Robust and Privacy-Preserving DS

Brendan McMillion
IETF 120 / July 23, 2024

Problems MIMI Needs to Solve:

- **Access Control**
  Only members of a group should be able to see/send messages
- **Spam & Abuse Filtering**
  Prevent users from receiving, or provider from storing abusive messages
- Many other things…

Current Approach:

Provider inspects Commit messages to learn & enforce & modify membership.

Why that's not great:

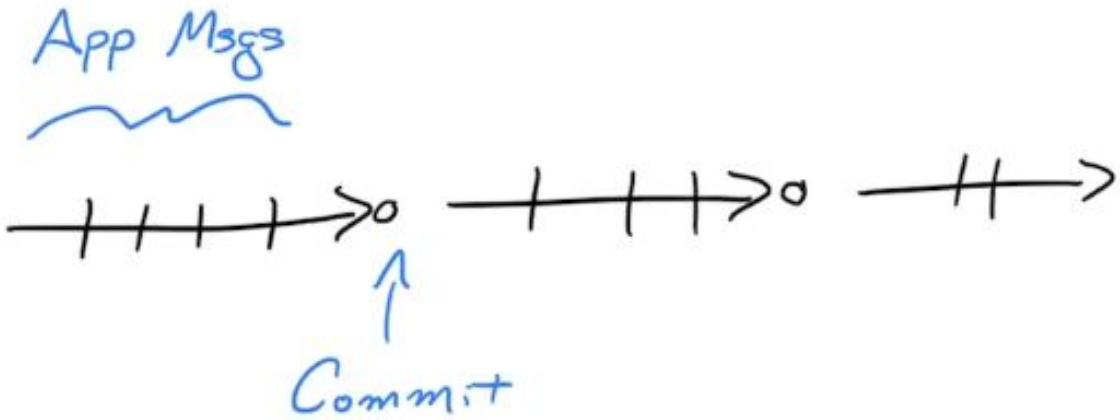
- **Invalid Commits**
  Providers can end up "enforcing" acceptance of broken Commits
- **No Membership Privacy**
  Requires that all handshake messages must be public

What about pseudonyms?

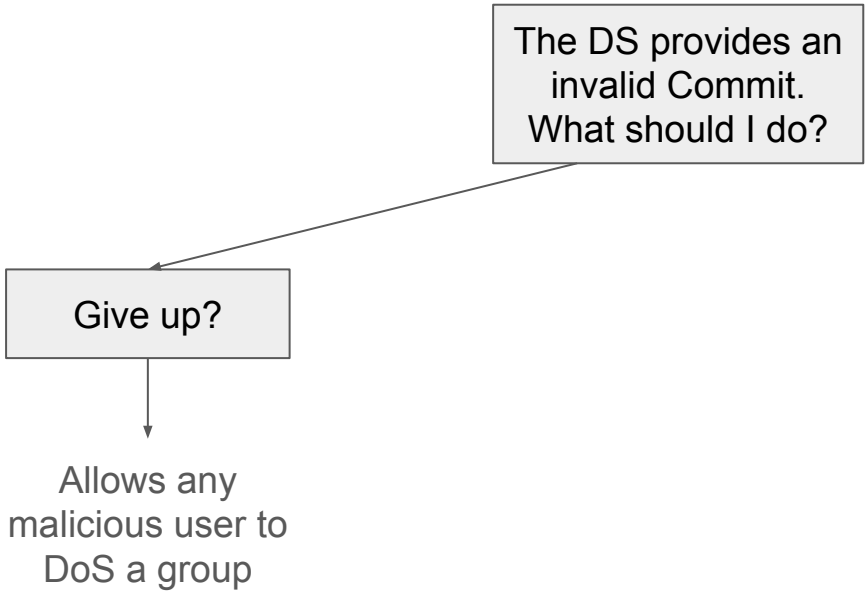
- Implementation baggage in terms of who can be added to groups and when
- Same issue with invalid Commits

What would be better?

