You Read It Out of Context: Causality in Secure Messaging

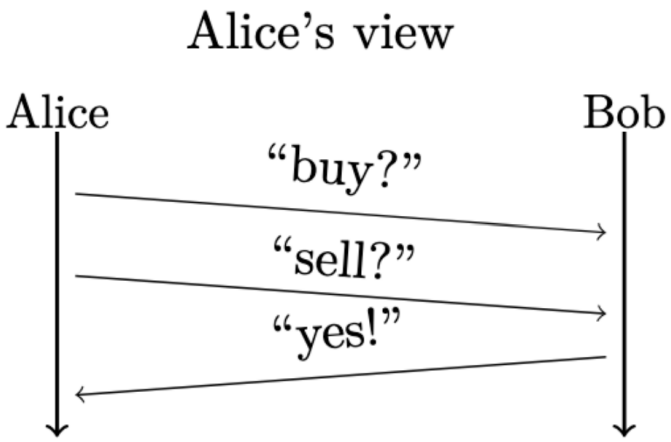
Igors Stepanovs

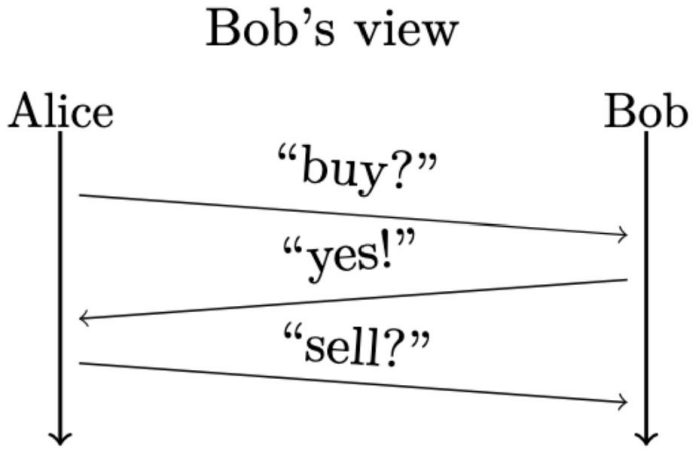
*ongoing joint work with **Joseph Jaeger** and **Akshaya Kumar***

