

A Vision for a Next-Generation Timeline

Jade & Ginger

About us



Jade

@jade:ellis.link

<https://jade.ellis.link>

JadedBlueEyes



Ginger

@ginger:gingershaped.computer

<https://gingershaped.computer>

gingershaped



About Continuuity

A community-driven Matrix homeserver • <https://continuuity.org>

Continuuity provides a full Matrix homeserver in a single service:

- **Easy to deploy** to supported platforms, and **easy to administrate**
- **Lightweight enough to run anywhere**
- **Full support for core Matrix features**
- **No external services required**
- **Works with MatrixRTC** through LiveKit and lk-jwt-service

We operate as a volunteer project supported by contributors around the world.

There's a workshop on setting up your own instance this afternoon.

We'll also be talking more about Continuuity on Thursday in Peter!

Next-generation timeline

What's that?

A collection of ideas we think can solve some longstanding issues with Matrix's timeline.

On Timelines

Elsewhere...

Store and Forward

Each client node owns the messages

Single Source of Truth

One authority owns messages

So what's Matrix doing?

Has this ever happened to you?

Surely time didn't stop while I was gone...

Saturday, 28 March 2026

08:58  AK91 joined the room

11:35  Matthew changed their displayname to Matthew|away

12:18  nex 🦋 left the room

Sunday, 11 October 2026

22:53  nex 🦋 joined the room

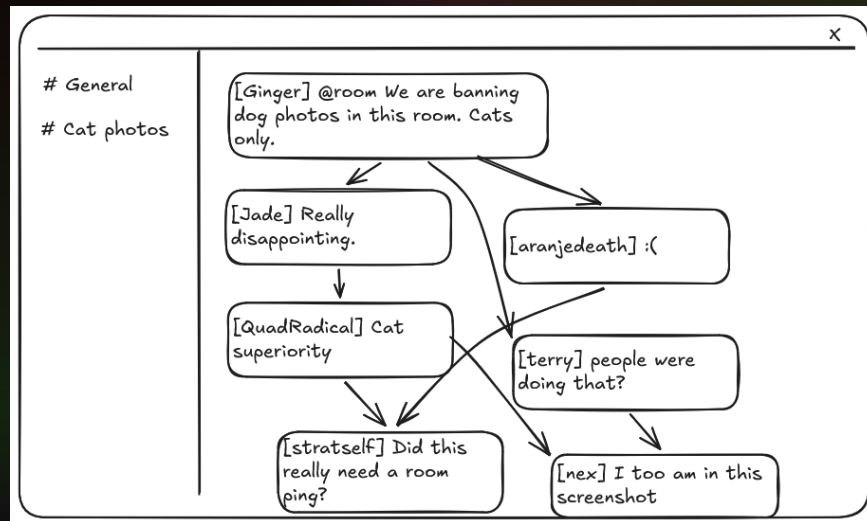
On Timelines

What *is* a room, really?

Because of physics and latency, a room isn't a linear stream of one thing after another. It's a graph that creates a partial ordering.

Clients need *some* sort of reasonable linear representation of a room.

Imagine if your Matrix client looked like this!



Next-generation timeline

What causes these issues?

Most servers take a very simple approach:

Clients see events in the same order the server does.

(for the most part)

If your server gets an event late, it'll get put right at the end of the room's history.

Continuuity can avoid this in some cases where it has to download recent missing events, but it can't do anything about events that show up late.

This also affects other servers

The `/backfill` endpoint returns a list of events too. Existing servers use the same ordering they do for clients when responding to backfill requests, and don't try to reorder incoming events when doing backfill themselves.

Stitched ordering

<https://codeberg.org/andybalaam/stitched-order>

by Andy Balaam, et al.

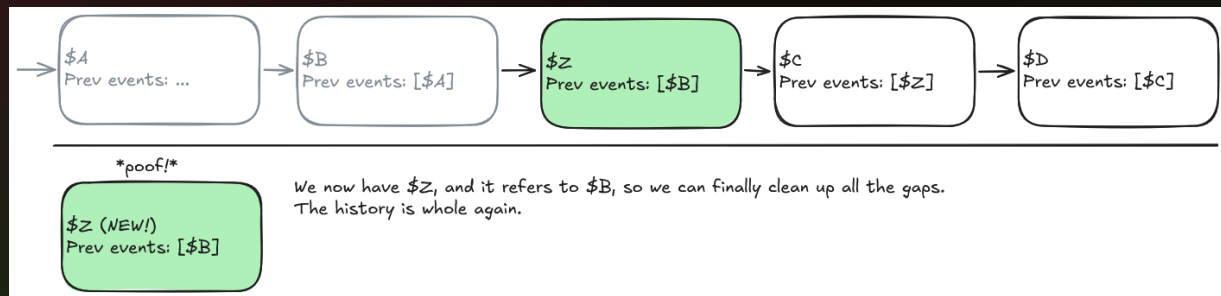
Stitched ordering

<https://codeberg.org/andybalaam/stitched-order>

Instead of a single contiguous history, we have two kinds of history entries: **events** and **gaps**.