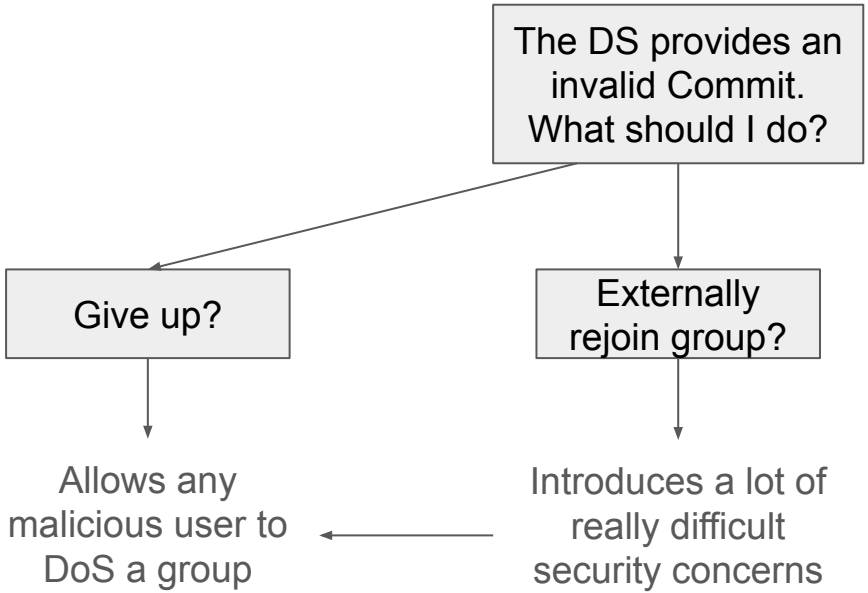
Ideal MLS Group



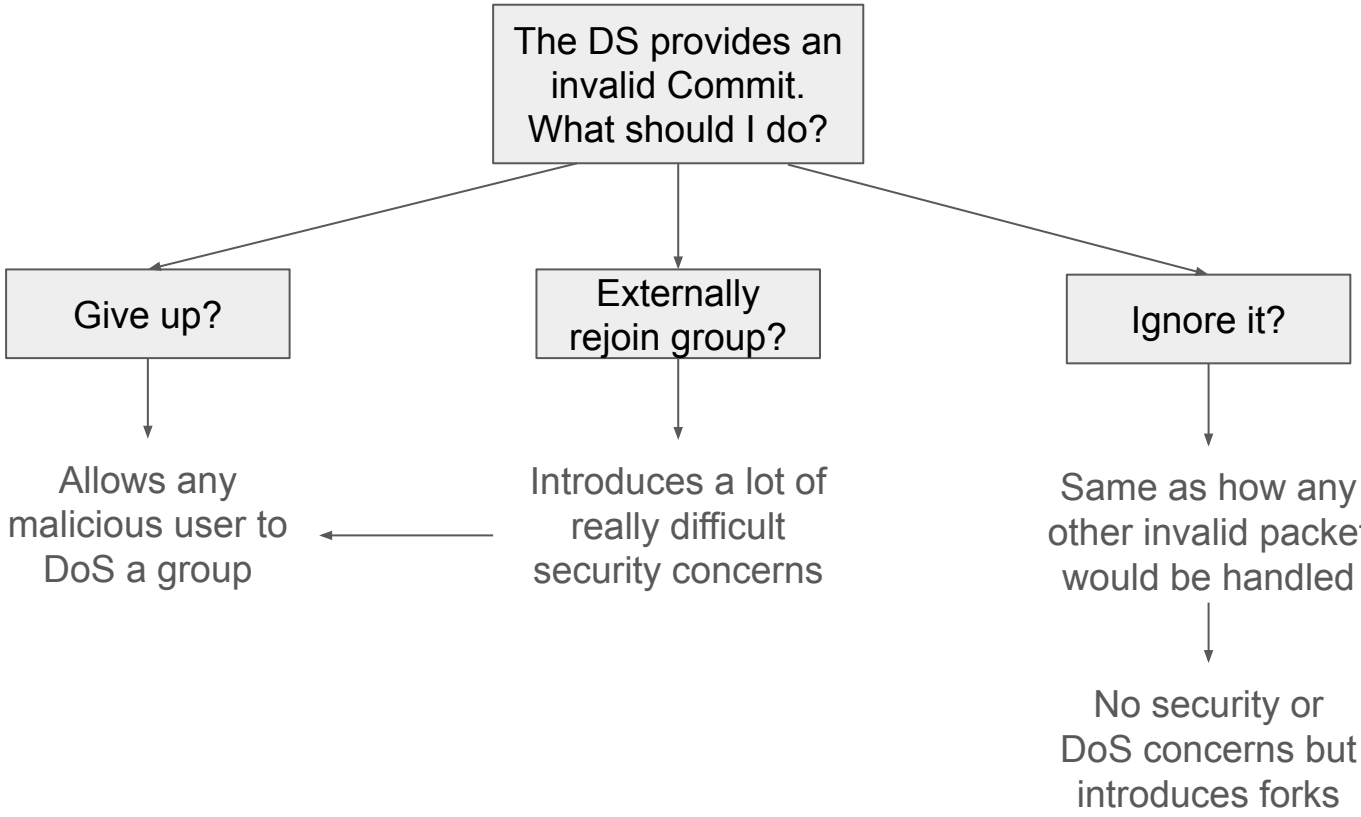
In reality, forks are inevitable:



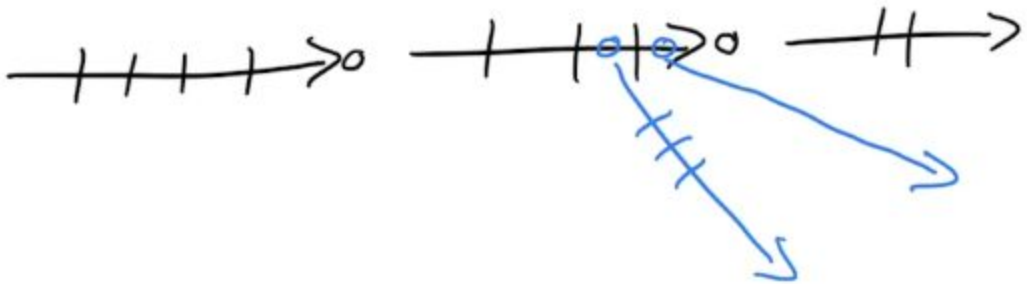
In reality, forks are inevitable:



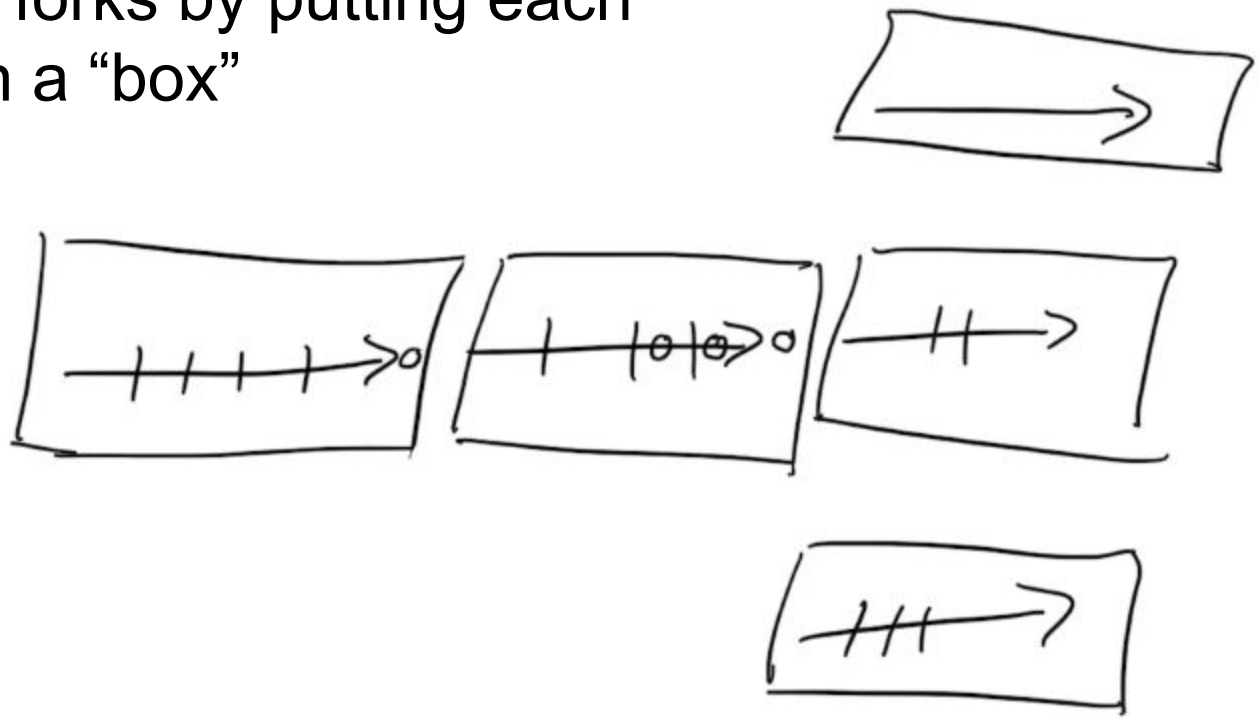
In reality, forks are inevitable:



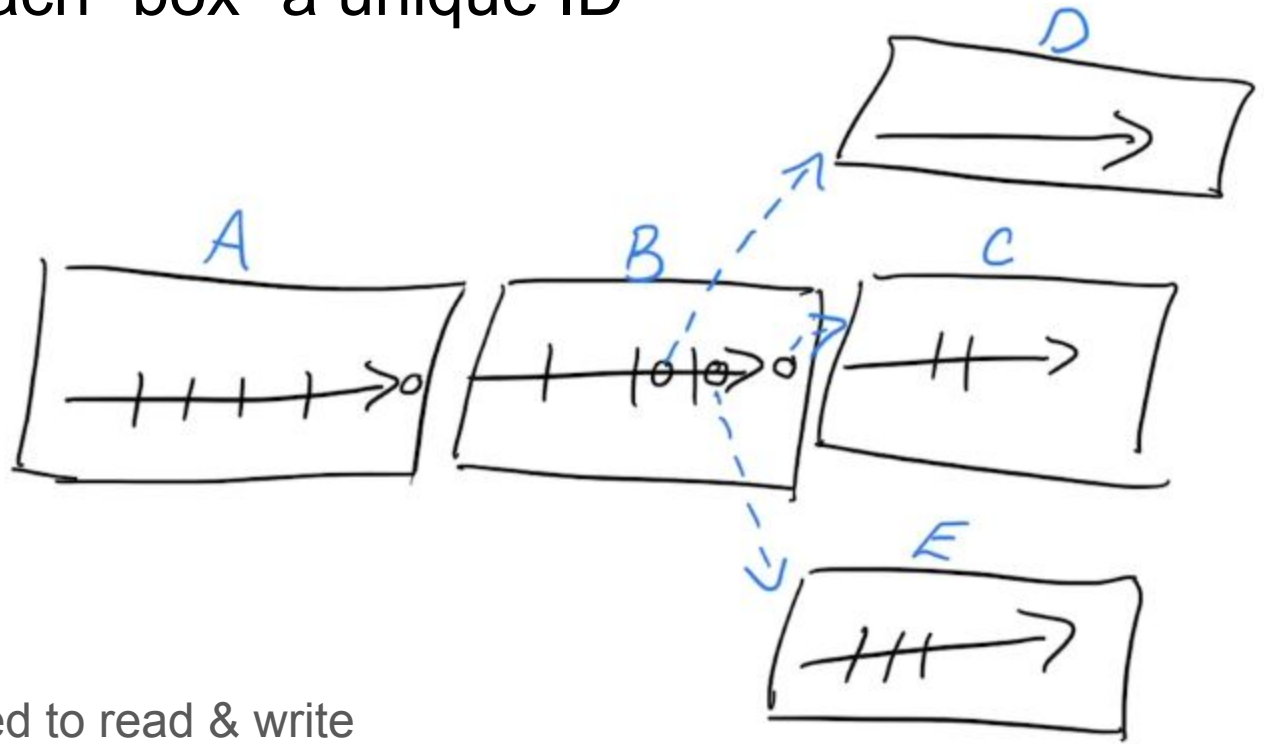
In reality, forks are inevitable:



Support forks by putting each epoch in a “box”