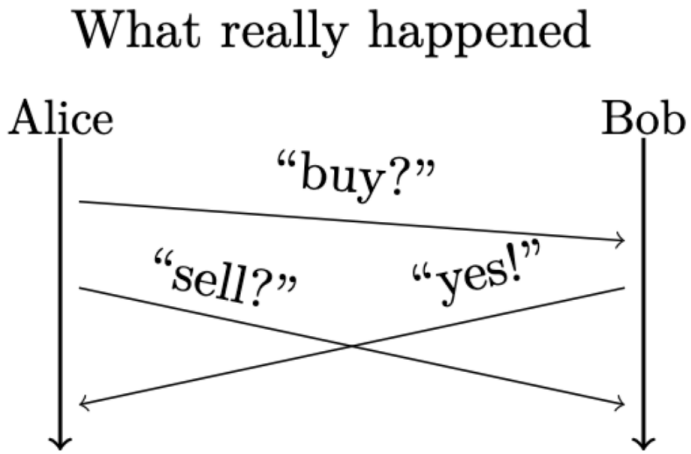
Georgia Tech

Estonian-Latvian Theory Days 2026, Tartu

The motivating example

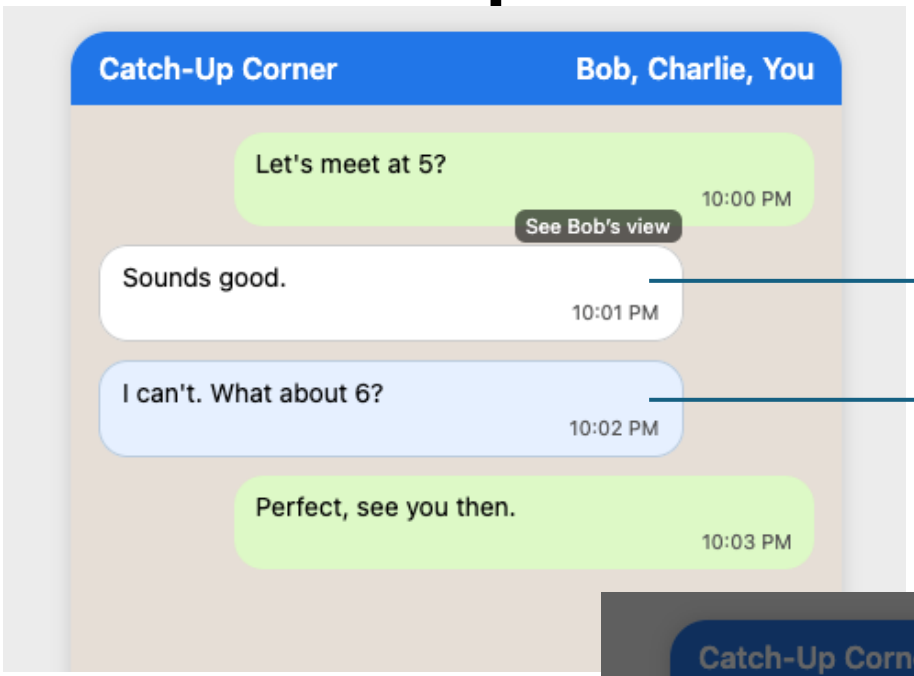






Cryptographically, nothing is broken.

Catch-Up Corner: our mock example of user-facing UI

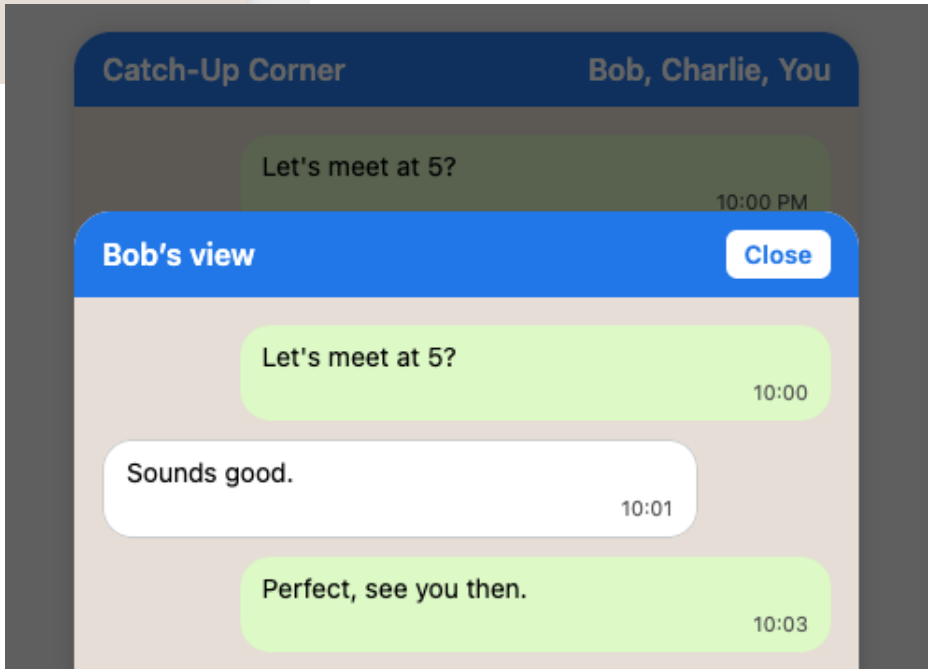


I expect: 6pm.

Bob

Charlie

Bob expects: 5pm.



Charlie expects: 6pm.

What current E2EE actually guarantees: overview

Confidentiality (no one else can read your message)	**YES**
Authenticity (you know who wrote it)	**YES**
Each sender's messages displayed in their original order	**YES**
Different senders' messages line up consistently for everyone	**NO**

Signal engineers spotted this in a 2014 blog post and parked it as 'a UI problem.' Still no deployed fix today.

What current E2EE actually guarantees: Signal's blog post

This allows each client to build a graph of message parents and children. One could imagine visualizing this like an old-school email client visualizes conversation threads, which would make transcript inconsistencies and "reply intent" clear. That way if Bob sends a message to everyone that says "Who wants to kill the president?" except Alice, to whom Bob instead sends "Who wants to get some ice cream?," when Alice responds "I do, how about this afternoon!" – it'd be clear to everyone in the conversation that Alice is responding to something they didn't see.

