Hungryserv

A Homeserver Optimized for Unfederated Use-Cases

Sumner Evans

26 August 2022

Beeper

A bit about me

My name is Sumner, I'm a **software engineer at Beeper**.

- I graduated from Colorado School of Mines in 2019 with a master's in CS.
- I teach as an adjunct professor at my alma mater.
- I enjoy skiing, volleyball, and football (soccer).
- I'm a 4th degree black belt in ATA taekwondo.

I became interested in Matrix when I was the chair of the ACM chapter at Mines. I was looking for an open source chat platform for the club to use, and Matrix fit the bill!

A bit about me

My name is Sumner, I'm a **software engineer at Beeper**.

- I graduated from Colorado School of Mines in 2019 with a master's in CS.
- I teach as an adjunct professor at my alma mater.
- I enjoy skiing, volleyball, and football (soccer).
- I'm a 4th degree black belt in ATA taekwondo.

I became interested in Matrix when I was the chair of the ACM chapter at Mines. I was looking for an open source chat platform for the club to use, and Matrix fit the bill!

What I work on at Beeper

I am on the newly created *Scaling* team.

Our current objective is to prepare Beeper for rocket-ship growth.

I was previously part of the *Bridges* team.

Notable projects include:

- Writing the LinkedIn bridge
- Implementing massive stability improvements in the Signal bridge
- Implementing incremental infinite backfill in our WhatsApp and Facebook bridges

What I work on at Beeper

I am on the newly created *Scaling* team.

Our current objective is to prepare Beeper for rocket-ship growth.

I was previously part of the *Bridges* team.

Notable projects include:

- Writing the LinkedIn bridge
- Implementing massive stability improvements in the Signal bridge
- Implementing incremental infinite backfill in our WhatsApp and Facebook bridges

Overview

1. A bit about Beeper
2. A bit about Beeper's current architecture
3. Hungryserv

This talk is interactive!

If you have questions at any point, feel free to interrupt me.

A bit about Beeper

Beeper’s mission

Our mission is to:

> *make it easy for everyone on Earth to chat with each other.*

We specifically chose the word “chat” rather than “communicate” because we are focusing on *people talking to one another*.

Beeper’s mission

Our mission is to:

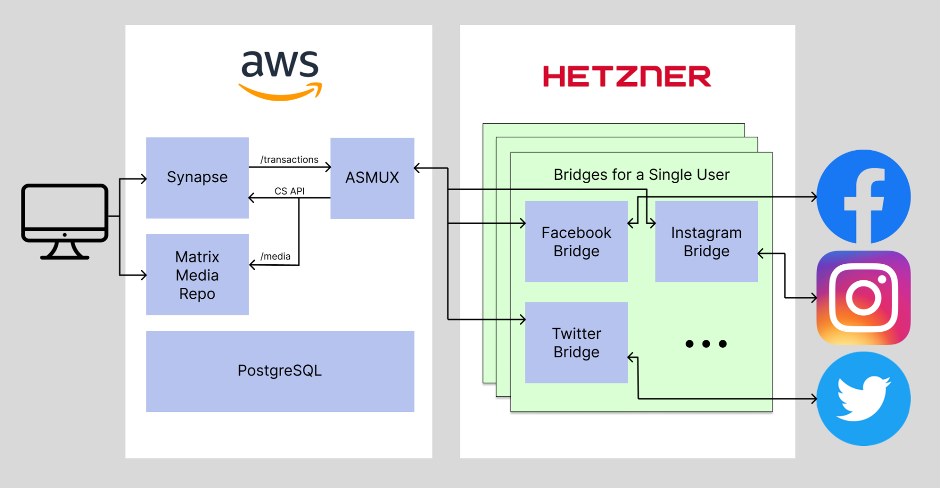
> *make it easy for everyone on Earth to chat with each other.*

We specifically chose the word “chat” rather than “communicate” because we are focusing on *people talking to one another*.

How we are getting there

A bit about Beeper’s current architecture

A diagram



Each user gets their own bridge for each network they connect.

Advantages of our architecture

- Each users’ bridge is **isolated**
- We can deploy **different bridge versions** to different sets of users
- Bridges can be stopped and started individually
- Double puppeting
- Encryption

Advantages of our architecture

- Each users’ bridge is **isolated**
- We can deploy **different bridge versions** to different sets of users
- Bridges can be stopped and started individually
- Double puppeting
- Encryption

Advantages of our architecture

- Each users’ bridge is **isolated**
- We can deploy **different bridge versions** to different sets of users
- Bridges can be stopped and started individually
- Double puppeting
- Encryption

Advantages of our architecture

- Each users’ bridge is **isolated**
- We can deploy **different bridge versions** to different sets of users
- Bridges can be stopped and started individually
- Double puppeting
- Encryption

Advantages of our architecture

- Each users’ bridge is **isolated**
- We can deploy **different bridge versions** to different sets of users
- Bridges can be stopped and started individually
- Double puppeting
- Encryption

Disadvantages of our architecture

- We have to run a **lot** of bridges
- Synapse is not designed for a dynamic numbers of bridges
- If two users join the same chat on an external network, we end up with two rooms with the same data
- Synapse becomes a bottleneck for all traffic

Disadvantages of our architecture

- We have to run a **lot** of bridges
- Synapse is not designed for a dynamic numbers of bridges
- If two users join the same chat on an external network, we end up with two rooms with the same data
- Synapse becomes a bottleneck for all traffic

Disadvantages of our architecture

- We have to run a **lot** of bridges
- Synapse is not designed for a dynamic numbers of bridges
- If two users join the same chat on an external network, we end up with two rooms with the same data
- Synapse becomes a bottleneck for all traffic

Disadvantages of our architecture

- We have to run a **lot** of bridges
- Synapse is not designed for a dynamic numbers of bridges
- If two users join the same chat on an external network, we end up with two rooms with the same data
- Synapse becomes a bottleneck for all traffic

Stats

None of that traffic is federated, but it has to go to Synapse which is designed with federation front-and-center!

Solution: build a homeserver dedicated to unfederated traffic

Hungryserv

Why is it hungry?

Because it’s unfed(erated)!

Our new architecture