Local-first software

Resilient and secure collaboration

MARTIN KLEPPMANN TU MUNICH

Bluesky: @martin.kleppmann.com

Mastodon: @martin@nondeterministic.computer

VolkswagenStiftung

COLLABORATIVE APPLICATIONS

Figma

Wide range of domain-specific collaboration software, e.g. for investigative journalism, medical records, data analysis, engineering/CAD, ...

CLOUD SOFTWARE

e.g. Google

amazon web services

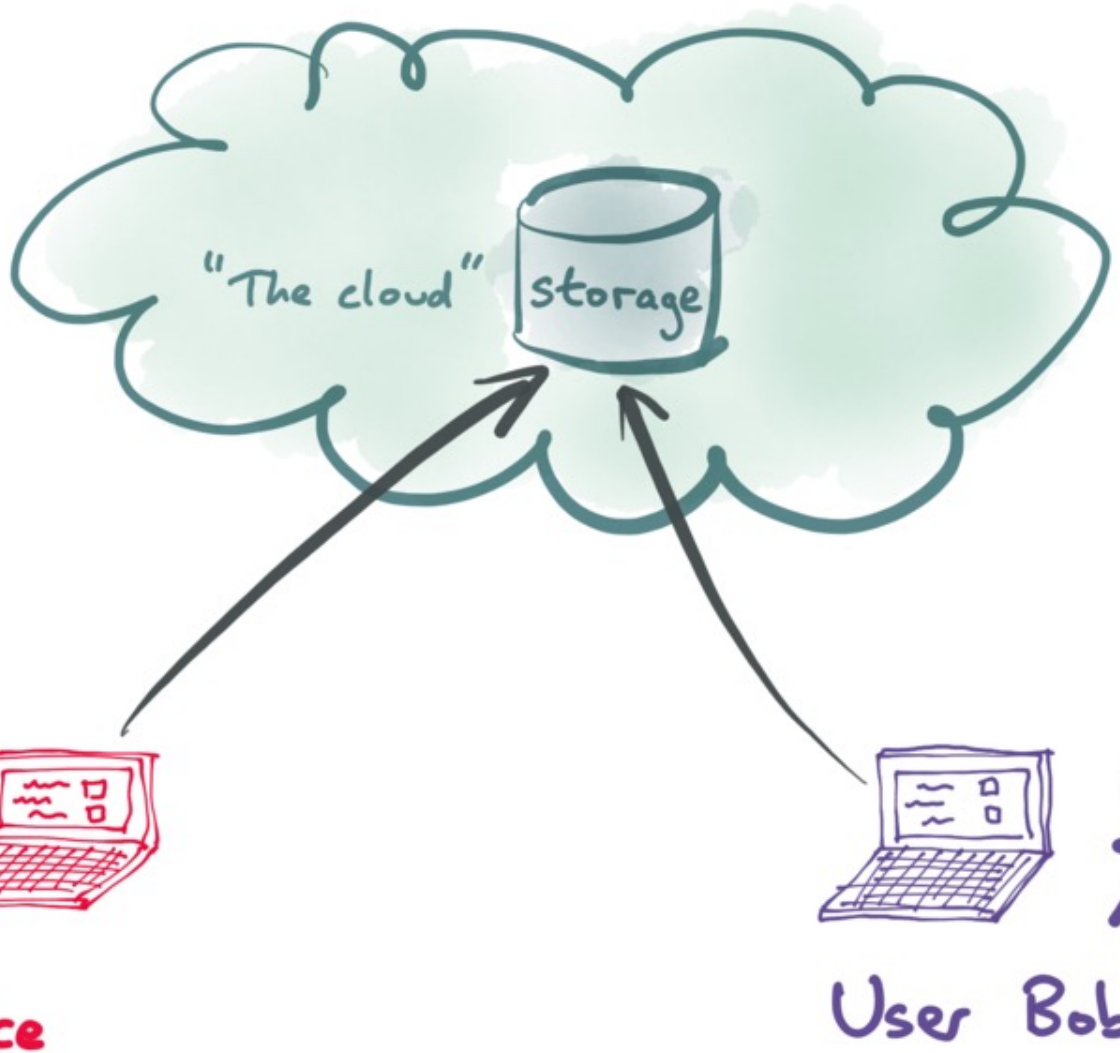
User Alice

User Bob

CLOUD SOFTWARE

e.g. Google

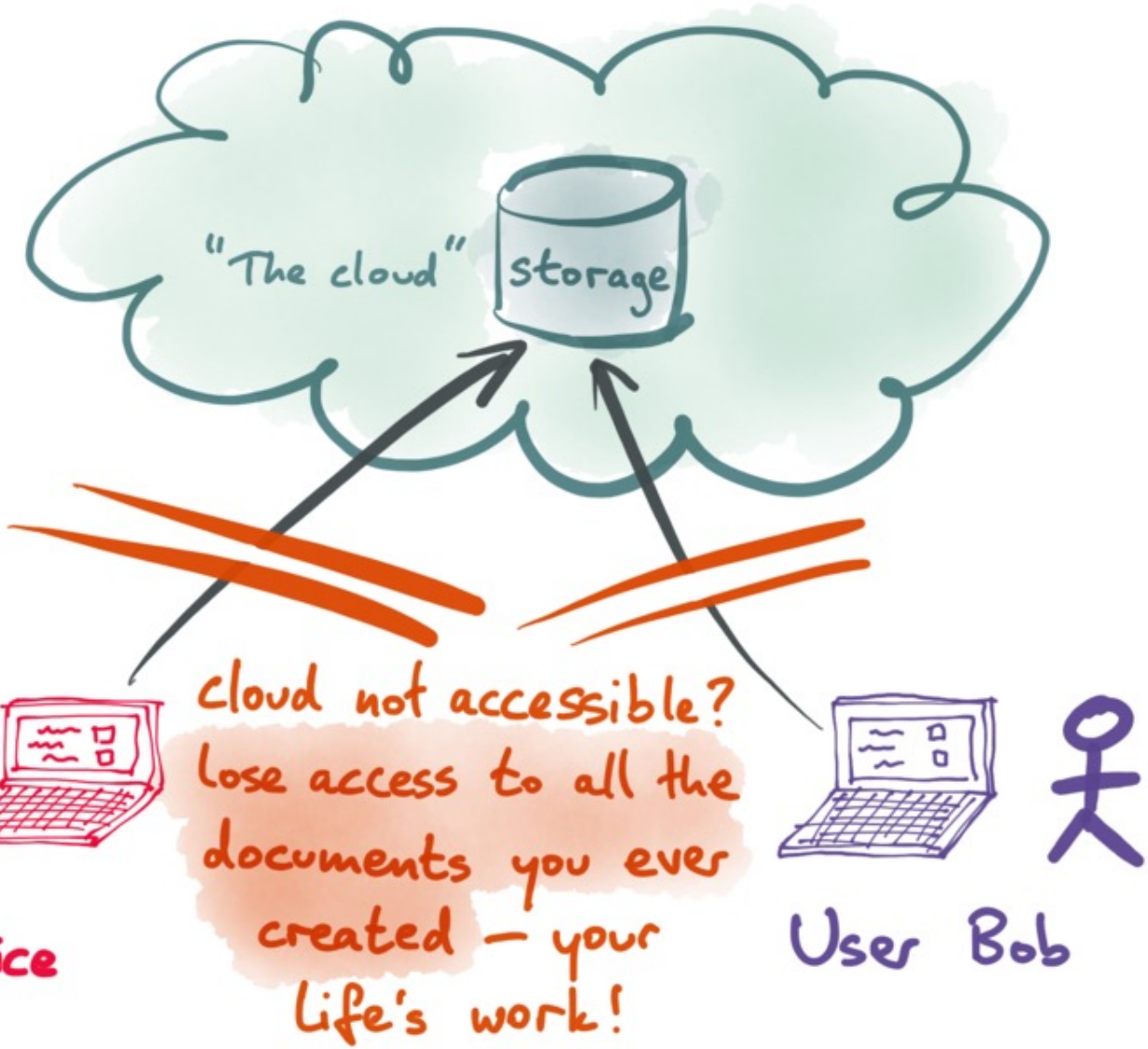
amazon web services



CLOUD SOFTWARE

e.g. Google

amazon web services



The Switch

A mysterious message is locking Google Docs users out of their files

(Loic Venance/AFP/Getty Images)

By **Brian Fung**
October 31, 2017

Imagine you're working on a Google Doc when, seemingly out of nowhere, your ability to edit the online file gets revoked. What you see instead is an error message indicating that you've violated Google's terms of service.

https://www.washingtonpost.com/news/the-switch/wp/2017/10/31/a-mysterious-message-is-locking-google-docs-users-out-of-their-files/