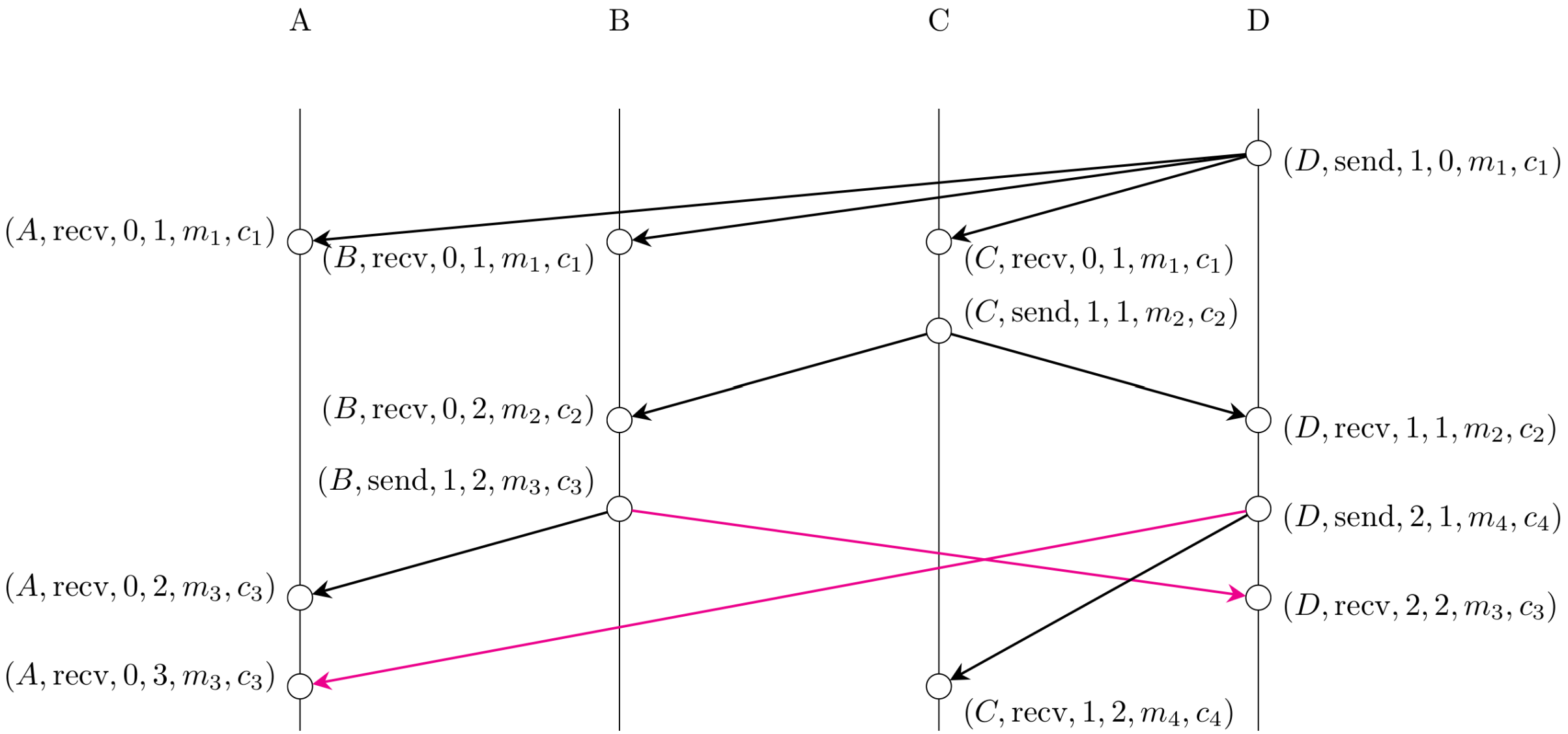
Unfortunately, that type of "threaded" UI isn't really possible or desirable on mobile devices, and even desktop communication apps are moving away from those types of threaded views. We want TextSecure conversations to remain

