

# A conceptual model of ATProto and ActivityPub

📅 November 4, 2024(<https://fediversereport.com/2024/11/04/>)   👤 Laurens Hof  
Bluesky (<https://fediversereport.com/category/bluesky/>), Deep dive  
🔗 (<https://fediversereport.com/category/deep-dive/>), Fediverse  
(<https://fediversereport.com/category/fediverse/>), ReStyle  
(<https://fediversereport.com/category/restyle/>)



If you were to design an open social networking protocol, what would it look like? Which metaphors and comparisons would you use to get a general idea of how the network functions? What would you answer if people ask if your network is decentralised and federated?

This article is not a deep technical explanation about how either ActivityPub or ATProto work. Instead I want to explain how these two protocols have different conceptual models of what an open social network looks like. These conceptual models differ more from each other than people expect. This causes people to apply concepts from the ActivityPub world to ATProto, even though these concepts do not fit with ATProto.

One of the main subjects of discussion (<https://pluralistic.net/2024/11/02/ulysses-pact/>) recently has been whether Bluesky is decentralised and if it is federated. Answering these questions requires a clarity on how ATProto differs conceptually from ActivityPub. Decentralisation and federation are valued for their impact on power structures, but there are multiple ways structure power in open social networks.

Here is a quick summary of both models, to help frame the article:

- The conceptual model of ActivityPub resembles that of email: independent servers sending messages to each other.
- The conceptual model (<https://bsky.app/profile/danabra.mov/post/3la4wq6ku3k22>) of

ATProto resembles that of the web: independent sites publish data, and indexers aggregate this data into different views and apps.

# A conceptual model of ActivityPub and the fediverse

The fediverse<sup>1</sup> is a network of independent social networking sites that use the ActivityPub protocol with each other. In the fediverse, each social networking site (often called a server or instance) is an independent network that can exist on its own. You can set up your own Mastodon server, not connect to any other server, invite some of your friends, and have a fully functional social networking site.

Because each fediverse server is its own independent and complete social networking site, it means that each fediverse server contains all the components that are needed for a social network to function. A fediverse server:

- Owns your identity.
- Stores your data.
- Transforms protocol data into a site with a timeline that you can look at.

Most people run a fediverse server to join a larger network of interconnected social sites: the fediverse. The ability for anyone to run a fediverse server is what makes the fediverse decentralised. To interact with the rest of the larger network, a fediverse server also performs a fourth function:

- Communicates with the rest of the network. In ActivityPub terms: the server gives you an inbox and outbox, and messages flow between these inboxes and outboxes on other servers.

This communication between servers is the ‘federation’ part of the fediverse.

## Decentralisation and Federation in the fediverse

The main reason for creating a super-network of independent social networking sites is to build other systems of governance. In many ways, the fediverse is a response to the centralised governance under a single billionaire of the current Big Tech platforms. The fediverse creates a governance structure where each social networking site is its own authority; a fediverse server has authoritative control over the users on their site, but no authority over any of the other ~30k independent servers.

Decentralisation and federation are crucial for the functioning of the architecture of the fediverse. Decentralisation means that anyone can set up their own social networking site, and federation means that all these independent sites can connect with each other without a single central authority. While these terms often get used as being inherently valuable, I see them as technical solutions to solve a governance problem: how can we build a social network without a single central authority?

Human nature is a funny thing however, and technical solutions to limit authoritative control usually means that chokepoints simply pop up in other places; whether that's the software that's used 75% of users being governed by a self-styled (https://www.theverge.com/23658648/mastodon-ceo-twitter-interview-elon-musk-twitter) 'benevolent dictator for life', or server admins having full centralised control over the users on their server.

## A conceptual model of ATProto and the ATmosphere

Bluesky PBC<sup>2</sup> is building an open social network with the explicit goal (https://bsky.social/about/faq) that the network should not be owned by a single company. The protocol they use is called AT Protocol, often called ATProto. The network built with ATProto is known as the ATmosphere. The approach Bluesky PBC takes to get there differs from the fediverse. The conceptual model of ATProto is that every account is its own independent place in the ATmosphere<sup>3</sup>, and every app is an aggregator that takes data from all the places in the ATmosphere and uses them to create their own service.

