

MIMI Content Format

draft-ietf-mimi-content-00

Rohan Mahy, IETF 117, 26-Jul-2023

What's new

Now a draft-ietf-mimi product

- replaced threadId with topicId - Slack-style “threads” is really just a topic
- inReplyTo now has a hash of the referenced message to prevent referencing
- clarified that replies are always to a specific version of a modified message
- changed timestamp to a whole number of milliseconds since the epoch to avoid confusion
- added Security Considerations section
- added IANA Considerations section
- added change log

Issues 1/4: choice of message ID

- My working assumption
 - content has a messageid chosen by the encrypting client (“inner ID”)
 - UUID + owning provider domain.
 - Q: why include owning provider domain?
 - A: owning provider can trivially check that the user part was not maliciously duplicated; should reject messages from user where domain of message ID is the wrong domain.
 - messageid is duplicated in the “envelope” (which would be a requirement on the transport protocol)
- Other proposals?

Issues 2/4: Sort order of messages

- Draft assumes
 - sort order is (time of encryption) timestamp, then messageid for tie-breaking.
- To make that safe(ish)†:
 - First provider adds a “first-received” timestamp and signs it in the transport “envelope”
 - Receiving client compares the content timestamp and the envelope timestamp and rejects if the two timestamps are too far away
- Other options?
 - pointer to previous message(s).
 - Can’t tell if more earlier messages will arrive. Do we care?

† concept mentioned on the mailing list but not included in the draft

Issues 3/4: Mentions

- Currently we use links to im: URLs inside Markdown or HTML

```
body.contentType = "text/markdown;charset=utf-8";  
body.content = "Kudos to [@Alice Smith](im:alice-smith@example.com) "  
              + "for making the release happen!";
```

```
body.contentType = "text/html;charset=utf-8";  
body.content = "<p>Kudos to <a href='im:alice-smith@example.com'>" +  
              "@Alice Smith</a> for making the release happen!</p>"
```

- Received a handful of comments that mentions should be more explicit
- Please send text!

Issues 4/4: Attachments

- Currently we use message/external-body (RFC 4483)

```
body.disposition = attachment;
body.contentType = "message/external-body; access-type=URL;" +
    "URL=\"https://example.com/storage/bigfile.m4v\";" +
    "size=708234961;hash=10AB568E91245681AC1B";
```

- Need to specify in transport protocol how attachments gets uploaded, with what encryption, and where:
 - user's local provider (most common model)
 - room's owning provider (handful of providers do this)