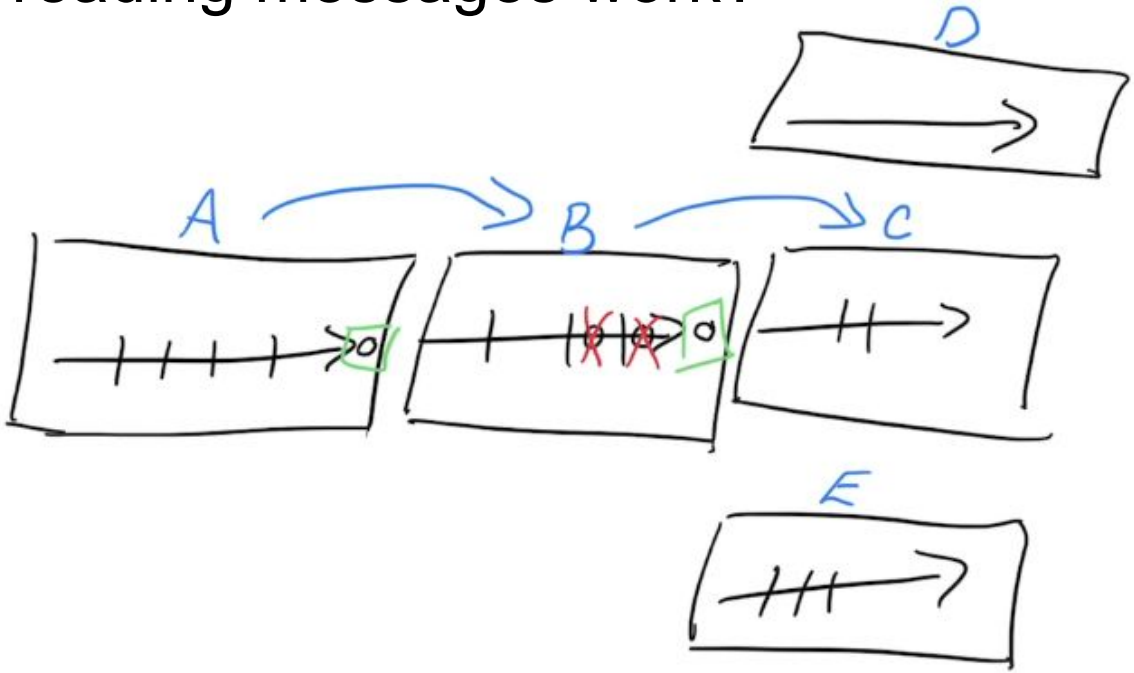


Give each “box” a unique ID

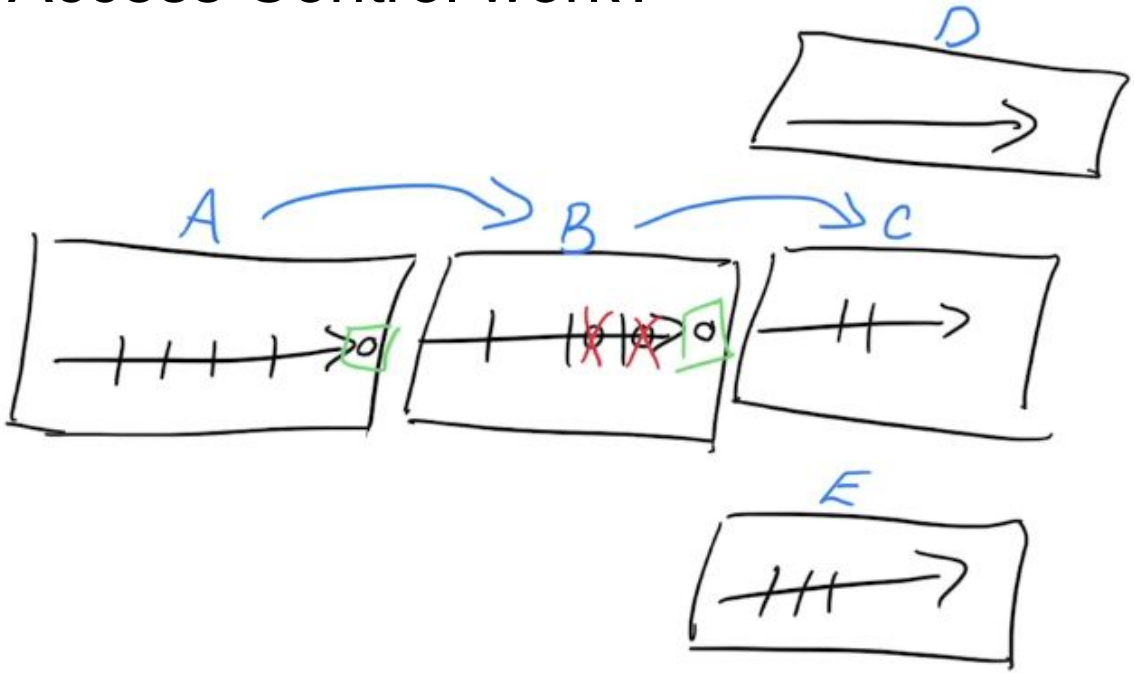


ID is used to read & write messages in a given box

How does reading messages work?