https://signal.org/blog/private-groups/

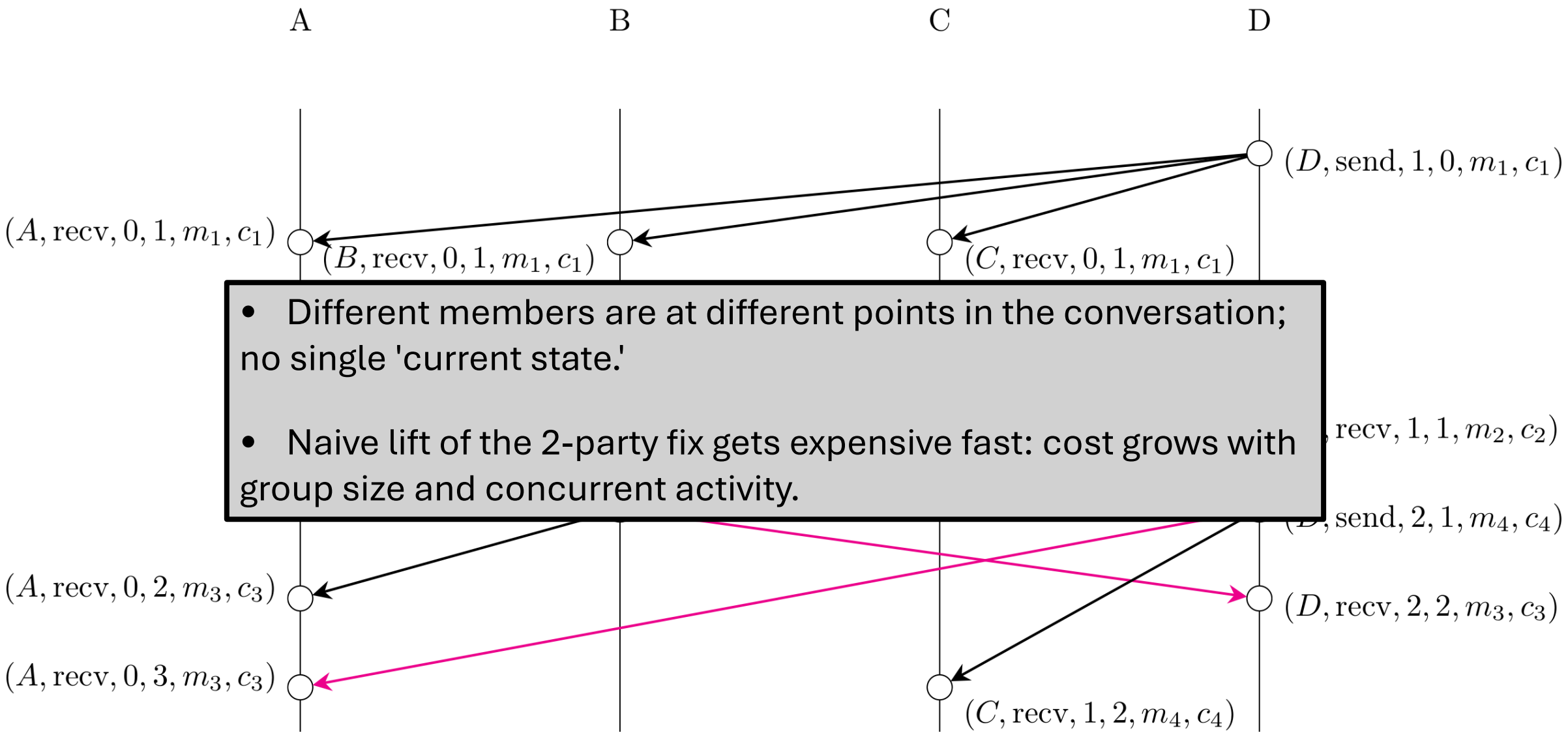
Two-party case is already non-trivial

- The same problem shows up in two-party Signal.
- Chen-Fischlin (2024) for the 2-party setting:
  - formalized causality preservation
  - proved Signal fails it even though no message can be forged or tampered with
  - gave a generic fix for 2 parties (not adopted by Signal yet)
- **Even two parties needed a research paper.**

Groups are qualitatively harder, not just bigger



Groups are qualitatively harder, not just bigger



- Different members are at different points in the conversation; no single 'current state.'
- Naive lift of the 2-party fix gets expensive fast: cost grows with group size and concurrent activity.

What distributed systems already proved: the fork-consistency blocking bound

- Why not just make everyone in the group see the same conversation?
- Mazieres-Shasha (2002): clients of an untrusted server can DETECT inconsistent views.
- Cachin-Shelat-Shraer (2007): you cannot PREVENT such forks without making clients wait on each other.
- For asynchronous messaging, that wait is unacceptable.
- We give up on agreement, settle for DETECTION. That is what 'awareness' means.

Related vocabulary you may know: Lamport's "happened-before" (1978); vector clocks (Fidge / Mattern, 1988); eventually-consistent databases (Google Docs); Byzantine broadcast; blockchains (consensus gives agreement, at a cost we cannot pay).

What deployed messengers actually do, and what is about to change

- Signal-family (Signal, WhatsApp, Messenger): no cross-sender ordering today.
- MLS (Messaging Layer Security), the new group encryption standard: detects forks in group-state changes (member changes, key rotations) even against a malicious server, but nothing for chat messages.
- **MLS-encrypted messaging is rolling out for billions of phones. Not yet shipped, but imminent.**
  - RCS: the SMS-replacement standard for Android and iPhone. Adopting MLS encryption now.
  - MIMI (More Instant Messaging Interoperability): cross-provider standard, driven by EU mandate.

