Local First, Distributed Authentication

For a secure, P2P, distributed, & data self-sovereign world.

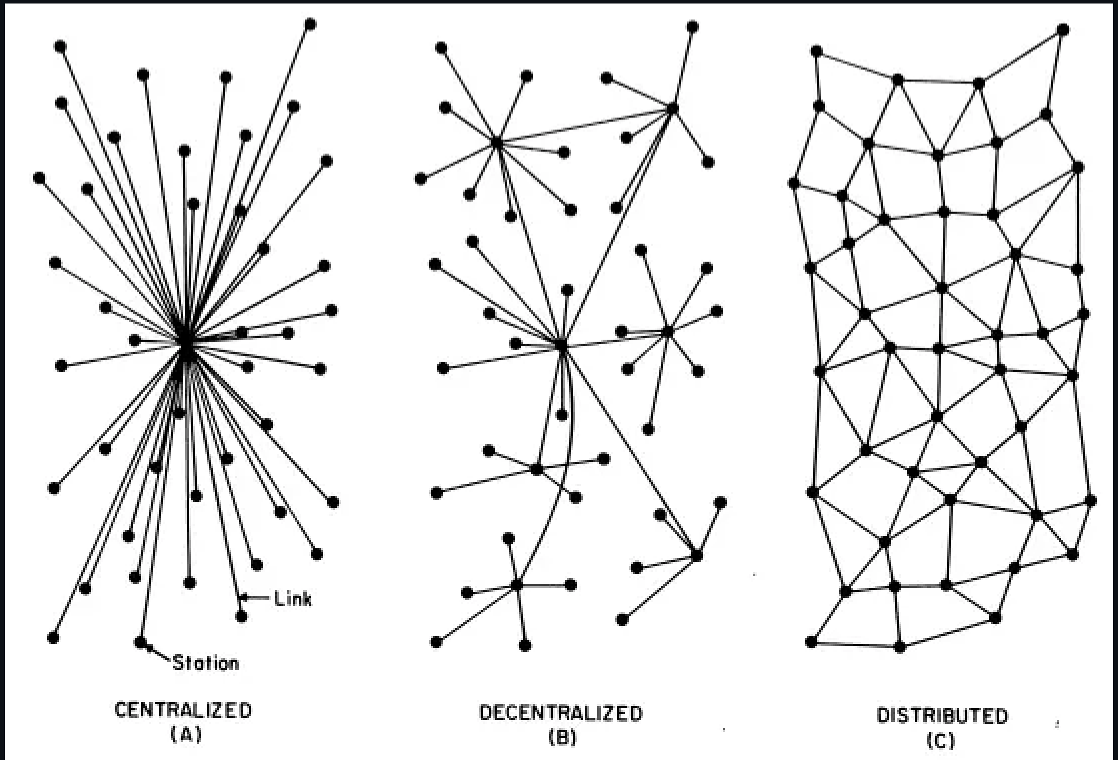
AZ Rust Meetup 11/06/2025

What is local first?

1. No spinners: your work at your fingertips
2. Your work is not trapped on one device
3. The network is optional
4. Seamless collaboration with your colleagues
5. The Long Now
6. Security and privacy by default
7. You retain ultimate ownership and control

