ElectricSQL - Local-First SQL

App code talks to local database, & data syncs in background.

Systemic solution to state transfer

Looks like they rely on ElectricSQL

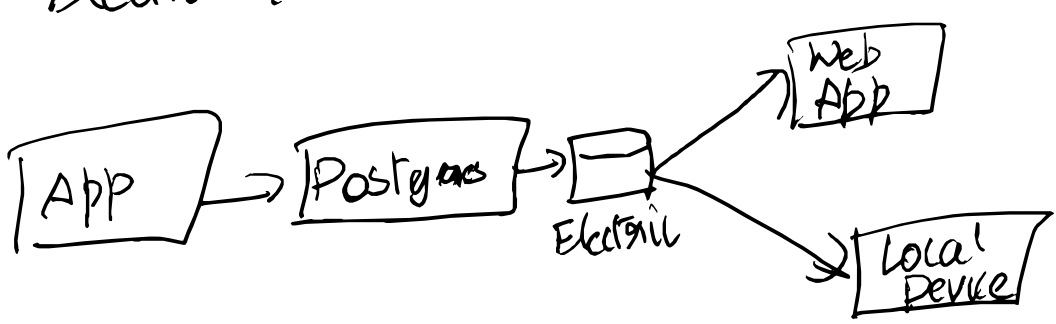
Why local first?

→ Resilience
→ Instantly reactive
→ Realtime multi-user
→ Conflict-free offline

ElectricSQL is a sync layer between local & cloud DB.

Build Active-Active Data model
Sync in background

Apply migrations to PG, stream to Electric, stream to users



Eventual consistency is used

standard atomic semantics

Uses Zod to generate type safe schema for client to use & Prisma ORM for TS

Built off postgres schema

Developing Local First

→ No trusted backend for auth & security.
- job queue
- logical replication
- Materialize

→ Realise auth & validation logic into DB rules
→ Row based security
→ Run auth in the sync service

→ Scalability

Protobuf for serialized data transfer

Network & storage

→ PGs is source of durability

→ Rebuildable cache in sync service

→ client

Webassembly build of sqlite can be used with app.

Sync done with CRDT's.

Shaped based replication → Replicate subset of PG data.

Server side: control auth
Client side: control write & conflict rules

Concurrent overlapping writes

Traditional causal consistency with CRDT's (TCC+)

Rich CRDT's & hybrid / mixed consistency.