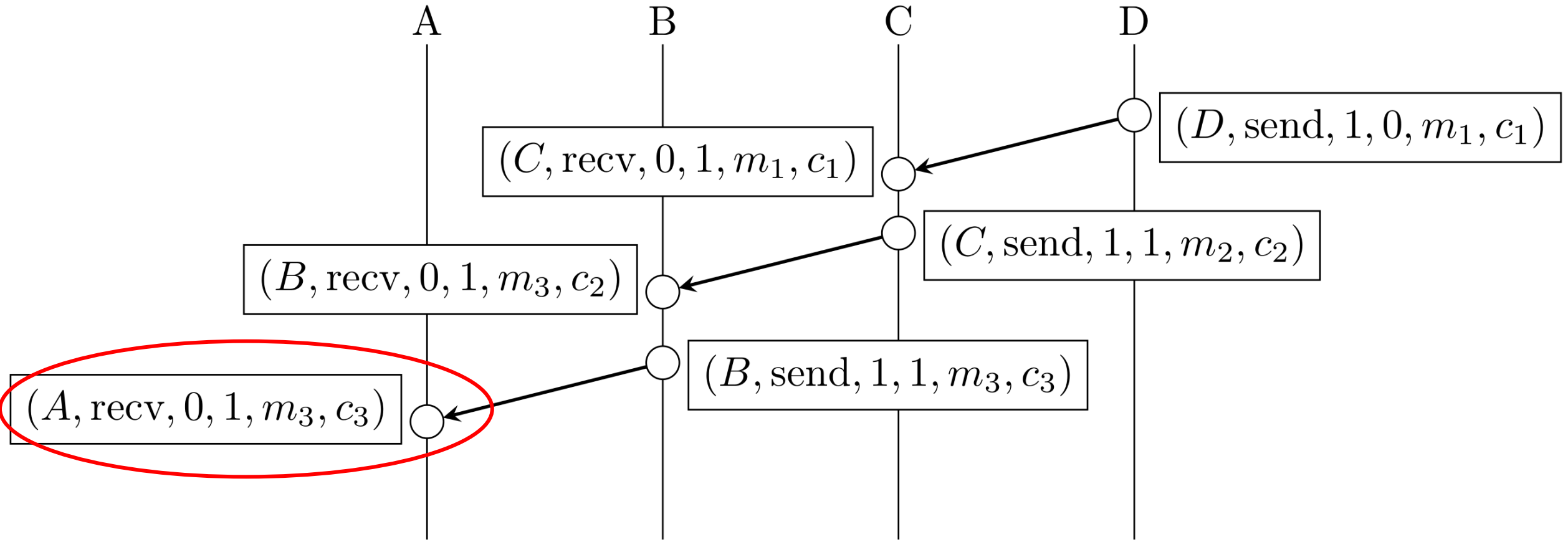
No deployed messenger solves this for groups today.

Today: the high-level perspective

- We're formalizing **causality**, correctness, and integrity for group chats.
- Approach:
    - define what causality means at the unencrypted layer first
    - then lift to encrypted channels using standard crypto primitives.
- Deployment: minimally extend existing protocols; reuse their crypto layer.
- Signal contacted: open to suggestions, will consider implementing once we are done.

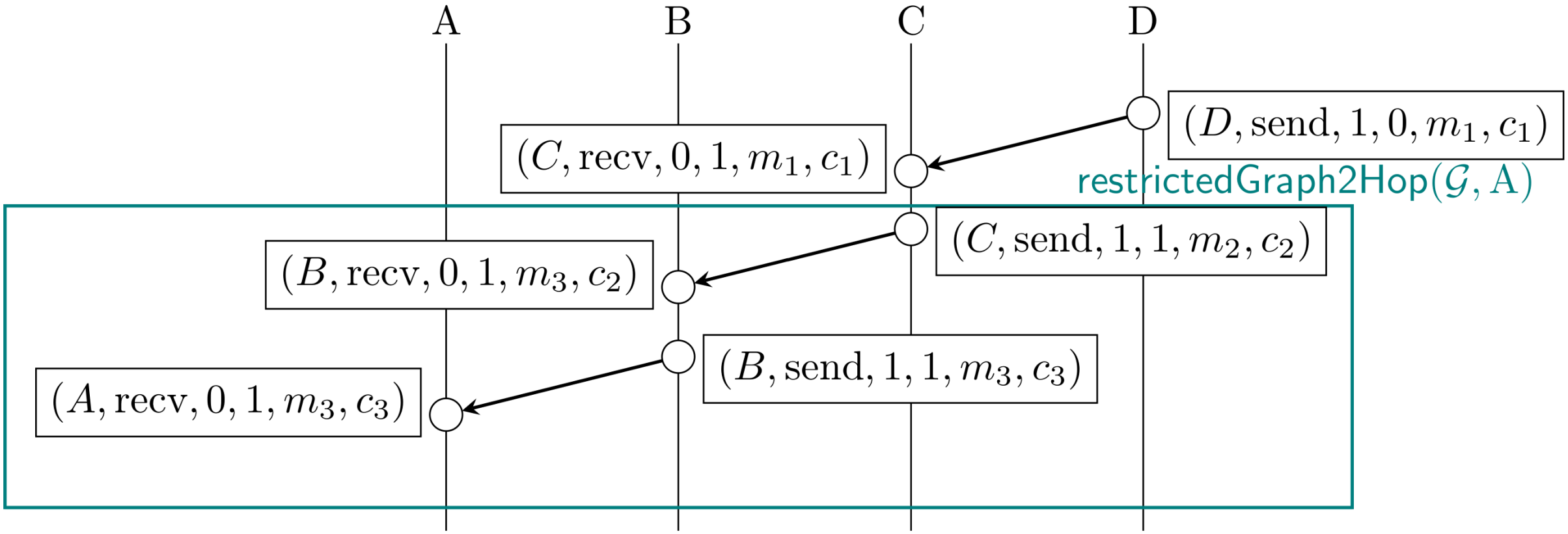
Today: high-level perspective only. No formal definitions.