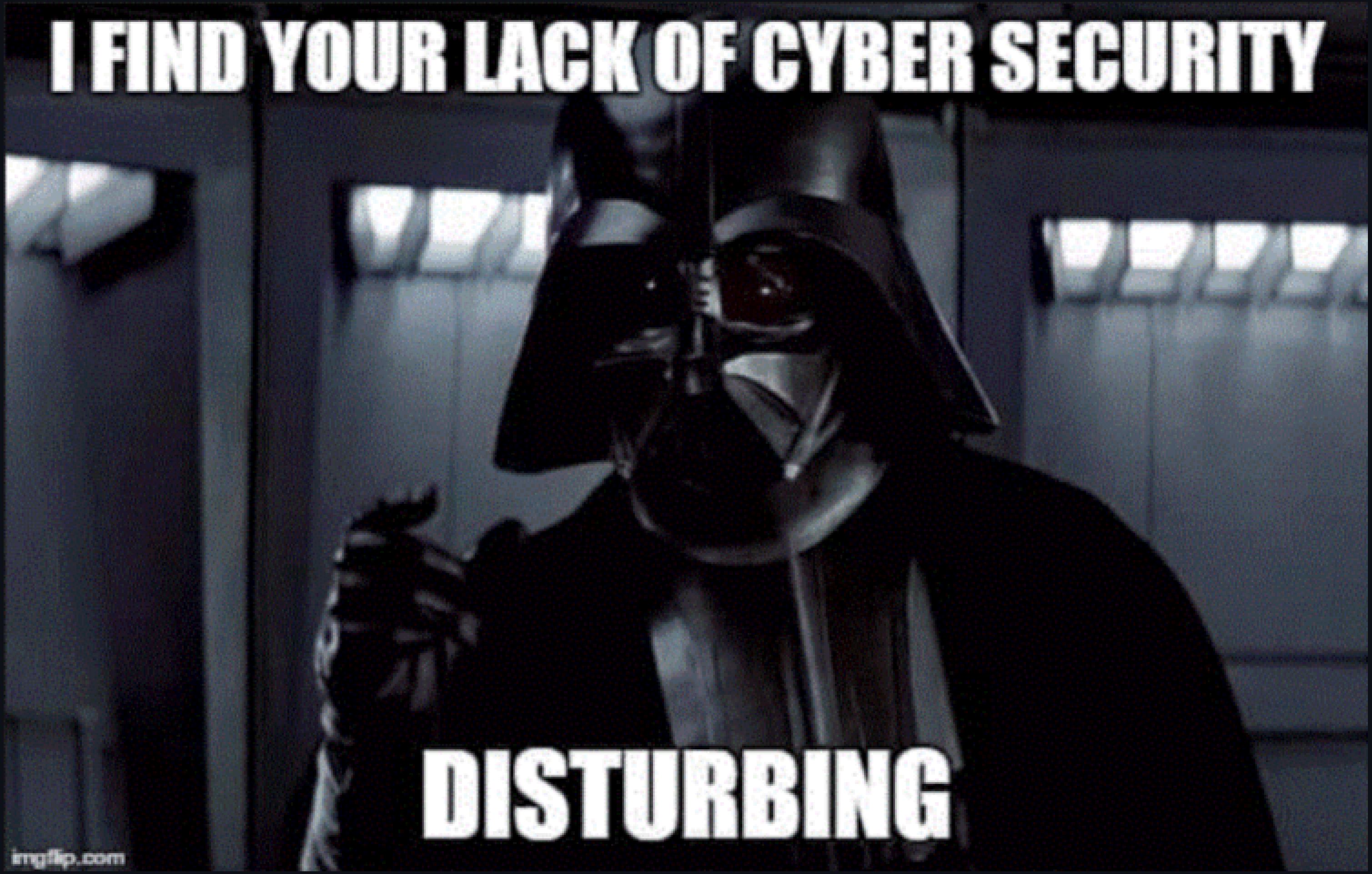
Ink & Switch first laid all of this out in an essay. (Highly recommend taking a look!)