Every account has its own place to store their data, a Personal Data Server<sup>4</sup>. This PDS is simply a database that contains all your own data on ATProto: the posts you made on Bluesky, as well as your RSVP to an event on Smoke Signal (https://smokesignal.events/).

Similarly, every application on ATProto functions as an aggregator, like how Google aggregates data from across the web. An ATProto 'app' (like Bluesky) gathers data from all PDSes in the network<sup>5</sup>, processes it, and provides the user with a 'view' of that data. These applications on ATProto are called AppViews.

In the case of Bluesky, they take all the microblogging posts stored on all the PDSes in the ATmosphere, aggregate them together. This aggregation allows Bluesky to give users the Discover feed, count the number of likes on a post, among other things. It is then presented (the 'view') to the user in a format that resembles Twitter<sup>6</sup>. But other AppViews are also possible: WhiteWind (https://whtwnd.com/) is a blogging platform built on ATProto: it allows you to write blogs, and if you use WhiteWind to write a blog posts, these posts are also stored in the same PDS that stores your Bluesky data. The WhiteWind application (AppView) aggregates

data from the entire ATmosphere, combining both WhiteWind blog posts and Bluesky microblogs. The View WhiteWind presents on their site is blog posts, and with Bluesky's microblogs functioning as a sort of comment section<sup>7</sup>.

In short, the conceptual model of ATProto has some resembles to how the web works. The web has many independent websites, and indexers such as Google aggregate the websites and present it back to users in their own 'view'. Similarly, the ATmosphere consists of many PDSes, and AppViews aggregate this data into user-facing products.

## Independence, openness and power on ATProto

As the question 'Is Bluesky decentralised and federated' is the greatest thread in the history of forums, and the discussion is still not locked by moderators after 12,239 (<https://knowyourmeme.com/memes/the-greatest-thread-in-the-history-of-forums-locked-by-a-moderator-after-12239-pages-of-heated-debate>) pages of debate, it's worth taking a step back at what these concepts are meant to accomplish. Decentralisation and federation in the fediverse mean an open network that anyone can join without any centralised control.

Anyone can run their own PDS and fully participate in the network without central permission. Anyone can run their own AppView<sup>8</sup>, and build their own product in the ATmosphere, without needing permission by any centralised authority. They can even reuse Bluesky's data for their own purposes, like WhiteWind does. On first glance, this seems pretty decentralised.

The question of federation becomes more complicated: who can communicate with what exactly? Any AppView can communicate with the PDSes, is that federation? The PDSes cannot communicate with each other directly (so no federation?) but do so via AppViews (so maybe federation?) What about AppViews communicating with each other? Picosky (<https://psky.social/>) and IRCsky are two different AppViews that both allow you to chat over ATProto, and see the same chat messages. Are these two AppViews federated? And how many individual parts of the system need to federate before you can describe the entire ATmosphere as federated? I don't know the answer to all of this, but I'm personally trending towards: are we even asking the right questions here?

To make matters even more complicated; many people are not asking the question 'Is the ATmosphere decentralised', but are wondering 'Is Bluesky decentralised'? Here the ATmosphere take a different direction than the fediverse. The answer for the ATmosphere is not: 'there should be many versions of the Bluesky app so users can switch to another app'. Having many instances of the Bluesky app provides no real additional benefit to making the network more open and less controlled by a single point of authority.<sup>9</sup> Instead, the conceptual model of how the ATmosphere is defending itself against an AppView turning 'bad', is to have

competing different AppViews where people can switch to instead. Bluesky PBC hopes that there will be a hypothetical GreenCloud AppView, which does microblogging slightly differently than Bluesky. This way, people have a different app they can use, in case the Bluesky app does not suffice.