Suppose we receive an event from another server with a prev event that we haven't gotten yet.

We add the event to the end of the history, and also add a gap before it to represent the missing prev event.



This is a pretty simple case. The algorithm can handle much more complex situations, with events that have lots of prev events.

Workarounds

Continuuity event storage 101

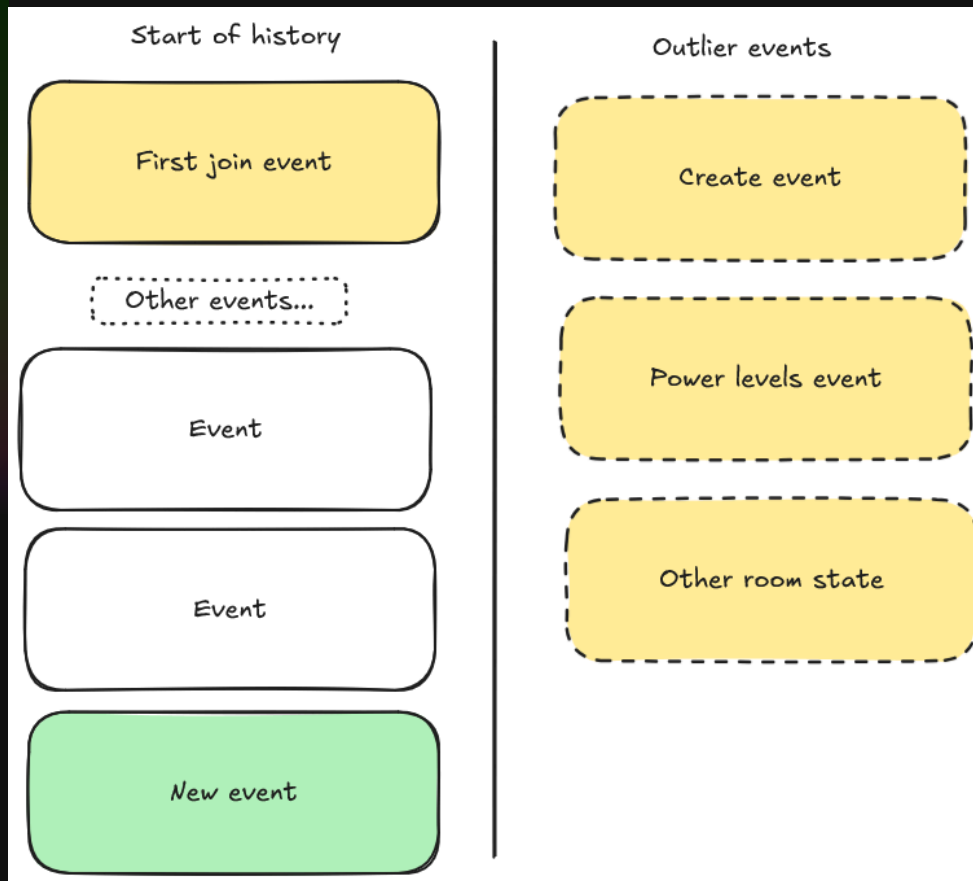
Workarounds

Outside context problem

Continuuity splits a room into a **history list** and a pile of **outlier events**. Outlier events can be used to authenticate other events but don't exist anywhere in the history.

Outlier events are moved to the history list if another server provides them during backfill.

The history *cannot* have any holes or gaps!



