vCon for MIMI instant messages

draft-mahy-vcon-mimi-messages

Rohan Mahy — rohan.ietf@gmail.com

IETF 122, 21-Mar-2025

Instant Messaging Use cases

- 1) IM applications allow multiple clients for the same user (ex: laptop and mobile, or old mobile phone migrating to new mobile phone)
- Securely share message history from an existing device to a new device = nice way to share messages that the new client could not have an end-to-end session for.

- 2) Group chat with substantive discussion, then add some users
- Allow an admin to “catch them up” by sharing message history from some relevant point in the past

Changes since -01

- Adjust to make consistent with the many changes in MIMI content -06
    - add salt
    - removed lastSeen
    - in_reply_to is just a message id
    - expires is an object to handle relative expiration
    - added mimi_extensions
- allow the parties list of 0 in a text dialog entry to indicate sending to every active participant in the room.
- added a description of using party_history to track changes to the participant list.
- added room metadata, and a way to modify it
- added support for message "tombstones" - for example, for expired self-deleting messages
- use a different field from content_hash when attaching an ExternalPart that did not have a hash in the MIMI message.
- use mediatype instead of mimetype throughout
- make the case of field names more consistent

The New Look — Original, Edit, Attachment


```
{
  "type": "text",
  "start": "2022-02-08T22:13:45.019-00:00",
  "duration": 0,
  "parties": [
    1, 2, 3
  ],
  "franking_tag": null,
  "frank": null,
  "originator": 1,
  "salt": "PC16IkqDbGRiffzjeyPKIg"
  "message_id": "yjetjodyYNLBto8YDnwLkEc89W09rOuEivbnxKGfHLQ",
  "mimi_extensions": "oQGiAXgrbWltaTovL2h1Yi5leGFtcGxlL3IvU
      mwzM0ZXTENZV093eEhyWW5wV0RRZwJ4GG1pbWk6Ly9iLmV4YW1wbG
      UvdS9hbGljZQ",
  "mimetype": "text/markdown;variant=GFM",
  "encoding": "none"
  "body":
    "Hi everyone, we just shipped release 2.0. __Good work__!",
},
```

```
{
  "type": "text",
  "start": "2022-02-08T22:14:08.621-00:00",
  "duration": 0,
  "parties": [
    0
  ],
  "originator": 2,
  "salt": "26QI8adeG9TDfPvh-j1Stg"
  "message_id": "-GqJXuoaJu1Xosu-N8nt6NWzA2RLmIRY0q9jf_kvWkM",
  "replaces": "Ygye96VUeBhTO6U3D0ZllYsSwSdKB3SiRJgxAb3cdAo",
  "in_reply_to": "yjetjodyYNLBto8YDnwLkEc89W09rOuEivbnxKGfHLQ",
  "mimetype": "text/markdown;variant=GFM",
  "encoding": "none"
  "body": "Right on! _Congratulations_ y'all"
},
```

```
{
  "type": "text",
  "start": "2022-02-08T22:53:41.134-00:00",
  "duration": 0,
  "parties": [
    0
  ],
  "originator": 2,
  "salt": "bqhJBqYaTyDmgTj-QtdZPA"
  "message_id": "Tdt8UCUl1ugSLRmuObvib_4bvu_Hz3NSwaYuy-TtET4",
  "disposition": "attachment",
  "language": "en",
  "ExternalPart": {
    "mimetype": "video/mp4",
    "url": "https://example.com/storage/8ksB4bSrrRE.mp4",
    "size": 708234961,
    "description": "2 hours of key signing video",
    "filename": "bigfile.mp4",
    "content_hash":
        "sha256:OczZYpW_L0B1DsMSLJqL2jTRJKoj7fTUJ2jhnX70-00",
    "enc_alg": 1,
    "key": "aZISOs306M7n_3csR-J1cw",
    "nonce": "5VM0QcZI3PNmnquV6CUgxg",
    "aad": ""
  }
},
```

Party history

Tracking changes to the room status

The `party_history` object contains the following fields.

- `party` is the party index of the modified party. It is mandatory.
- `originator` is the party that originated the change. It is optional.
- `time` is the time that the change was effected
- `event` is the type of change. It is mandatory and for MIMI VCONs can be one of the following new event types:
  - `add`: user added by someone else
  - `self_add`: user added herself directly
  - `leave`: user leaves of their own accord
  - `remove`: user is removed by another user¶
  - `ban`: user is banned from the group
  - `update`:
- `name` is the new name of the user. It is only permitted for an update event.
- `role` is the new role assigned to the user. It is only permitted for an update event.

```
,
{
  "party_history": [
    {
      "party": 4,
      "event": "add",
      "originator": 1,
      "time": "2022-02-08T22:14:09.277-00:00"
    },
    {
      "party": 5,
      "event": "add",
      "originator": 1,
      "time": "2022-02-08T22:14:09.277-00:00"
    }
  ]
},
```

Message Tombstones

When a message has been deleted/retracted, or it expires, it can be valuable to present a record that such a message was previously present, by generating a dialog object with the following fields (which are all mandatory):

- `type` is "tombstone"
- `start` is the time the tombstone was generated
- `message_id` is the message ID of the message.
- `status` is the status of the former message, either "retracted" or "expired".

Next steps

- Remove message id, instead reference something in common in the main draft
- Once a JSON Schema is available add JSON Schema for what is added
- Anything else needed?