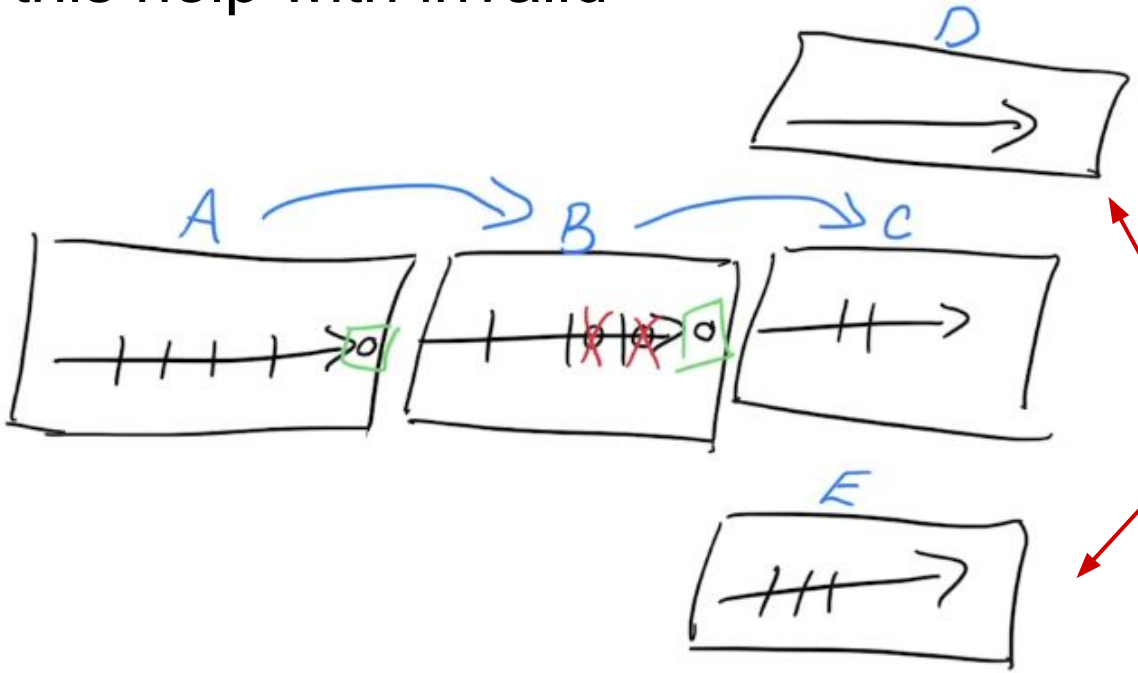


How does Access Control work?



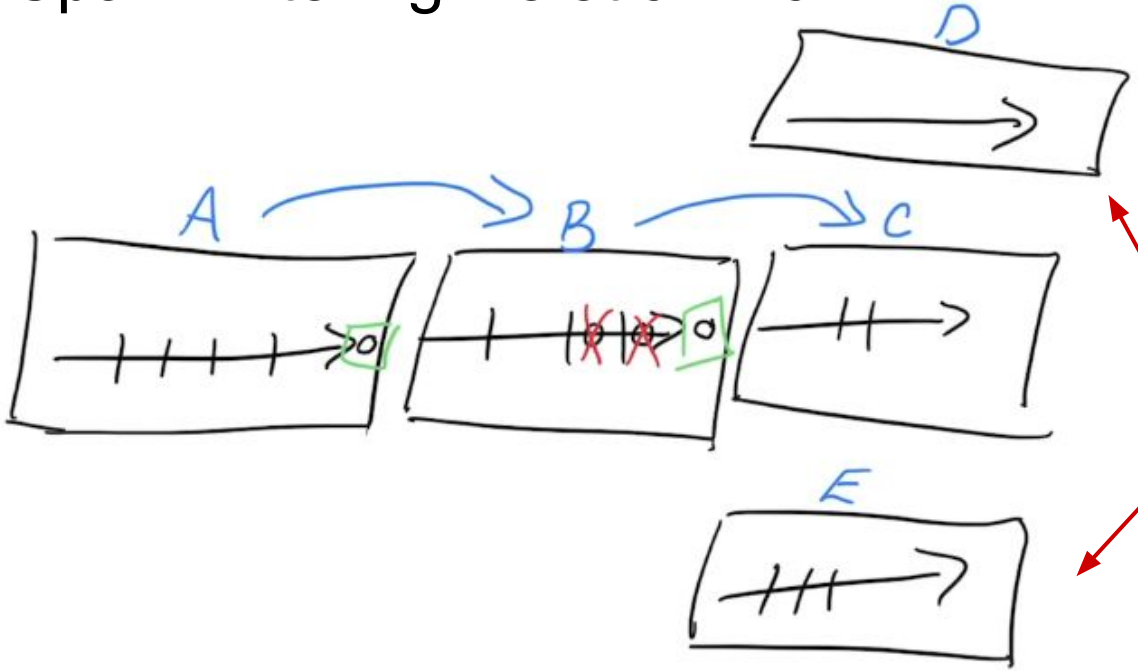
ID is used to read & write messages in a given box + ID is derived from MLS key schedule = Only group members allowed

