

# December 2024 Minutes

## *Forum and Threaded Discussions Task Force*

---

### Housekeeping

- Julian noted that the event description in the SWICG calendar calls for a monthly meeting from 1700 to 1800 UTC, although the scheduled time is pegged to 1300 to 1400 Eastern Time (observing DST).
  - Dmitri (absent from meeting) to update event description

### as:Article and Mastodon treatment of non-notes

- Darius provided an update – been under the weather and busy with some other work-related items, but:
  - Mastodon team cautiously optimistic about upstream PR, some concerns were voiced over things like inline images
  - Hopefully by next month will have something more concrete to show to them; re: demo package
  - Evan (@evan@cosocial.ca) planning to get some people together in-person, to work together at FOSDEM in Brussels (Feb 2025); specifically to address long-form text issue
  - Discussions about this in the task force would be considered the crucial pre-work
  - Darius will update the group if something happens before January (re: code or PR package)
  - A test instance up and running, Darius plans to make it more accessible for others to check out

### Silverpill's FEP 171b is now an official draft, [open for comments on SocialHub](#)

- No specifics, just a news item.

### Context collections FEP Convergence

- Rationale for recommendation [outlined in meeting agenda](#).
- Evan (@evan@cosocial.ca) and a (@trwnh@mastodon.social) met up just prior to the WG meeting to discuss and work out differences between their FEPs; main notes:

- Using `context` to represent a reply tree is good
- Restricting usage of `context` is not the goal of 7888 or the ForumWG
- Co-exists well with 76ea's `thr:thread` property to denote a reply tree, etc.
- Recommending use of `as:context` is one good way forward
- Evan recommends that the "should" in the proposal be changed to a "may"
- **PROPOSAL: Publishers SHOULD use ``context`` for grouping related objects in a thread (but this is not the only way to use context).**
  - **RESOLVED** with 8 ayes, no nays, and no abstentions

## Brainstorming focus items for 2025

- Emelia (@[thisismissem@hachyderm.io](mailto:thisismissem@hachyderm.io)) – multiple contexts?
  - a (@[trwnh@mastodon.social](https://trwnh@mastodon.social)): we need to also handle the fact that some contexts may not resolve
  - Emelia: `as:context` can be an array of values in JSON-LD
  - a: `inReplyTo` can have multiple values too; but in general, on the producing side we generate a single value – generally expect context/thread to remain the same (singular values)
  - Sebi: "we thought a lot about multiple contexts" - led us to the conclusion of using `profiles/describes` property; per spec can only have one value
- Julian: Handling when implementors (e.g. lemmy/piefed) don't have the concept of topics
  - a: there are multiple different models of how items are grouped together; reply tree model works for large part of the fediverse; mastodon has concept of reply tree represented internally as a conversation (vs context); this could be expanded into a conversation having an owner, etc.; mastodon has the conceptual ground to build upon
  - Evan: reply trees work well on microblogs, blog comment trees, threaded posting systems, forums; other applications expect a more serial model... messenger/chat systems, where ordering of objects is not in a tree, no explicit relation between them; hashtags, locations, few other ways to use context
  - Emelia: clarify overlap between replies collection and context collection?
  - a: in general will include both ancestors and descendants; could add filters, look at tags, etc. to get subsets. If you are querying by context, you are looking for all objects related by said context
  - Evan: full tree
- Julian: Mastodon reply-tree service proposed (<https://github.com/NeuromatchAcademy/mastodon/pull/44>)
  - Julian: worried about scalability and performance of a backend service iterating through an entire reply tree; advocates that retrieving `as:context` is more performant especially if we build in some tooling for synchronization and member checking

- Emelia: historical reports of harassment due to `inReplyTo`; when looking at context including descendents, then how do we generate the tree?
  - Evan: fep 76ea goes into detail about how reply trees can be managed
  - a: answer is "who has the authority?"; who decides what goes into the collection? the `attributedTo` actor. For the replies collection, this exists IN PARALLEL TO `context`; in some ways a subset of the thread; could be a point of contention for systems that expect all objects to exist in general vs. conversation oriented
  - Julian: upends expectation that objects are independent
  - Darius: does this relate to announce leaking? recommendations that you not forward the entire object, just the ids
    - Emelia: related but different; announce leaking -- should only ever do objects by ref (by the id)
  - a: the paradigm shift is more social rather than technical -- that you cannot just rely on inReplyTo to prove that an object is approved
    - some duplication as context includes replies, but they are distinct collections. They are decided by different authorities. `replies` is decided by whoever wrote the post
- Ted (@tallted@mastodon.social): This sounds a lot like reinventing netnews, without taking the lessons that were learned from it; blurring the ideas of message store/relay/display; for all of this to work, the system has to pick up all replies, and let the client filter.
  - Julian/a: anything specific to share? lessons, etc. – definitely of interest in not repeating the same mistakes