Distributed networking

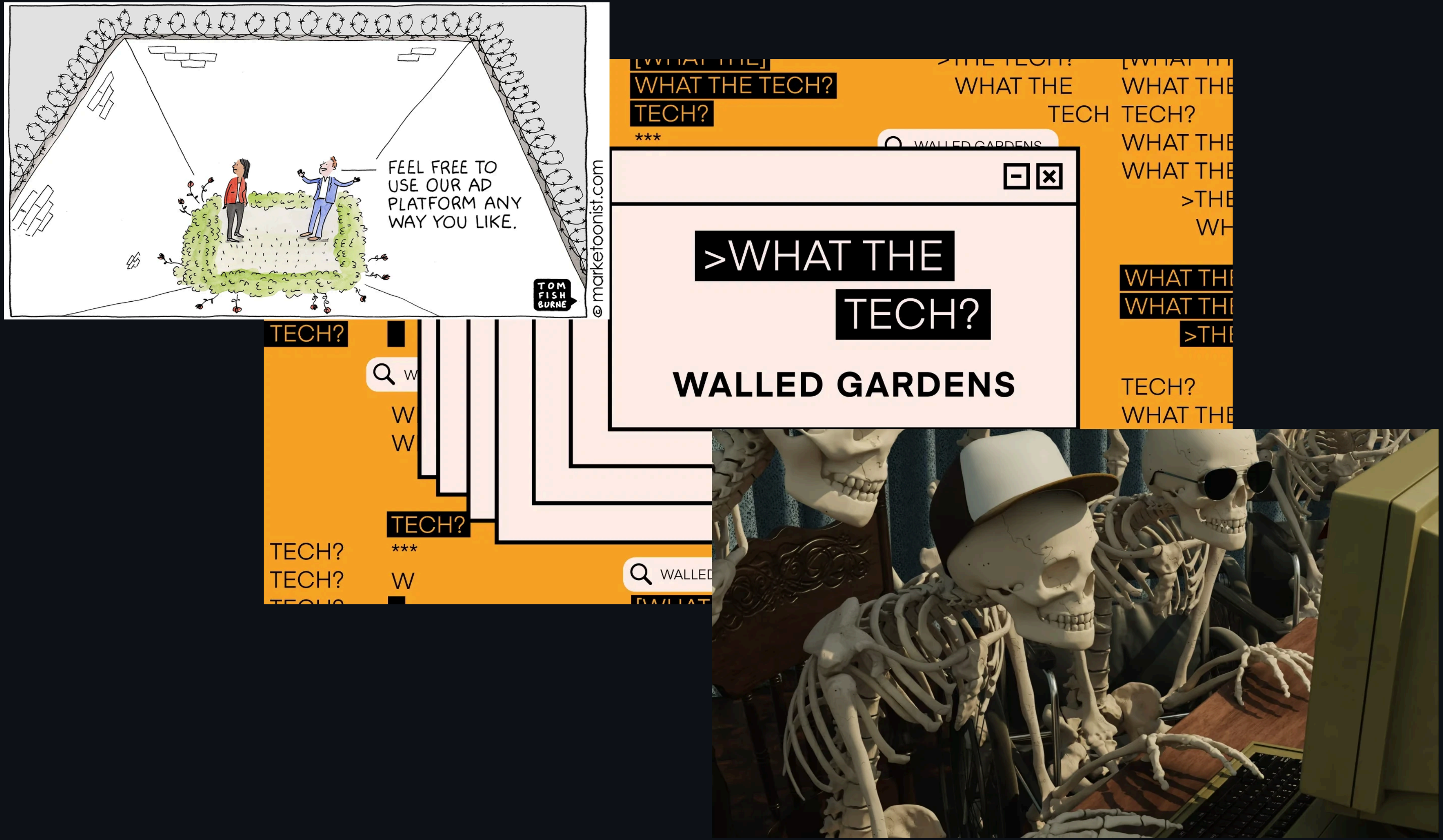


On Distributed Communications

The embodiment of online security



The forced solution



Can a distributed, P2P solution provide an alternative?