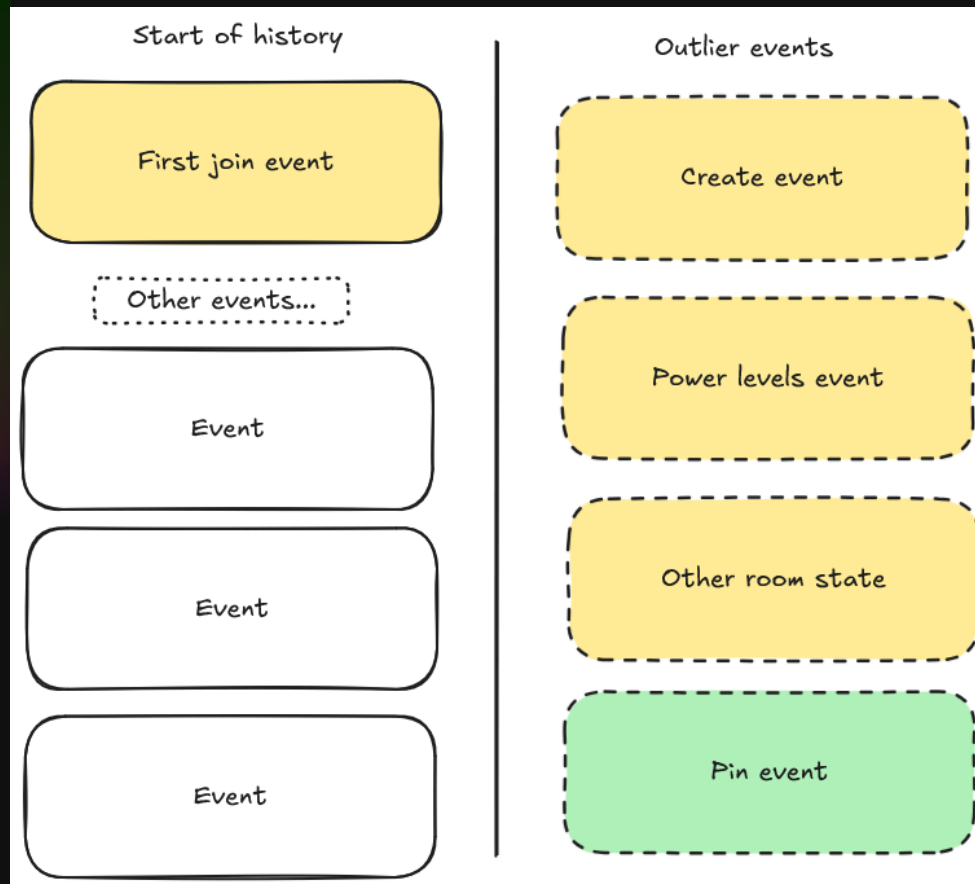
Workarounds

Outside context problem

When a client asks for a pinned event that Continuuity doesn't already have, it's downloaded from another server as an **outlier**.

Continuity cannot provide context for it because it doesn't know where it goes in the history list, and the history list can't have gaps or missing events anywhere except at the start or the end.

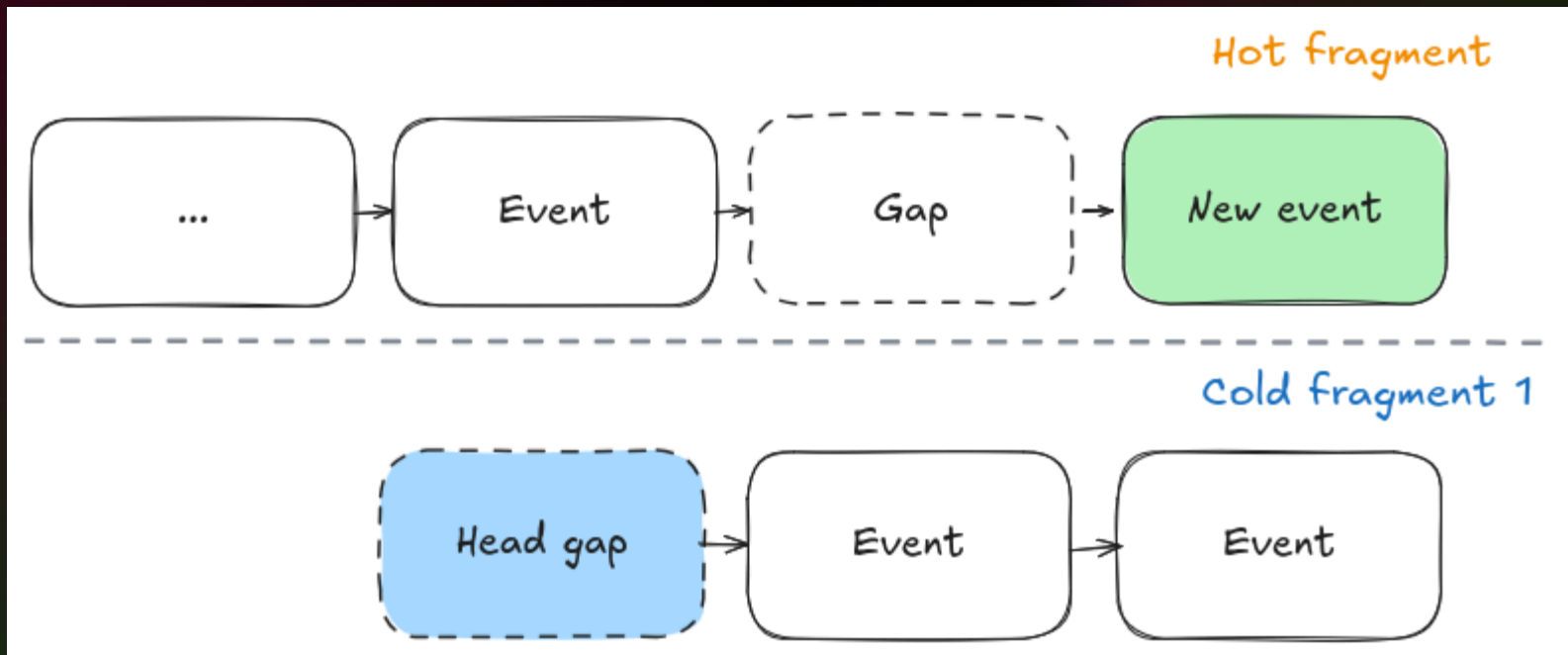
But what if there was a better approach?