Framework: a hierarchy of causality notions



Q: How much causality information A should receive from B?

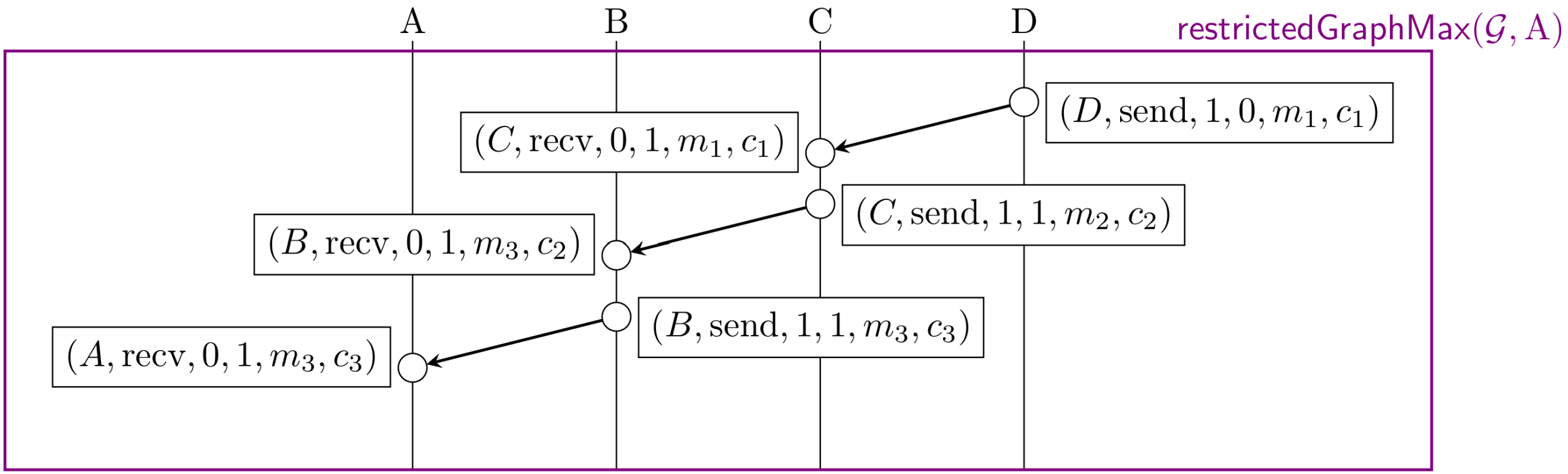
Framework: a hierarchy of causality notions



Q: How much causality information A should receive from B?

A: Only B's first-hand experience!

Framework: a hierarchy of causality notions



Q: How much causality information A should receive from B?

A: All of it!