Benefits

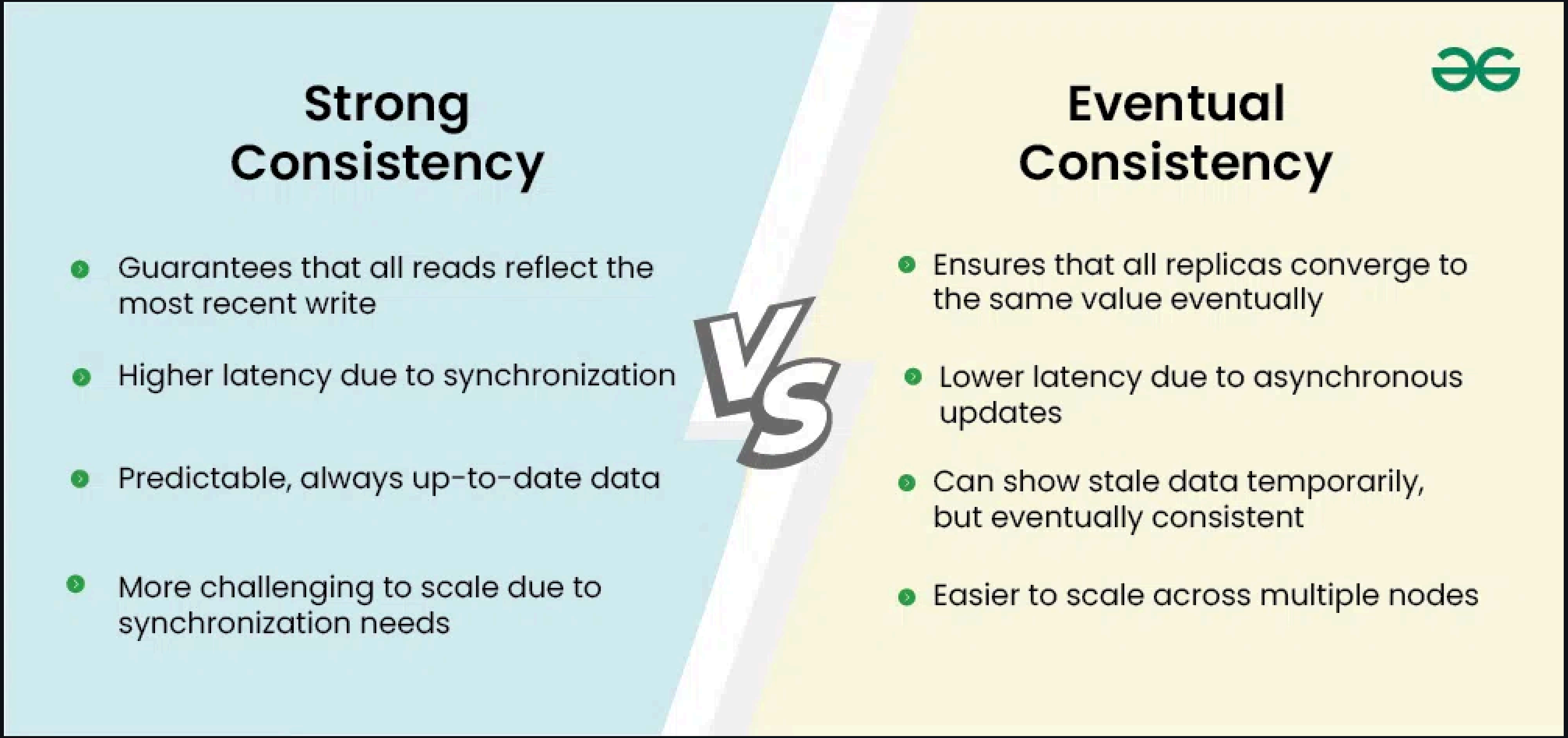
- Compliance with Local First principles
- Data and algorithmic sovereignty
- Resiliency
- Privacy
- Security as a foundation
- ...

Challenges

- No central gatekeeper
  - login
  - identity
  - collation
  - db management
- No way to provide strong consistency

Strong vs. Eventual Consistency

Perspective Shift: Causal order/alignment



Strong Consistency

- Guarantees that all reads reflect the most recent write
- Higher latency due to synchronization
- Predictable, always up-to-date data
- More challenging to scale due to synchronization needs

