

How can we speed up launch as much as possible?

1. Lazy loaded members
2. Lazy loaded receipts
3. Go faster stripes on the transport
 - Make sure we have HTTP/2 enabled given cloudflare gives us it for free serverside.

Also:

Initial sync that is filtered to just the room summaries (name, avatar, unread msgs, total member count, topic(?))

- We would get toDevice & ephemeral messages (presence?) here which we **have** to process otherwise we'll lose them forever.
- => Show a room list
- => View a room
 - We can't call /roomInitialSync or /state as they don't take a sync token
 - → so we could add a sync token and a filter (to lazy-load members) to /state?
 - → or we could have a per-room /sync (and possibly obsolete /state?)
 - In either of these, we would then expand the sync filter and keep polling with the wider filter.
- => Switch room.
 - ...same as View.
 - ...which is going to be horribly slow if you see a stale snapshot which only updates once /state returns.

<Much discussion>

Two options:

- We do a highly filtered initial sync
- We set a background /sync thread going which syncs in batches of rooms (10 at a time or whatever)
- Option 1:
 - Spin up a new /sync thread if the user views a given room, which lasts for as long as they are viewing that room, which lets us have realtime updates for that room. The room data is duplicated in the background /sync thread.
- Option 2:
 - When the user views a room, we cancel the current background /sync request, and start a new one from the same since_token but specifically filtered to the room they are looking at.
 - Meanwhile, all /sync responses give incremental messages from all rooms... (and full context for the requested prioritised rooms)

- => This is a hybrid between initial & incremental sync. You get initial sync for the batch of rooms you requested, and incremental updates for everything else.
- So, rather than just being blocking initial /sync and then lots of incremental /syncs, we do a quick initial /sync, then a hybrid /sync phase (where we prioritise rooms the user is looking at) and then finally go into normal incremental /sync.
- => This is effectively paginated sync, where we ask /sync to return in batches of rooms.
- Gotchas:
 - Need to track carefully whether a message is 'live' or 'historical' or not.
- This works for granularising incremental sync just as well as initial sync.
- This could also kill off peeking (i.e. room/initialSync and /events) if we can just arbitrarily add room_ids parameters into our /sync request.