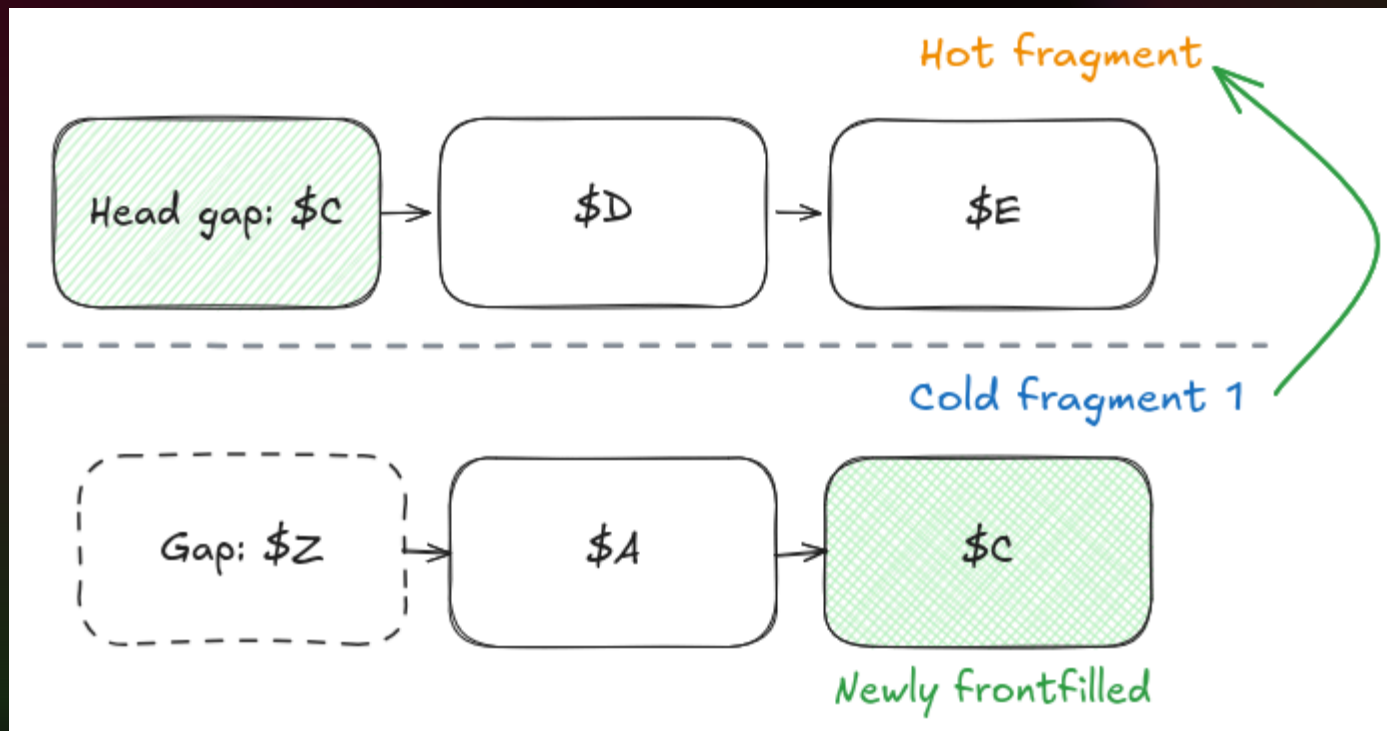
Timeline fragments

A piece of history

Timeline fragments



Timeline fragments



Security and state DAGs

Your Turn

♥ Talk to us

<https://continuuity.org>

Join our Matrix space!



Or find the team roaming around Matrix Conf

Questions!