Technology

Google refuses to reinstate man's account after he took medical images of son's groin

Experts say case highlights dangers of automated detection of child sexual abuse images

Tech companies like Google have access to a vast trove of data – but no context for it, says an ACLU technologist. Photograph: Avishek Das/Sopa Images/Rex/Shutterstock

Johana Bhuiyan

Tue 23 Aug 2022 00.32 BST

Google has refused to reinstate a man's account after it wrongly flagged medical images he took of his son's groin as child sexual abuse material (CSAM), the New York Times first reported. Experts say it's an inevitable pitfall of trying to apply a technological solution to a societal problem.

https://www.theguardian.com/technology/2022/aug/22/google-csam-account-blocked

SUSPENDED ACCOUNTS

Microsoft: 182,000 in 2020

of which 730 (0.4%) were restored!

Google: "disables tens of millions of accounts per year"

"The vast majority are spammers and bots"

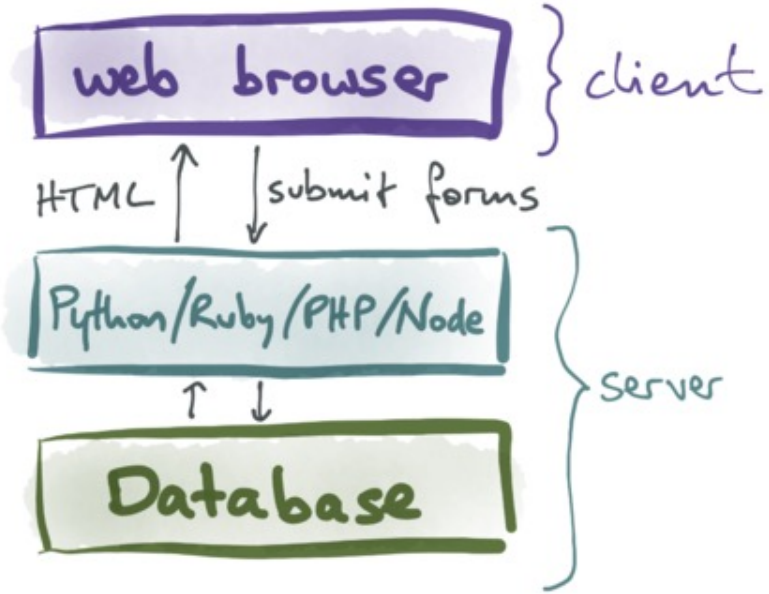
A precarious situation!