VS

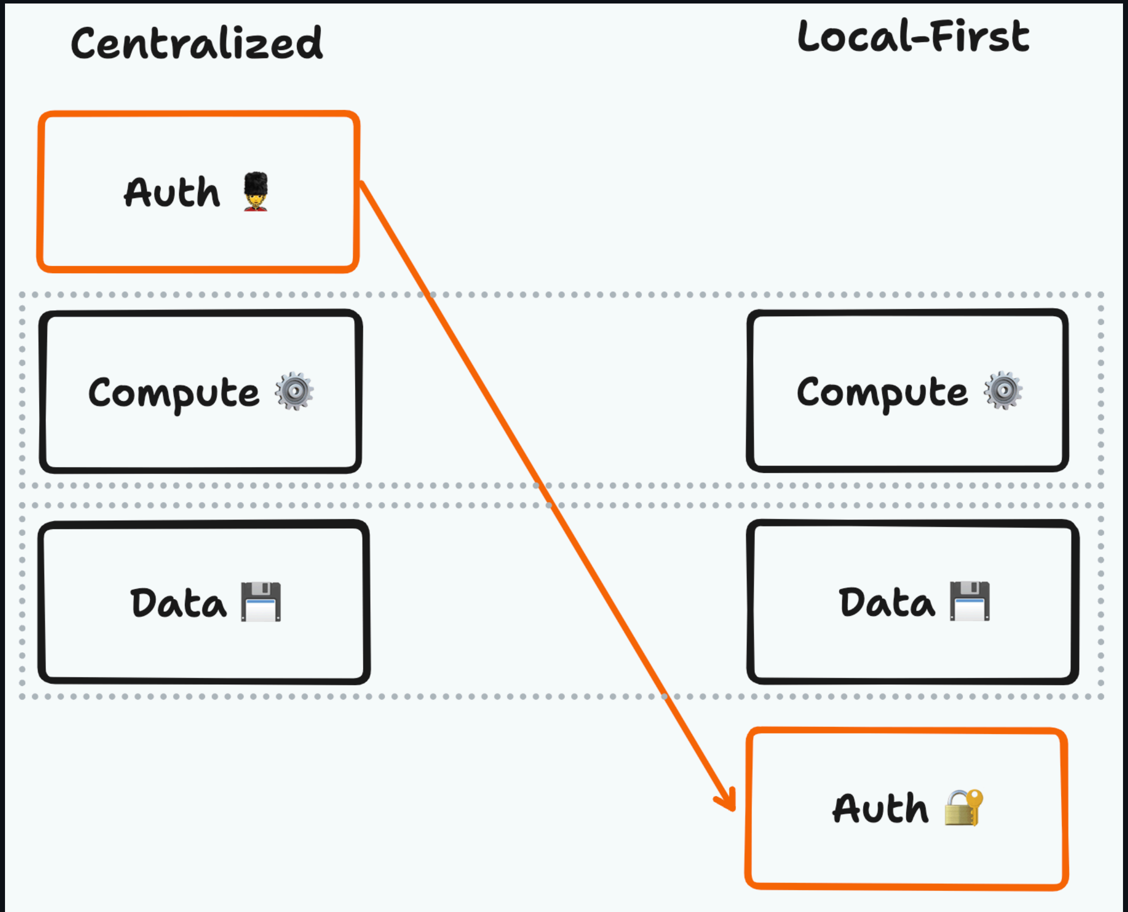
Eventual Consistency

- Ensures that all replicas converge to the same value eventually
- Lower latency due to asynchronous updates
- Can show stale data temporarily, but eventually consistent
- Easier to scale across multiple nodes

