Why Openfire

2025-10-01: Openfire is great, but we are switching to Prosody. This page is kept for posterity.

Openfire has been selected as one of the Software Components for WikiSuite. Here are the general component criteria.

Why Openfire instead of the many options in this space? To properly explain this one, it's important to first distinguish the various "real time" use cases. All use cases need chat, most need audio and screensharing. However, there are some "key distinctive features" which make some tools great at one use case, but poor for another.

If you are interested in helping develop the software, please see: Openfire Meeting and Jitsi Meet development.

See also Realtime collaboration use cases

We picked Openfire for the following reasons:

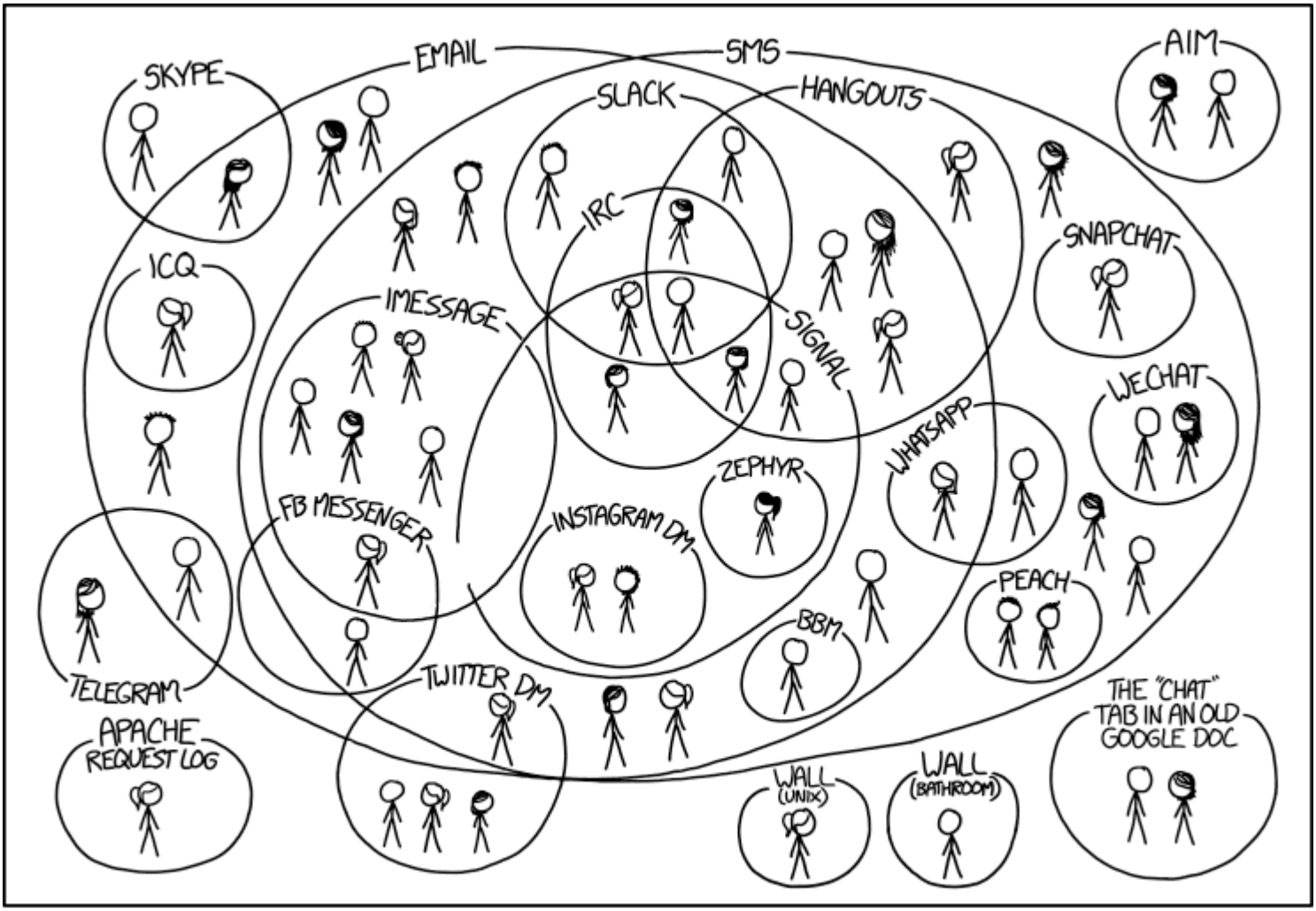
- XMPP support, and thus presence (using standards)
- WebRTC support (via inclusion of Jitsi Meet)
- Great admin panel
- Vast feature set
- The community
- Popularity
  - Over 1 million install (just via Docker pull!)
  - Millions of downloads from main site
- Generally satisfies the usual component criteria

We ultimately want WikiSuite to be awesome at covering all the use cases above, and we do so by mostly combining:

- Server-side: Openfire + Converse (XMPP) with Jitsi Meet (WebRTC)
- Desktop client: Pàdé, a Chrome extension with tons of features. See Pàdé Presentation

Why XMPP is important

https://xmpp.org/about/technology-overview.html
https://xmpp.org/about/myths.html



Why not BigBlueButton

Tiki Wiki CMS Groupware has built-in (but optional) integration with BigBlueButton since Tiki 5 (2010), and the two communities worked closely together for a tight integration and thus, for WikiSuite, this would have been the simple option.

While Openfire Meetings and BigBlueButton broadly share the same feature set (videoconferencing, screensharing, etc), there are fundamental difference. BigBlueButton is a distance education tool.

- So the focus is **one to many**.
- No presence feature (it's for a scheduled class, and not ad hoc collaboration)
- No XMPP support (ex.: Federation)
- ~~Still in 2017, the Flash version is the main one, and the HTML5 version is not ready for prime time~~ Now OK, and developed

For WikiSuite, it's critical to also cover all the other use cases above.