Source: https://tweakers.net/reviews/9094/all/account-geblokkeerd-wat-nu.html [in Dutch]

Cuneiform script on clay tablet, ca. 3000 BCE. Image from Wikimedia Commons.
https://commons.wikimedia.org/wiki/File:Early_writing_tablet_recording_the_allocation_of_beer.jpg

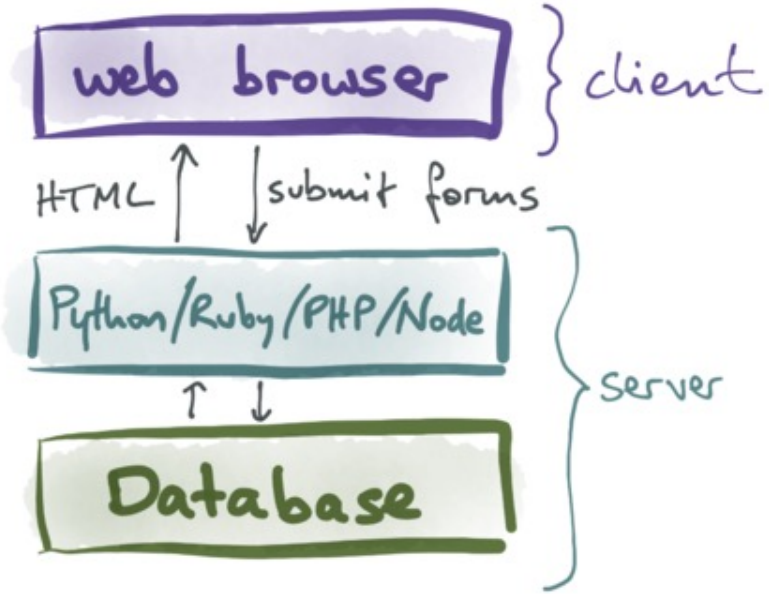
WEB APP ARCHITECTURE THROUGH THE AGES

ca. 2000–2010

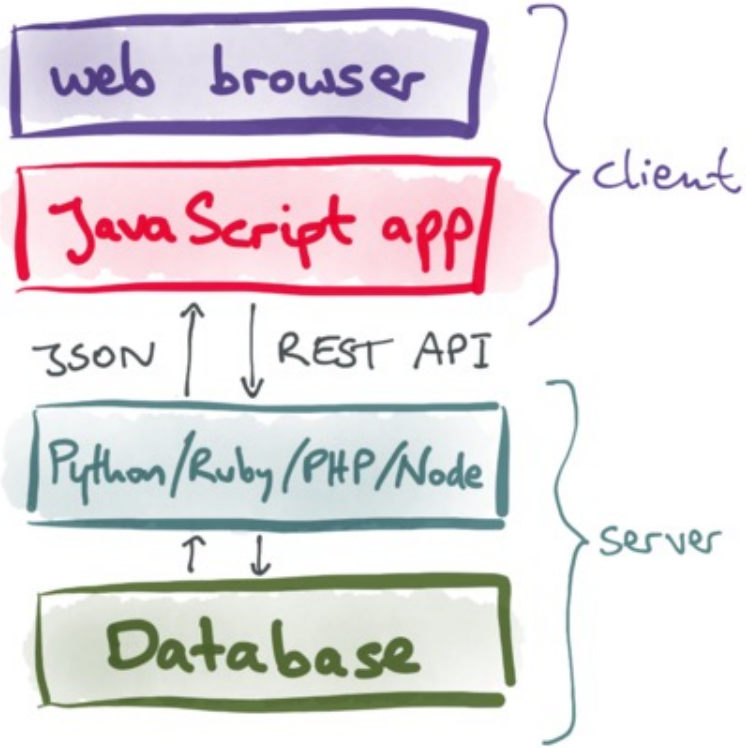


WEB APP ARCHITECTURE THROUGH THE AGES

ca. 2000–2010



ca. 2010–2020