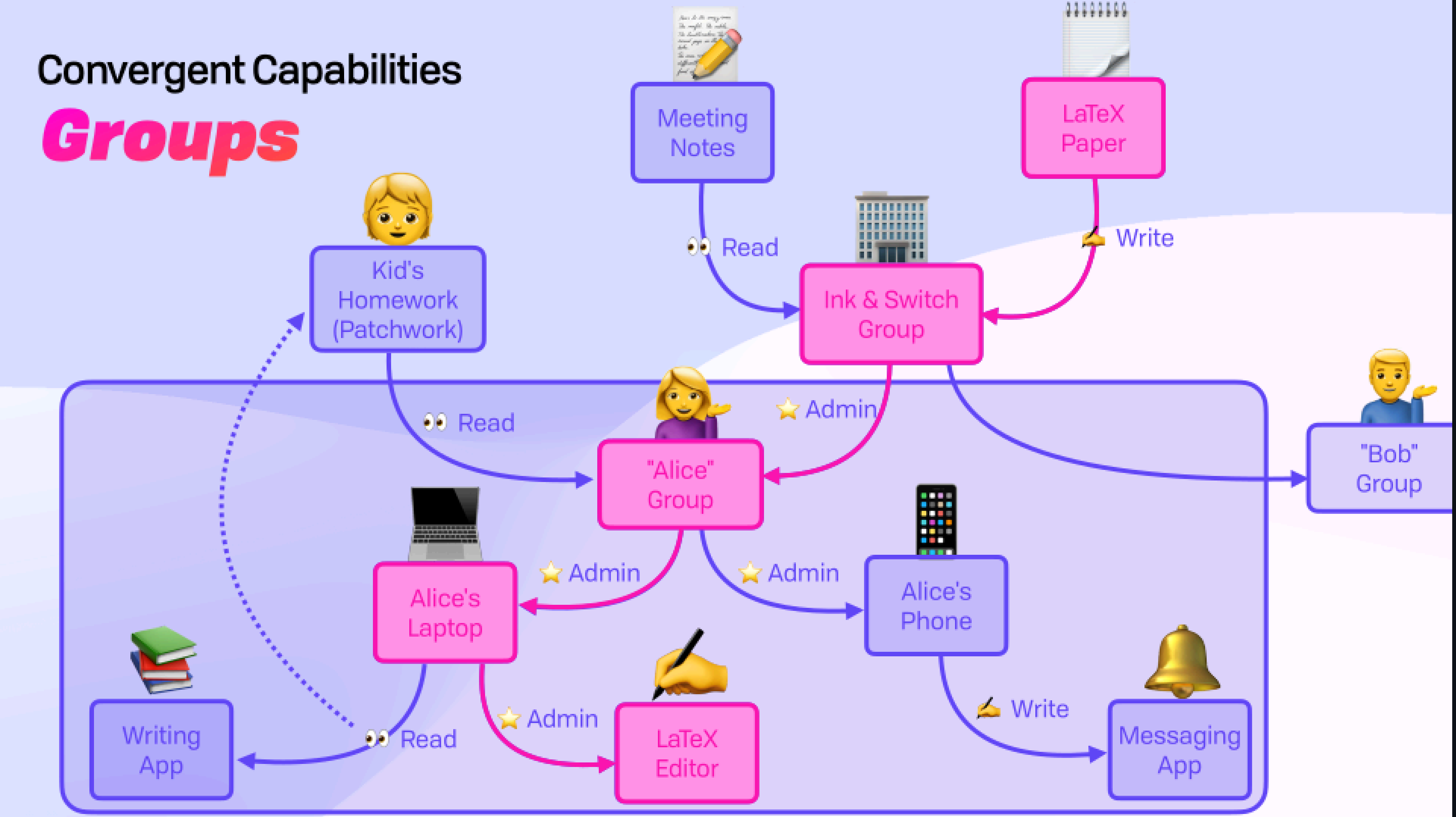
Causal order

- Time clock drift precludes ordering by timestamp
- Order can be achieved by hashing what has been seen before with each update
- Concurrent updates are possible, so must be commutative (CRDTs can help here)

Moving Authentication and Security management to the bottom of the stack



Convergent Capabilities (no cloud auth!)



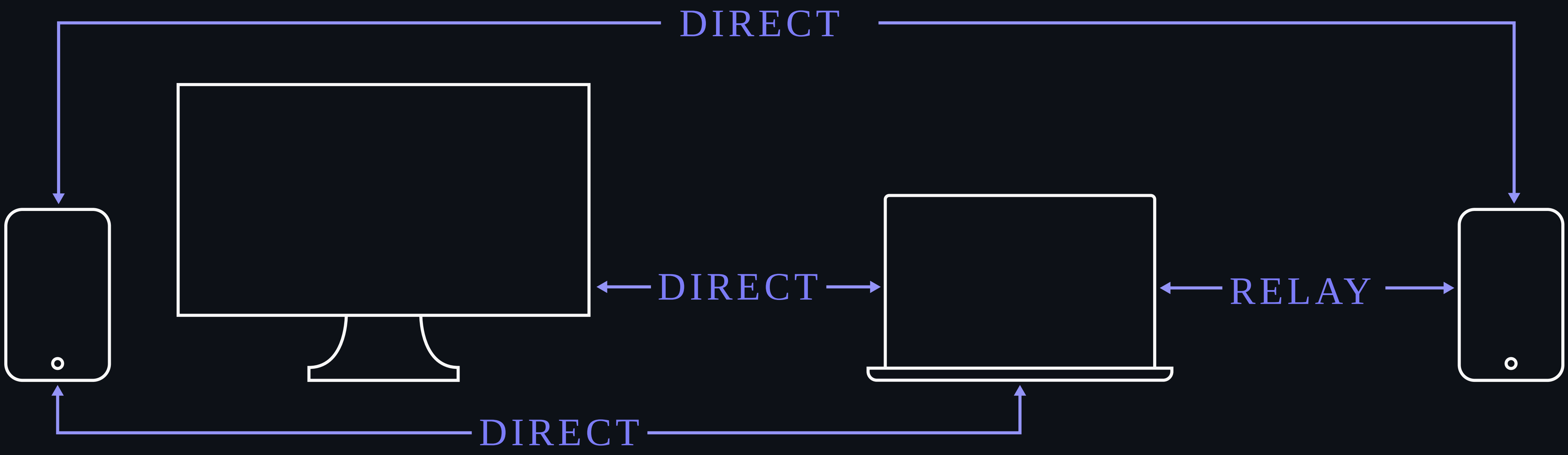
CRDTS and other Data Storage Paradigms

- "Conflict Free Replicated Data Types"
  - The application can update any replica independently, concurrently and without coordinating with other replicas.
  - An algorithm (itself part of the data type) automatically resolves any inconsistencies that might occur.
  - Although replicas may have different states at any particular point in time, they are guaranteed to eventually converge.