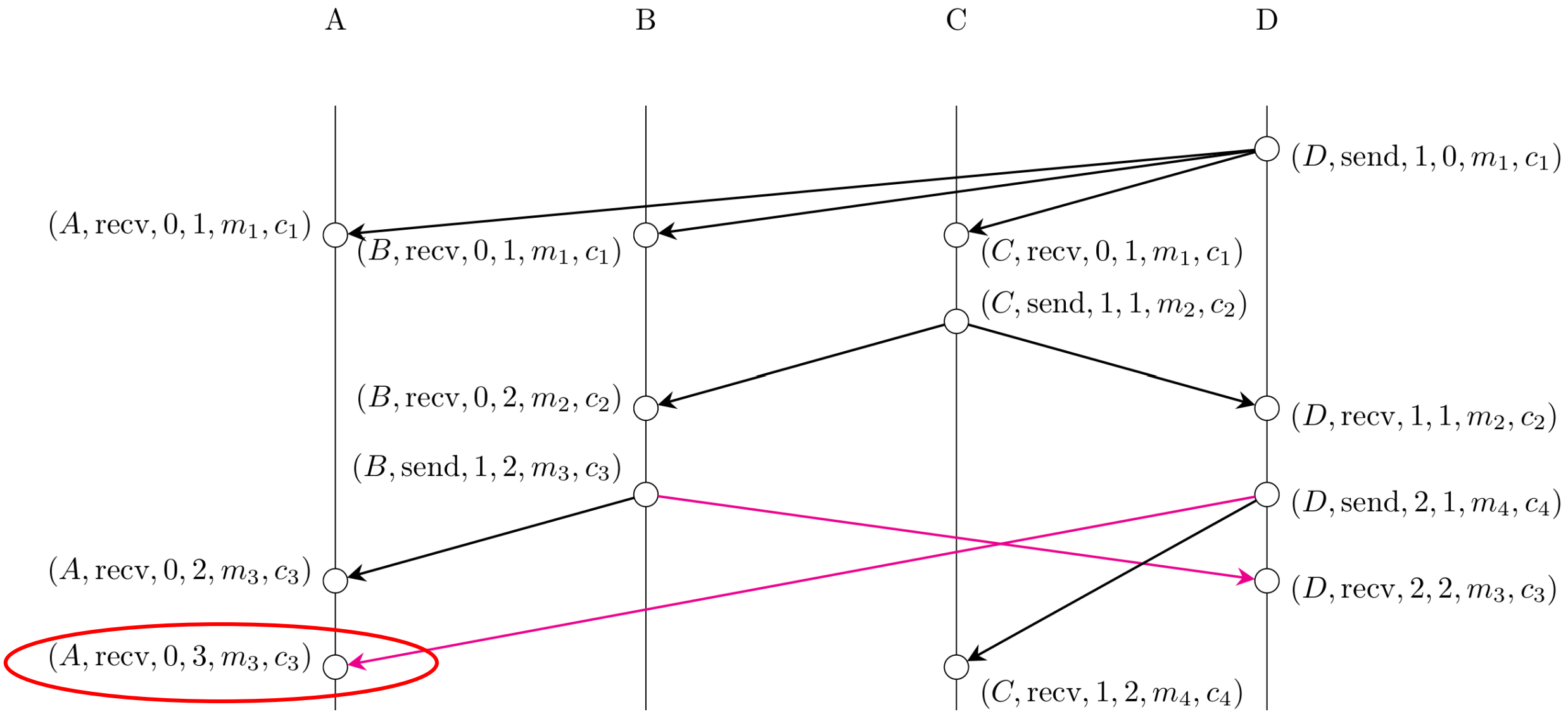
Stronger guarantees cost more per-message bandwidth.

Our proposed direction

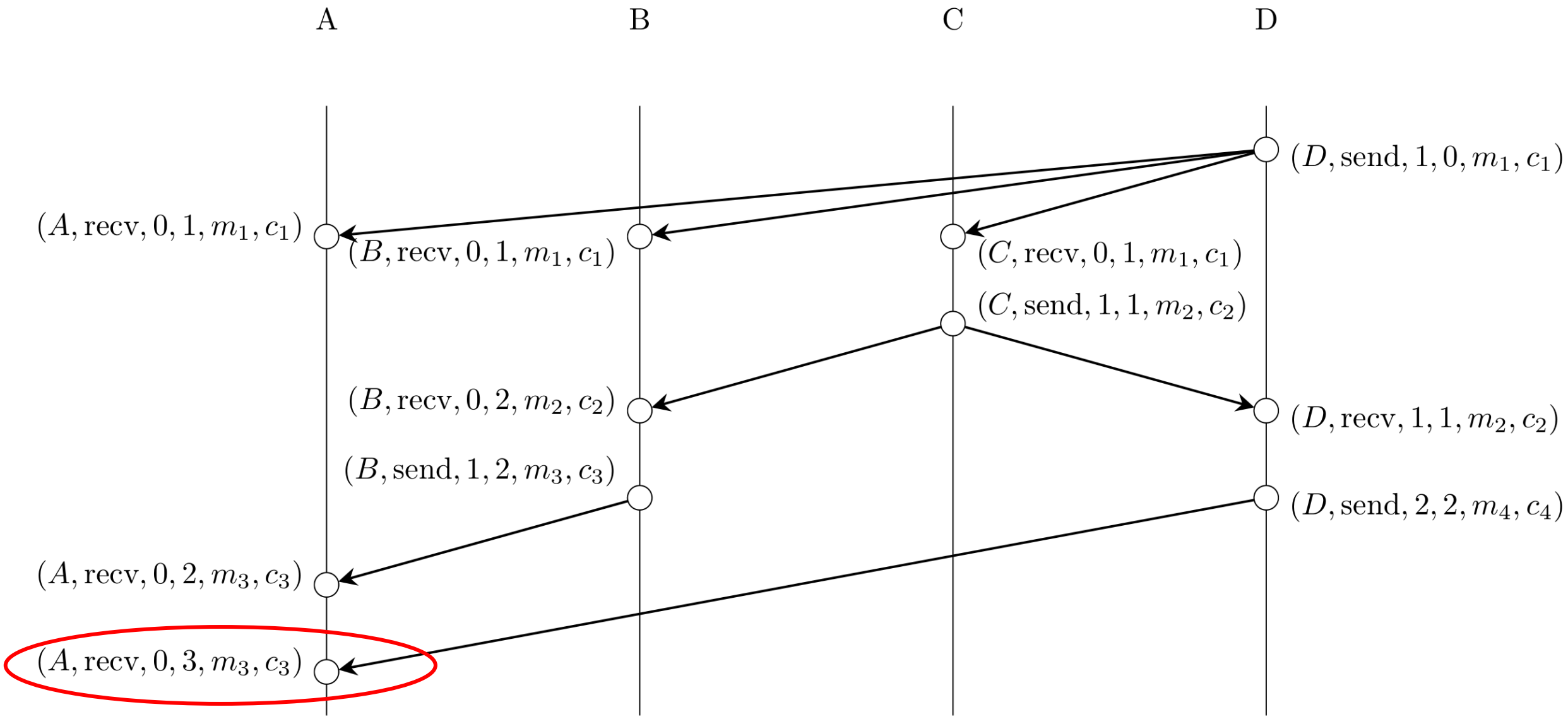
- Detect divergence.
    - Each user gets their own awareness, based on what they have observed.
- Causality data piggybacks on existing messages; no new round-trips.