The leaders of both projects (Fred Dixon and Marc Laporte) met 2 or 3 times over a few years to try to find a way to make it work. But ultimately, the basic DNA / drive / philosophy / focus which made BigBlueButton successful for

distance education lead to some design choices (ex.: no XMPP) which are hard to change afterwards.

Why not Etherpad or Hackpad

- Too few features
- No XMPP support

https://github.com/dropbox/hackpad

Why not Jitsi Meet (standalone)

- Jitsi Meet is part of the solution (WebRTC), but alone is not sufficient to cover the desired use cases, which is why WikiSuite uses Jitsi Meet as part of Openfire. This is similar to how Jitsi Meet is part of Atlassian HipChat

Why not Apache OpenMeetings

Apache OpenMeetings is an interesting option with a diversity of paid support options and quite a few features, however, the focus is more about scheduled meetings or classes than ongoing collaboration. For example, XMPP is not supported.

- WebRTC support?

Why not WebHuddle

WebHuddle is inactive: https://sourceforge.net/projects/webhuddle/ and requires Java applets which are restricted by browers nowadays: WebHuddle "works in most web browsers enabled with Java."

Why not Spreed

This was the situation:

- Spreed WebRTC implements a WebRTC audio/video call and conferencing server and web client.
- No XMPP
- low activity level

But then NextCloud reactivated the project: https://github.com/nextcloud/spreed/ (well done!)
There is discussion on XMPP: https://github.com/nextcloud/spreed/issues/826 and link to a client: https://github.com/nextcloud/jsxc.nextcloud but where is the XMPP server?

Why not Hubl.in

Hubl.in is part of OpenPaaS, and is a newer option. Social networking + videoconferencing + realtime collaborative editor + others.
This is WebRTC (which is great) but it doesn't handle XMPP.

https://github.com/linagora/hublin

Why not Tox

Tox is interesting

- But no XMPP
- Serverless aspect of Tox doesn't have huge value for us because WikiSuite is a server, and we have Syncthing for P2P file sync.

Why not Retroshare

Retroshare is very interesting

- But no XMPP
- Serverless aspect of Retroshare doesn't have huge value for us because WikiSuite is a server, and we have Syncthing for P2P file sync.
- Retroshare is more focused on disseminating files, than on collaborating on files

http://retroshare.net/

Why not Zulip

Not based on XMPP
Lots of chat features but what about videoconference?

Why not Otalk

❝

> *So—tl;dr: The only pieces we are not open-sourcing are the UI, glue code, and operationalized infrastructure—but that's it!*
> *Source: https://blog.andyet.com/2015/06/09/what-is-being-open-sourced-from-talky/*

Why not Tigase

Tigase is an XMPP server and it could have been the base

Tigase and Openfire are both written in Java.
License: Tigase is AGPL, Openfire is Apache

Openfire has more users and contributors

Why not ejabberd

ejabberd is an XMPP server and it could have been the base.

ejabberd has two editions:

- Community Server (eCS) (Free/Libre/Open Source)
- Business Edition (eBE) (not Free/Libre/Open Source)

As of 2018-10, many features you'd expect for a typical project are only available in the Business Edition (eBE) (not Free/Libre/Open Source). This open core model is a disincentive for community contributions. In contrast, these features are Free/Libre/Open Source in Openfire. And if anything is missing in Openfire, we can contribute and everyone in the community will react favourably.

See also:

- https://feedback.process-one.net/support/solutions/folders/6000076557
- https://www.process-one.net/en/ejabberd/protocols/
- https://github.com/processone/ejabberd

ejabberd is written in Erlang: not a deal breaker, but less known than Java, used by Openfire.

Why not Prosody

- Prosody is an XMPP server. ~~This could have served as the foundation.~~ 2025-10-01: We switched to Prosody.

Prosody is very good for a multi-tenant use case.
Prosody is written in Lua (not a deal breaker, but less known than Java)
Prosody is lacking a web admin panel
There is no WebRTC support or plugin

Why not Mattermost

- Although there is a XMPP bridge, this is not a real XMPP server
- Active Directory (AD) / LDAP support is not in the Open Source version:

- https://docs.mattermost.com/deployment/sso-ldap.html
- https://mattermost.com/why-enterprise/

Why not Matrix.org

- Very interesting but not XMPP
  - https://matrix.org/faq/#what-is-the-difference-between-matrix-and-xmpp%3F

Why not Rocket.chat

- not XMPP

https://rocket.chat/

Why not Let's Chat

- not XMPP

http://sdelements.github.io/lets-chat/

Why not CandyJS

CandyJS is interesting. And we experimented with it to be the XMPP client and contributed the conversion to Bootstrap for responsive design.

But:

- it's mostly designed for team chat (vs 1 on 1 chat)
  - Converse has both a pop up chat and a full page interface
- activity level is low

https://candy-chat.github.io/candy/

Why not Signal

Both the server and the clients are Open Source.
It's backed by a Foundation
https://en.wikipedia.org/wiki/Signal_(software)

Signal is a very interesting option, but we preferred XMPP

Why not Telegram

The server part is not Open Source, but the clients are: https://telegram.org/faq#q-why-not-open-source-everything

Telegram is an interesting option, but we preferred XMPP

Why not proprietary software or Software as a Service (SaaS) like Skype, Google Hangout, etc.

Why Free Libre Open Source software

Related links

- https://blogs.gnome.org/tbernard/2018/05/16/banquets-and-barbecues/

See also

- Why
- Why Cyrus IMAP and Cypht
- Why Free Libre Open Source software
- Why Not
- Why Syncthing
- Why Tiki Wiki CMS Groupware