Rust Ecosystem tools

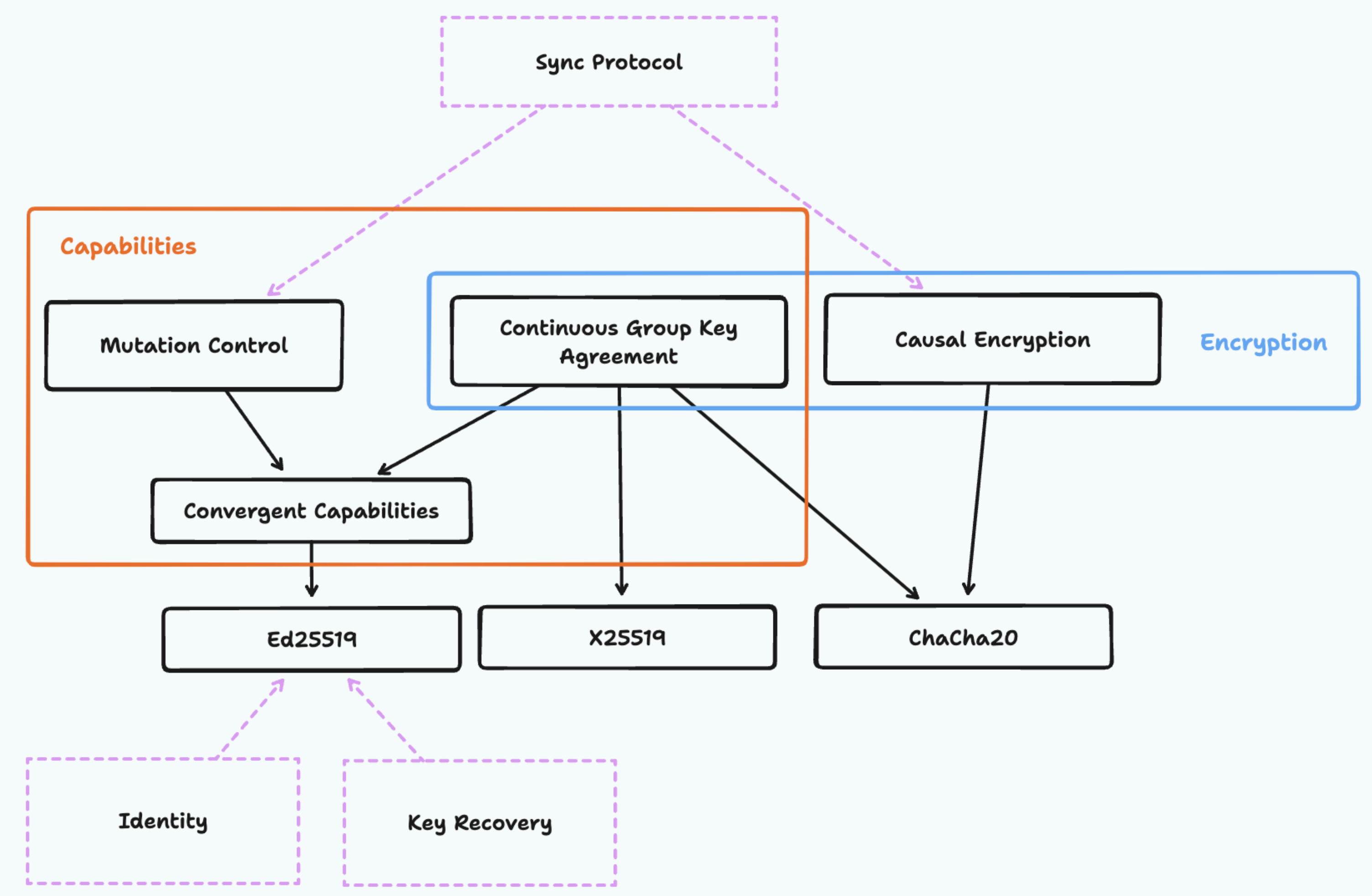
- Iroh
- Automerge, Loro, etc.
- Keyhive/Beelay - (Local First Conf 2025 talk by Brooklyn Zelenka)