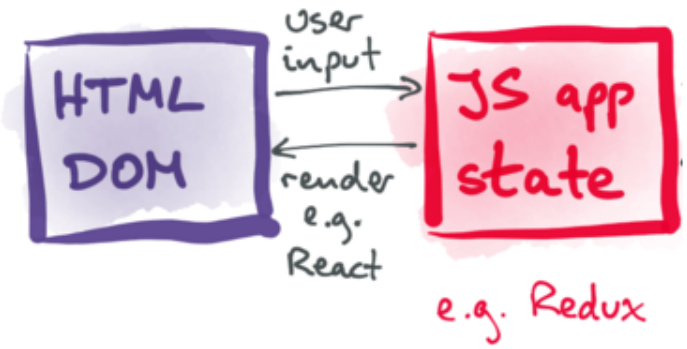


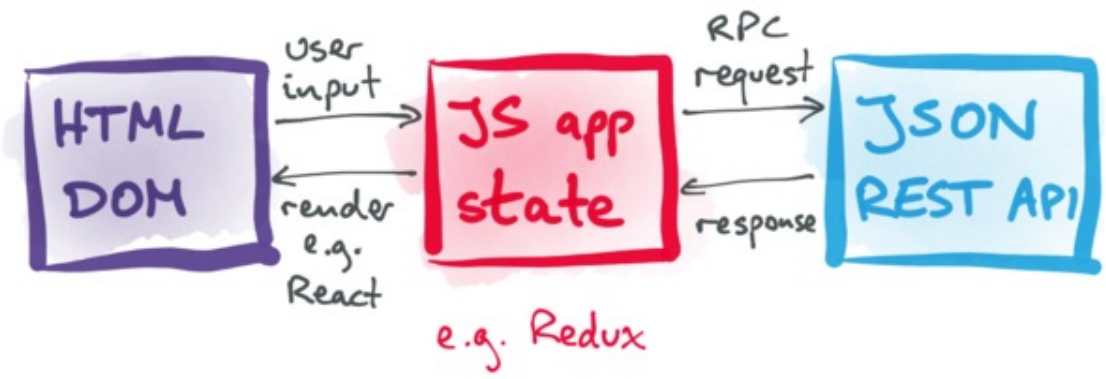
Google Docs

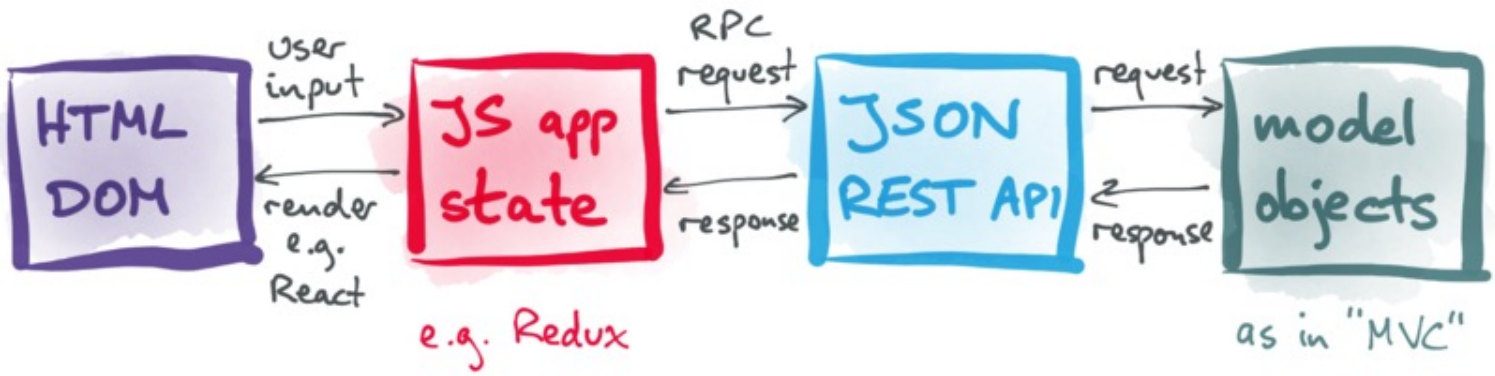
Office 365

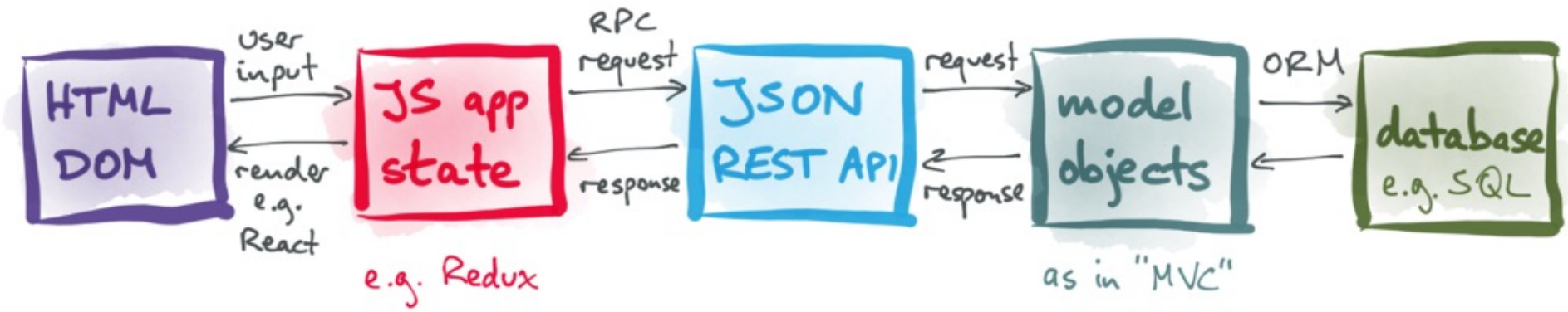
Figma

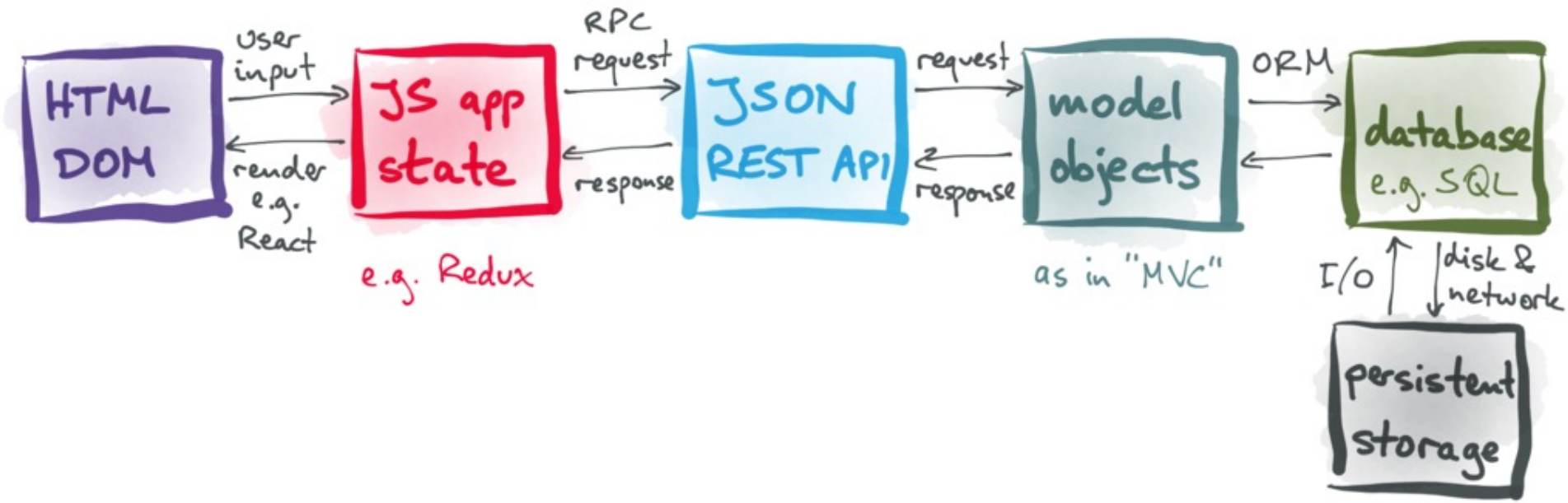
Trello


HTML
DOM
User
input
render
e.g.
React
JS app
state
e.g. Redux

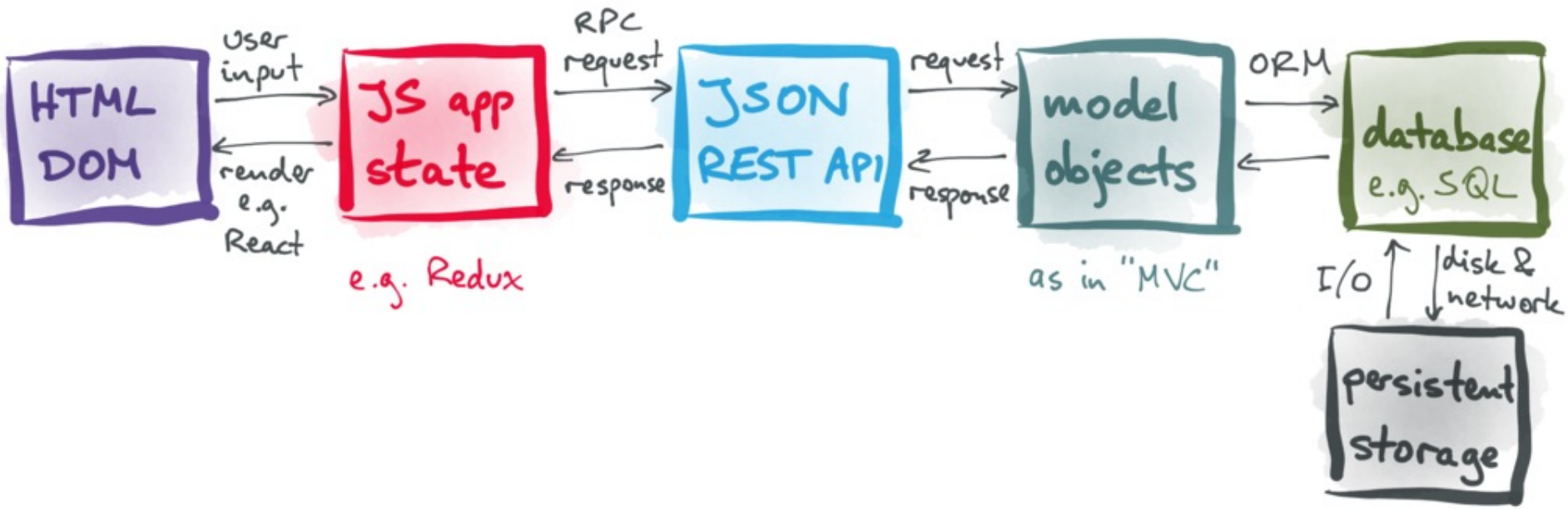

HTML DOM
user input
render e.g. React
JS app state
RPC request
response
JSON REST API
e.g. Redux


HTML DOM
user input
render e.g. React
JS app state
e.g. Redux
RPC request
response
JSON REST API
request
response
model objects
as in "MVC"


HTML DOM
user input
render e.g. React
JS app state
e.g. Redux
RPC request
response
JSON REST API
request
response
model objects
as in "MVC"
ORM
database e.g. SQL


HTML DOM
user input
render e.g. React
JS app state
e.g. Redux
RPC request
response
JSON REST API
request
response
model objects
as in "MVC"
ORM
database e.g. SQL
I/O
disk & network
persistent storage

Six different representations of app state?!