Our goal: "the best causality awareness extractable from the message flow already in motion."

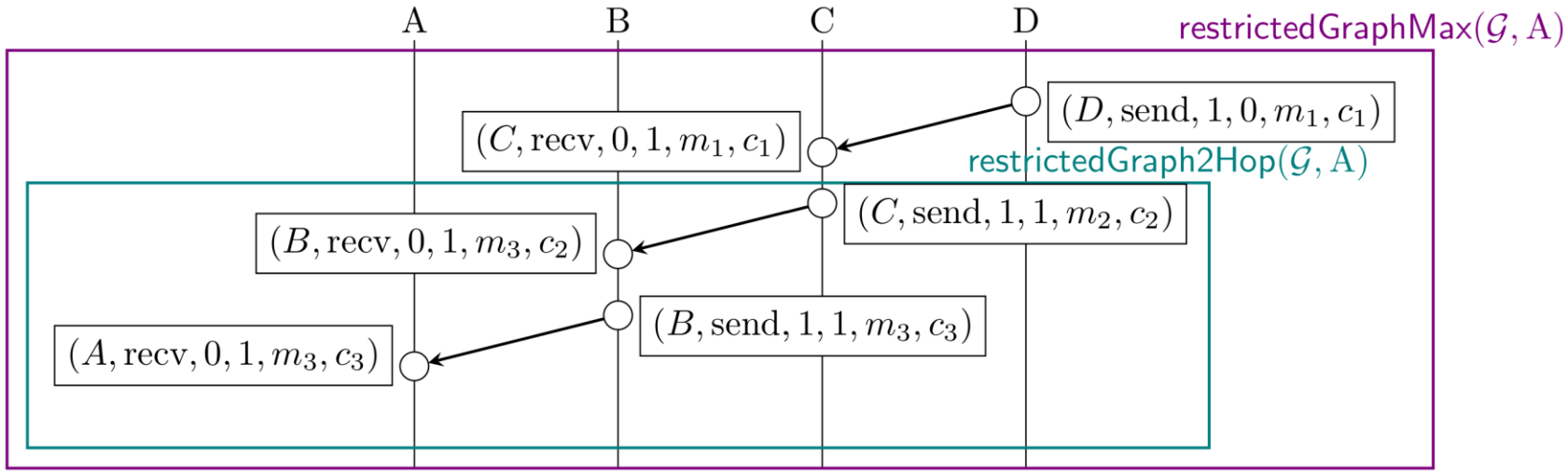
The “best possible” awareness: global transcript



The “best possible” awareness: A’s local transcript



Verifying causality on demand



- Idea: separate a small in-band commitment from the heavier off-band data.
- Per-message overhead: just a small cryptographic commitment (e.g. one hash).
- Heavier causality data: encrypted, stored on an untrusted server.
- Recipients fetch and verify on demand. Automatic triggers (new device, anomaly, periodic checkpoint), not user-click.

Automatic verification, UI as surfacing layer.

Open questions

- Is 'awareness' the right relaxation, or is there a better intermediate target?
- What is the right UX? When should the user see a warning?
- Insider attacks are out of scope. Can we ship without addressing them?
- What can we borrow from blockchain, distributed databases, or fork-detection literature?

Questions / Connections / What are we missing?

Igors Stepanovs

https://igors.org/