Iroh



- P2P networking over Quic
- High hole punch success rate using relay and successful NAT traversal
- IPV4 & IPV6

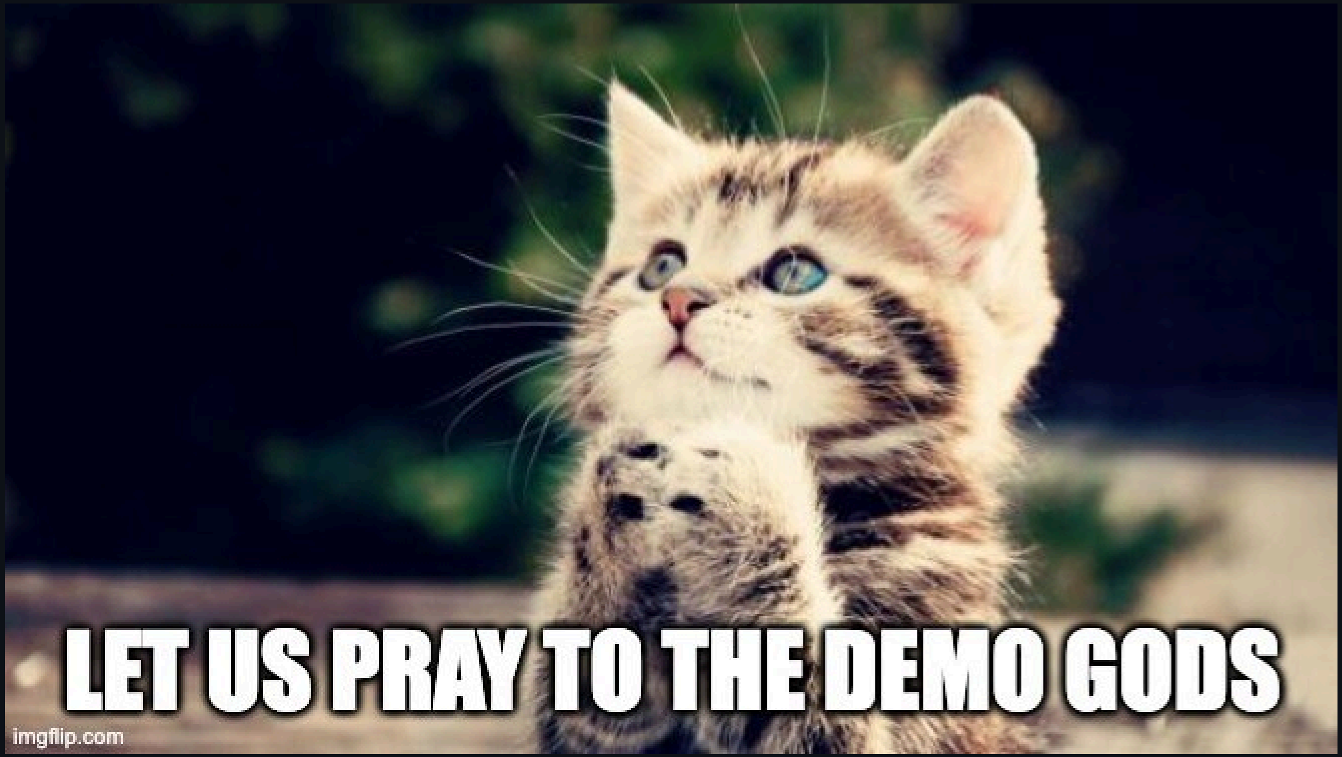
Keyhive/Beelay



Putting it all together for an E2EE and encrypted at rest chat application

- Iroh custom protocol
- Beelay/Keyhive as an actor (primary target is wasm in browser, so it is not Send)
- Tauri
- Leptos

Demo!



This is just the beginning

- Distributed heterogeneous model training (Nous Research)
- Weird
- Malai
- Ink & switch research projects
- Symplasma (website - WIP) (github)

Thank you!