Why does this help with invalid Commits?



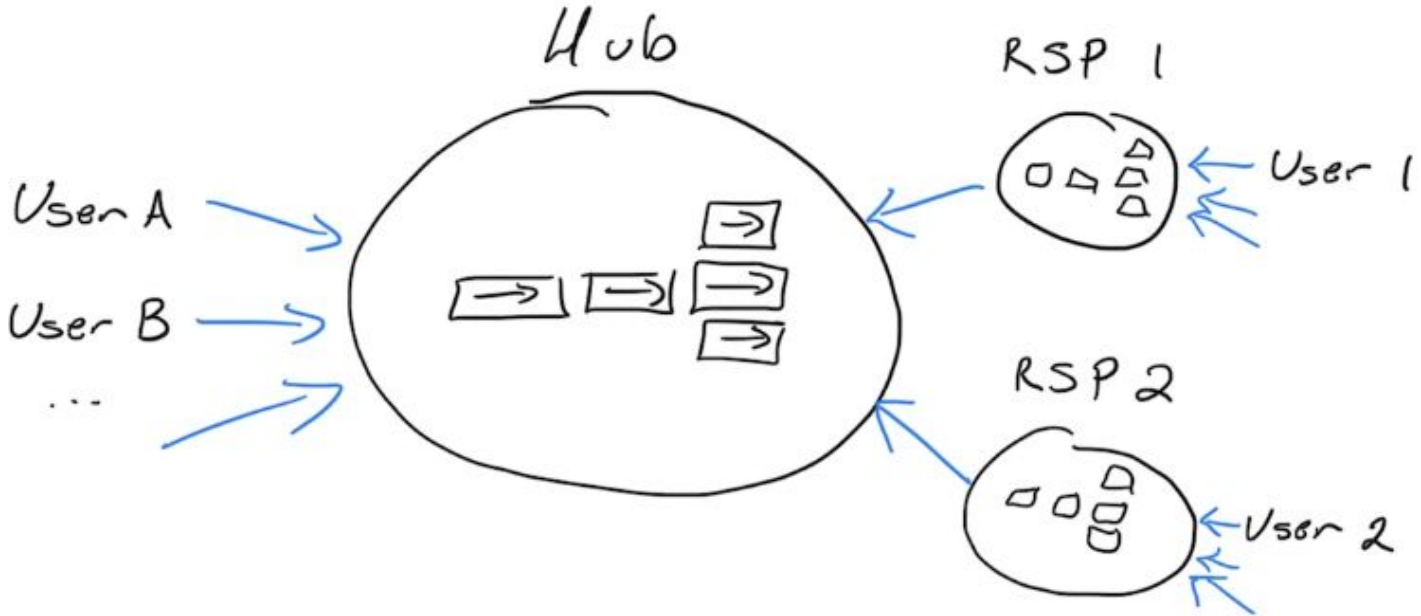
Allowed to exist! But never accessed

How does Spam ~~Filtering~~ Deletion work?



Forks which were created and never accessed by anyone else: 👎

Why does this help Membership Privacy?



When group membership is encrypted, hub provider has NO visibility into other Service Providers' users

I don’t want to do a lot of roundtrips

If a Service Provider is confident that a series of epochs will be requested together, they can be provided proactively

(Hash box IDs that we’re not sure the requester knows)

“I would like to read messages in box A”