Local-First apps

Concepts and examples

Kenneth Geisshirt

https://k.zigzak.net/

Three questions

- Can your application start without network connectivity?
- Can your users complete a task when connectivity is lost?
- Do you self-host your users' data?

Agenda

- The Local-First manifesto
- The CAP theorem
- OT and CRDT
- Two examples/demos

The Local-First manifesto

You own your data, in spite of the cloud published in 2019

- Initially: collaboration and ownership
- Also important: privacy
- Today: geopolitical situation (digital sovereignty)

The seven ideals

1. Fast - no network roundtrip
2. Data synchronization across multi devices
3. Accessible when offline
4. Seamless collaboration with other users
5. Retain ownership after operations have ceased
6. Security and privacy by default - end-to-end encryption
7. Full control of who and how data is accessed

The CAP theorem

For any distributed data store, you can only pick two out of three:

1. Consistency
2. Availability
3. Partition tolerance

What it means

- AP - work offline (eventually consistent)
- CP - Database with logical replication
- CA - Single-node database

Distributed data stores

- Operational Transformations (OT)
  - Convergence of operations
  - Device 1: $O_1(S)$
  - Device 2: $O_2(S)$
  - We need to find $O^t{}_1$ and $O^t{}_2$ such that $O^t{}_2(O_1(S) = O^t{}_1(O_2(S))$
- Conflict-free Replicated Data Type (CRDT)
  - Devices can update replica independently
  - Implement algorithm to resolve inconsistencies

Automerge

Documents are stored in *repositories*

```
const repo = new Repo({
  network: [network],
  storage: new NodeFSStorageAdapter(`.data/${name}`),
})

let handle = repo.create({ todos: [] })
```

Automerge - Implementing the commands

```
let localEdit = false
function edit(fn) {
  localEdit = true
  try {
    handle.change(fn)
  } finally {
    localEdit = false
  }
}

const commands = {
  add(title) {
    if (!title) return console.log("  usage: add <title>")
    edit((doc) => {
      doc.todos.push({ title, done: false })
    })
  },
  // ... more commands
```

#[RxDB - A database-like experience

RxDB - Replication

Server:

```
const server = await listen(PORT)
await startWebsocketServer({ database: db, server })
```

Client:

```
replicateWithWebsocketServer({
  collection: todos,
  url: `ws://localhost:${PORT}`,
  replicationIdentifier: `todos-ws-${PORT}`,
  live: true,
})
```

RxDB - Implementing the commands

```
const commands = {
  async add(title) {
    if (!title) return console.log("  usage: add <title>")
    await todos.insert({
      id: crypto.randomUUID(),
      title,
      done: false,
      createdAt: Date.now(),
    })
  },
  // ... more commands
```

Take-home message

- Can your application start without network connectivity?
- Can your users complete a task when connectivity is lost?
- Do you self-host your users' data?

If you answered **no** to any of the questions, you should consider a local-first architecture for your next project