The hypothetical GreenCloud microblogging app built on ATProto does not actually exist. But the code for the Bluesky app is available as open source<sup>10</sup>, anyone can run their own Bluesky if they want to. The interesting problem here is that nobody has done so: running a competing service to Bluesky is totally possible, but why would you? It costs money, time and expertise to do so, and there is little gain to doing so.

How applicable the concepts of decentralisation and federation are to the ATmosphere is debatable, but they are used as an approximation for the core question: how is power distributed in the network? And Bluesky and the ATmosphere make it clear that technological architecture can only help so much here: Sure, you can be completely independent of Bluesky PBC on the ATmosphere, as everything is open. But in the end, 99% of users are exclusively on infrastructure owned by Bluesky PBC. No technological architecture can compensate for that degree of the power distribution.

This is not lost on Bluesky PBC and their investors either. Blockchain Capital, lead investor in the series A, has the investment thesis that the value is in growing the ATmosphere, writing (<https://www.blockchaincapital.com/blog/bluesky-13m-users-and-growing-our-investment-in-blueskys-re-imagined-social-network>) that they are “investing in more than a product but rather a vision of what social infrastructure could be”. The challenge here is clear: get more people to build products on ATProto. The sales pitch is attractive: there is a social graph with 13 million accounts that are free for the taking for any developer to build on top upon. The sales pitch is also unusual, however: Bluesky PBC is asking for other organisations and companies to be their competitors so that both can contribute to a growing ecosystem. How this will work out remains an open question.

## On Identity

Technological solutions to prevent control of chokepoints usually mean that these chokepoints simply pop up in different places, and the ATmosphere is no different than the fediverse in this regard. This explanation of how the ATmosphere works, is missing a crucial part: how Decentralised Identity works on ATProto. Explaining how the system in detail is the subject of my next article. And that system might just be both more centralised than people expect, more decentralised than people think, and it's most centralising aspect might just be... a clock.

## Notes

1. The fediverse is defined here by the Mastodon-dominant supernetwork that mostly uses ActivityPub and is mostly used for microblogging, with some link-aggregators on Lemmy, some video on PeerTube on the side. I'm aware that this definition does not cover the entirety of the network, as well as that you can contest every word in that definition. But its a close enough approximation for how the word is used in casual day-to-day life. ↩
2. For clarity, 'Bluesky PBC' refers to the Bluesky Public Benefit Company, while 'Bluesky' refers to the microblogging app made by Bluesky PBC. ↩
3. See also Paul Frazee's thread (<https://bsky.app/profile/pfrazee.com/post/3l6xwi52zti2y>) on how every user is basically a website. I hesitate to use the term website here, as that comes with certain preconceived notions of what a website is, and a PDS repository is different in some manners. Maybe over time it will turn out that the equivalence of 'PDS repo' with 'website' will makes sense to people however, I'm unsure. ↩
4. Technically, a repository on a PDS, a PDS can contain the repositories for many different accounts. ↩
5. This is mostly done via a Relay. Relays are an optional part of the network, and can best be seen as an extension of an AppView. This extension is in itself also flexible, multiple AppViews can use the same Relay. ↩
6. The extra step here is that the AppView sends data to your client, such as the bsky.app website, the official mobile clients or a third-party client like deck.blue ↩
7. Example of a WhiteWind blog that combines Bluesky's microblogs here ([https://whtwnd.com/bnewbold.net/entries/Notes%20on%20Running%20a%20Full-Network%20atproto%20Relay%20\(July%202024\)](https://whtwnd.com/bnewbold.net/entries/Notes%20on%20Running%20a%20Full-Network%20atproto%20Relay%20(July%202024))). ↩
8. And/or run their own Relays, which multiple people are in fact doing. ↩
9. This is the problem that the fediverse has; there are 10k Mastodon servers, accounting for 80% of active users, but the software is controlled by a single person. Many concurrent deployments of the same software does not reduce the amount of control that this software has, it arguably increases the control instead. ↩
10. The core functionalities all are, some parts are not. ↩