity-level elements (or ignored if present)
  - Updating many tuples expensive (pipelining is not allowed)
- Option 3: combination of #1 and #2
  - Does each component in full state publication inherit version, expiration?

Open Issues: Identification of tuples

- In presence publishing, tuple-IDs identify the specific event segments
  - Chosen by the publishers
  - Persistent
- Need a way to generate these IDs in an organized manner
  - Two PUAs must be able to independently publish the same tuple
  - A PUA must be able to override another PUAs publication
  - Seems to suggest a predefined, finite set of tuple “types” with possibility for extensibility
- Idea 1:
  - Specific naming convention for tuple-IDs
- Idea 2:
  - Using namespaces and/or IANA registry for tuple-IDs

Open Issues: Collision recovery

- Two agents publish the same event state or event state segment
  - How to recover if a collision occurs?
  - How to avoid the case of battling automata?
- Current proposal:
  - Query principal for further action
  - MAY subscribe to the event package for current composite state
- Do we need more on this?

Open Issues: Requirement #14

- Requirement 14 reads:
  "PUAs MUST have a capability that allows them to query for the identifiers of all of the segments of presence information that have currently been published for a presentity (provided that the PUA is authorized to receive this information)"
- Currently not possible
  - Since a few slides back we solved the identification problem already…
  - What was the use-case behind this requirement?
- Proposal 1:
  - Abandon requirement
- Proposal 2:
  - 200 OK to PUBLISH contains "raw" aggregated presence document with all published tuple-IDs
- Proposal 3:
  - SUBSCRIBE to that event package gives you the data
  - Don't necessarily reveal the tuple-Ids though
- Proposal 4:
  - New event package for "raw" data

Open Issues: Requirement #19

- Requirement 19 says:

  “There must be a way for a publisher to tell a presence agent that a piece of published presence should be passed on to watchers without modification”

- Currently a signed tuple in a publication implies this
- Proposal 1:
  - Keep as it is
- Proposal 2:
  - Define an explicit mechanism, use Require header

Open Issues: Hard State Publishing

- Publishing “default” presence or hard state is out-of-scope for PUBLISH
  - Handled by an XCAP-usage
- Current draft shortly mentions other sources for event state
  - Properties of such sources are not discussed
  - No detailed description of how e.g., hard state is composed in
- Should the draft talk more about other sources of event state?
- Proposal:
  - No, leave this to other documents, e.g., XCAP-usage draft

Open Issues: Refresh after version expired

- Same error response in all cases
  - Versioning precondition fails because state has expired
  - ESC has rebooted and lost versioning information
- Will result in EPA “querying principal”
  - Simply re-publishing without the versioning precondition would suffice
- Proposal 1:
  - In case no version information whatsoever present at ESC, ignore the precondition
- Proposal 2:
  - New response code for “Precondition not applicable”

Open Issues: Relationship to Dialogs

- In current example, subscription precedes publications
  - Do not share the dialog – it's simply a coincidence
  - However, current draft is silent about reusing dialogs
- Proposal:
  - Add text similar to what MESSAGE has about using existing dialogs

Open Issues: Editorial – examples

- Example focuses on the bigger SIP events picture
  - Includes subscriptions and notifications
  - Misses publish refresh case totally
- Proposal:
  - Rewrite the section focusing on the publications/refreshes

Open Issue: Editorial – number of documents

- Definition of policies and process for defining new applications of the publish mechanism
  - Draft is silent on the exact procedures in applying PUBLISH to new event packages
  - Currently presence is tightly tied in
- Proposal 1:
  - Keep together, but still add text describing the above details in the manner of RFC 3265
- Proposal 2:
  - Split into 2 drafts; framework and presence publishing
  - The framework draft would describe the above details in a similar manner to RFC 3265
  - Presence draft would explain how these prerequisites are met for presence event package

Final note

- Let’s get this thing over with already!
  - Review and comments much appreciated ;-)
- Thank you and see you in SIP WG on Wednesday!