Bridging Open Protocols: XMPP and ActivityPub Gateway via Libervia

Jérôme Poisson (Goffi)

2023-02-03 FOSDEM'24

Forewords: Libervia, XMPP, and ActivityPub

NLnet

Thanks to NLnet/NGI zero discovery

Libervia

- XMPP client, not limited to instant messaging (IM)
- Aims to be a universal communication tool
- Multi-frontends capability
- Can function as a server component, thus acting as a gateway

XMPP

- Initially designed as an IM protocol
- Can do much more
- Extensive flexibility (`X` in `XMPP` stands for e**X**tensible)
- Governed by a structured standardisation process
- Requires an XMPP server

ActivityPub (As Seen by an XMPP Developer)

ActivityPub (As Seen by an XMPP Developer)

- They reinvented the wheel!

ActivityPub (As Seen by an XMPP Developer)

- They reinvented the wheel!
- Or did they?
- Straightforward integration for HTTP services
- easier integration on existing projects
- More lenient specifications compared to XMPP
- Extensions with FEP (https://codeberg.org/fediverse/fep)?

Let's Mix Them Together

- Introducing the XMPP Gateway for ActivityPub
- Serves as a component on the XMPP server
- Facilitates access to the expansive ActivityPub network
- Integration with XMPP network and functionalities

A Few Words on Gateways

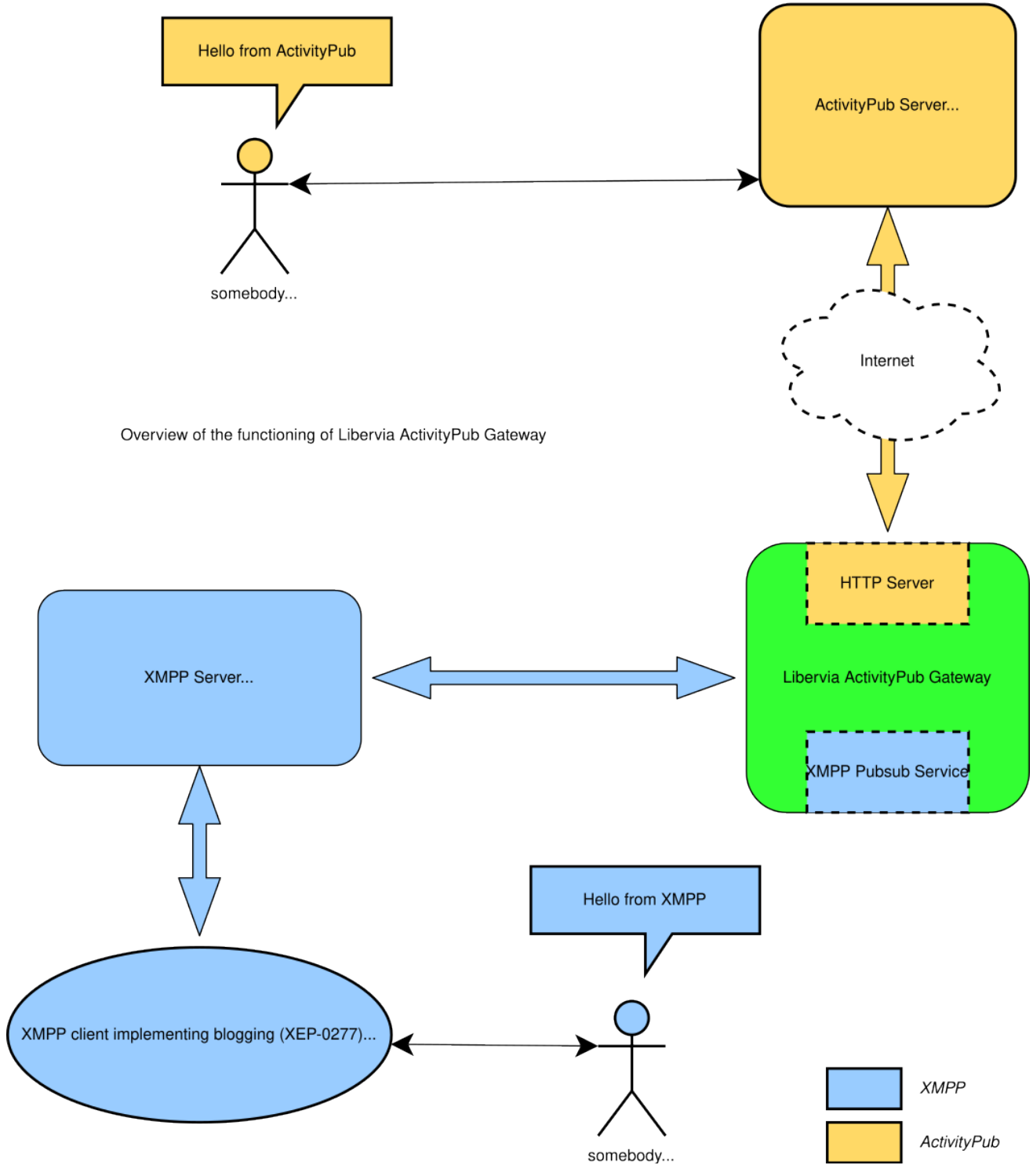
A Few Words on Gateways

- Open protocols must interoperate; it's essential socially, philosophically, and politically
- Creation and maintenance demand effort
- Necessitate higher resource allocation
- Gateway developers should target:
  - User-transparent integration
  - Comprehensive feature mapping
- XMPP's versatility allows extensive adaptability

Architecture

How the ActivityPub Gateway Functions

- HTTP server for ActivityPub interactions
- Translates public ActivityPub messages to Pubsub blogs
- Direct messages are converted to XMPP messages
- Integrates a virtual Pubsub service for XMPP
- Compatible with all XMPP clients, not just Libervia


Hello from ActivityPub
ActivityPub Server...
somebody...
Internet
Overview of the functioning of Libervia ActivityPub Gateway
HTTP Server
XMPP Server...
Libervia ActivityPub Gateway
XMPP Pubsub Service
Hello from XMPP
XMPP client implementing blogging (XEP-0277)...
somebody...
XMPP
ActivityPub

Pubsub Caching

- Maintains a local copy of parsed public ActivityPub messages
- Reconstructs message threads for UX
- Includes full-text search capabilities
- Allows for purging of outdated messages

Feature Mapping

Subscribers/Subscriptions

- XMPP subscribers are normally private to node owner(s)
- XEP-0465: Pubsub Public Subscriptions
- Privacy is a priority, with an opt-in mechanism
- Requires a specific server component

Share and “Like”

- XEP-0277 already facilitates sharing/boosting
- XEP-0470: Pubsub Attachments for generic metadata handling
- Chooses “noticed” over “like” to sidestep its inherent design flaws

Reactions

- Also leverages XEP-0470: Pubsub Attachments
- Currently not utilized in private messages, though implementation is straightforward

ID Metadata

- Bidirectional mapping
- Avatars via XEP-0084, XEP-0153 (planned)
- VCard data via XEP-0054, XEP-0292 (planned)

Calendar Events

- Compatibility with Mobilizon
- XEP-0471: Events
- Private or public events
- Supports a range from simple calendar entries to comprehensive event planning
- Includes RSVP mechanisms, attachments, and links to related blogs for discussions

The Mentions Case

- XMPP utilizes XEP-0372: References
- ActivityPub employs the `Mention` object
- XMPP references are typically sent as separate follow-up messages
- Gateway parses messages to detect AP actor mentions, bypassing Mention XEP

Summary

ActivityPub	XMPP
Direct Message	XMPP Messages
E2ee not supported	E2ee Supported (OMEMO, OX, etc.)
Public Post	XEP-0277 (blogs)
Follow/Followers	**XEP-0465 (Pubsub Public Subscriptions)**
Announce (Boost)	XEP-0277 (blogs)
Like,Reactions (Pleroma)	**XEP-0470 (Pubsub Attachments)**
Actor Metadata	XEP-0084/XEP-0153 (Avatar), XEP-0054/XEP-0292 (vCard)
Events (Mobilizon)	**XEP-0471 (Events)**

bold: XEP created for the projet

Some Difficulties

Private Messages

- End-to-end encryption (E2EE) not currently feasible with ActivityPub
- XMPP's OMEMO encryption poses challenges:
  - Keeping plain text may conflict with XMPP clients' default OMEMO settings
  - Decrypting/re-encrypting at the gateway risks a false sense of security

Collaborating with Diverse Projects

- Navigating different contexts and programming languages
- Projects include Mastodon (Ruby/JS), Pleroma (Elixir/JS), and Mobilizon (Elixir/JS)
- Challenges with setup, unhelpful error messages, and often needing to consult source code

Random Thoughts

- Lessons to learn from ActivityPub usage, noted for its strong focus on accessibility:
  - Use of like/favourite
  - Alt text for images
  - Emphasis on broad accessibility
- Points to address in either:
  - Specifications
  - Separate, informal documentation
- Influence specifications and documentation to steer development and guide end-user behaviour positively

For Developers and Users

XMPP Client Integration

- Employing XEP-0106 for JID Escaping
  - **louise\40example.net@ap.example.org**
  - *[AP]* **louise@example.org**
- Labeling ActivityPub JIDs and messages distinctly
- Caution advised for private messages due to E2EE considerations
- Utilization of Ad-Hoc commands (XEP-0055) for extended functionalities

Installation and Resources

- Official website: https://libervia.org
- Detailed documentation: https://libervia.org/documentation (section “Libervia Components”)
- XMPP chat room: libervia@chat.jabberfr.org

Video

Video

Demo Video

Conclusion

Conclusion

- *Blog*: **https://goffi.org**
  - Powered by Libervia/XMPP
  - Accessible via ActivityPub (**@goffi@goffi.org**)
  - Also available via the good old **Atom feed**
- *ActivityPub Handle*: **@goffi@mastodon.social**
- *Chat Room*: **libervia@chat.jabberfr.org** (official Libervia)
- *FOSDEM*: Meet me at **Real-time Lounge/K level 2**
- *Brython Talk*: **Sunday 16:30**, Python devroom (**UD2.218A**)
- *Feedback*: **Try out** the gateway and share your thoughts!

Thank you for attending and supporting open communication technologies!