Lots of code is just converting data from one representation to another.

CLOUD APPS / WEB APPS / SAAS

Pros:

- Real-time collaboration
- Access from any device, anywhere

Cons:

- Service shuts down?
  ⇒ everything lost
- Weak offline support
- Server, not user, owns data

CLOUD APPS / WEB APPS / SAAS

Pros:

- Real-time collaboration
- Access from any device, anywhere

Cons:

- Service shuts down? ⇒ everything lost
- Weak offline support
- Server, not user, owns data

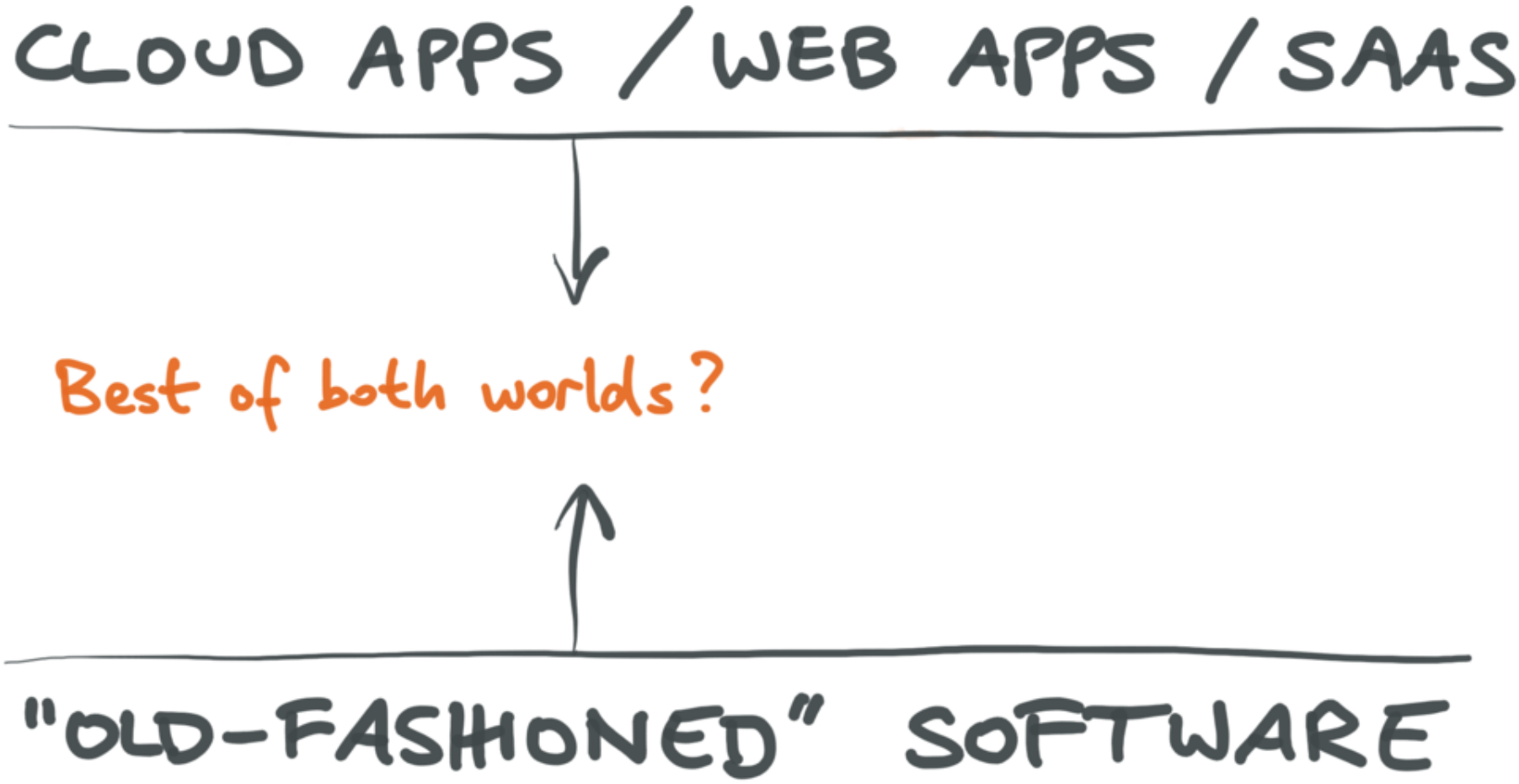
"OLD-FASHIONED" SOFTWARE

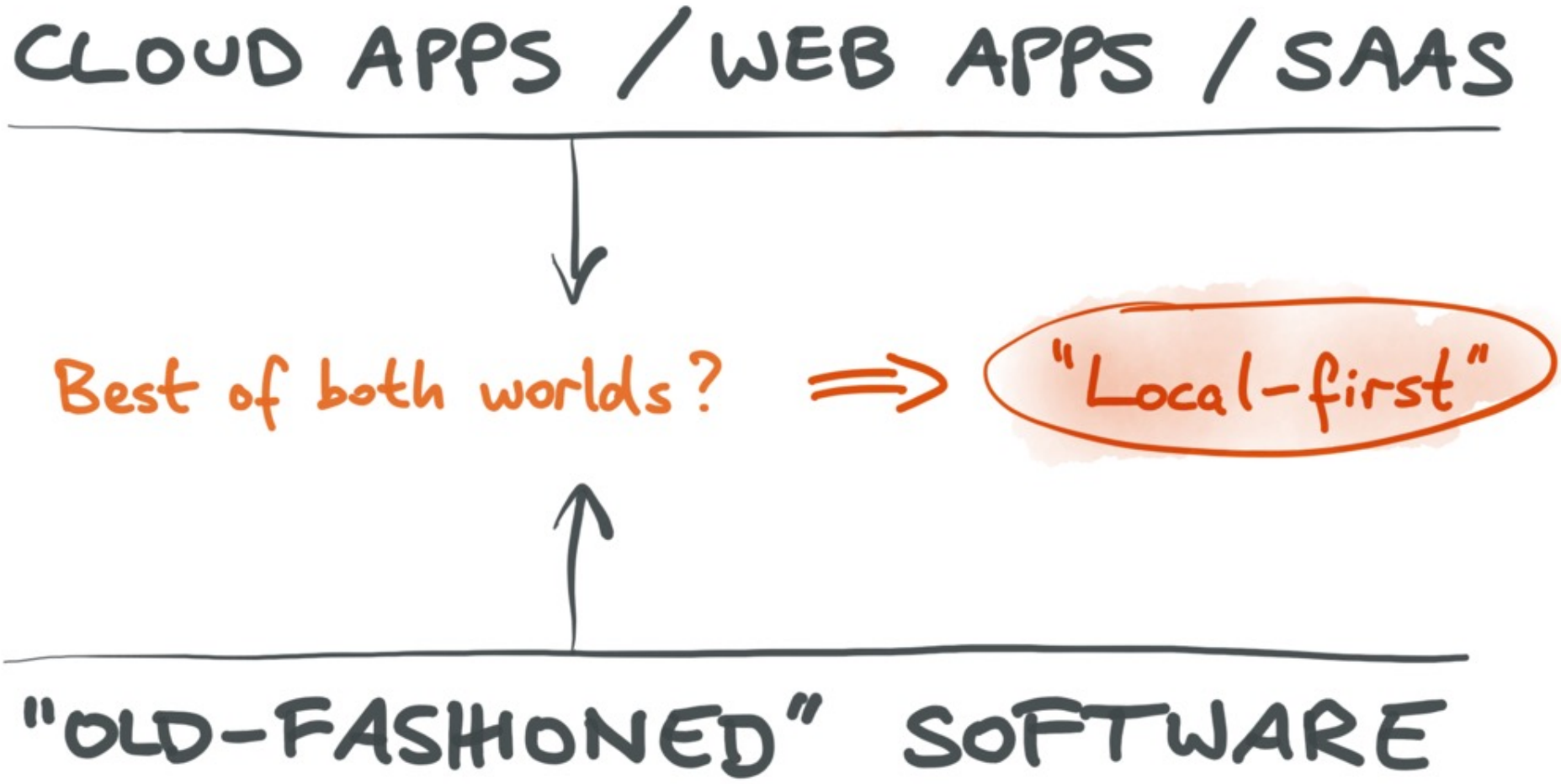
Pros:

- Data & software on your own device: works offline, full control, backups, run software forever (in VM)

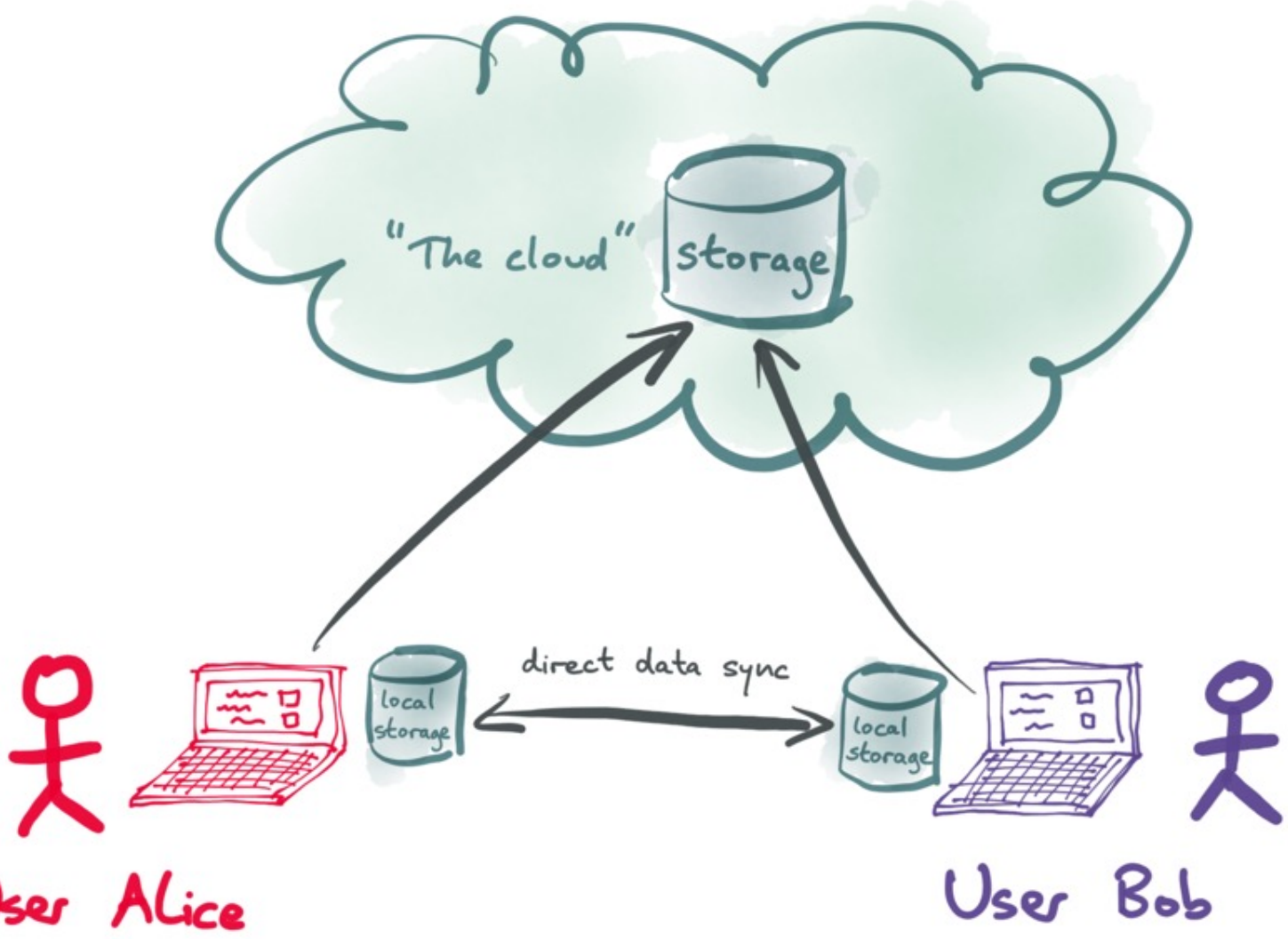
Cons:

- Manual file copying between devices + collaborators
- Only one person at a time can edit file (or manual merge)


CLOUD APPS / WEB APPS / SAAS
Best of both worlds?
"OLD-FASHIONED" SOFTWARE


CLOUD APPS / WEB APPS / SAAS
Best of both worlds?
"Local-first"
"OLD-FASHIONED" SOFTWARE

LOCAL-FIRST SOFTWARE



TRADITIONAL WEB APP MODEL:

"If it's not stored in the server database, it didn't really happen"

Can't reach the server? Can't do anything!

TRADITIONAL WEB APP MODEL:

"If it's not stored in the server database, it didn't really happen"

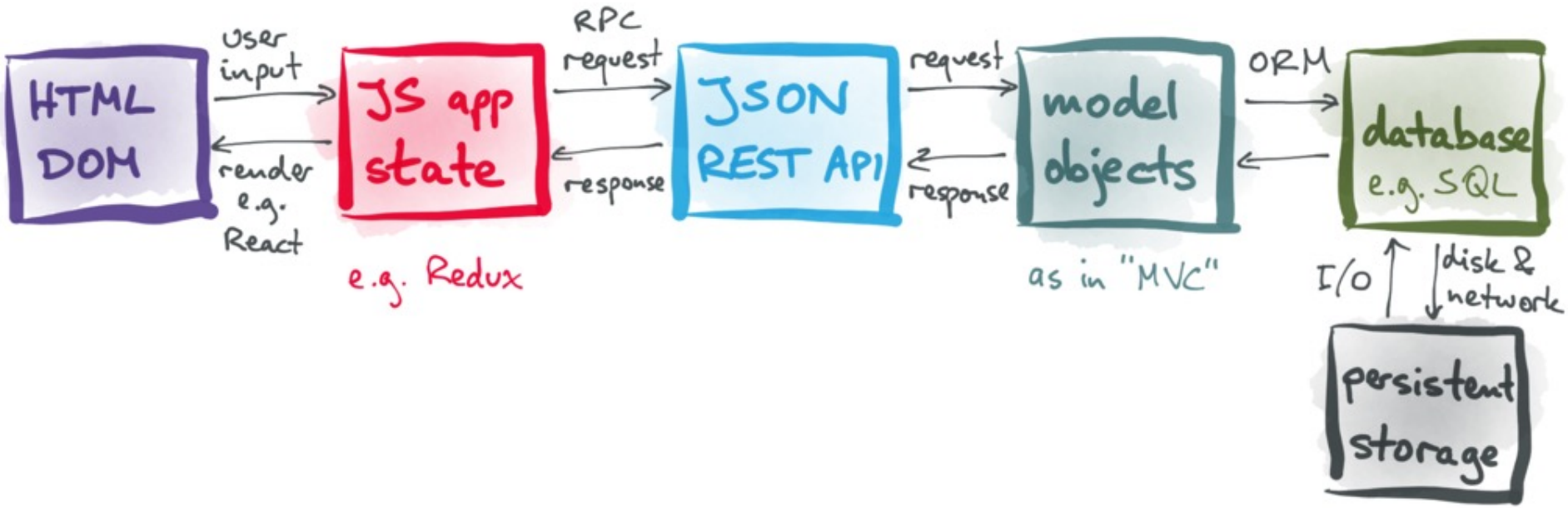
Can't reach the server? Can't do anything!

LOCAL-FIRST MODEL:

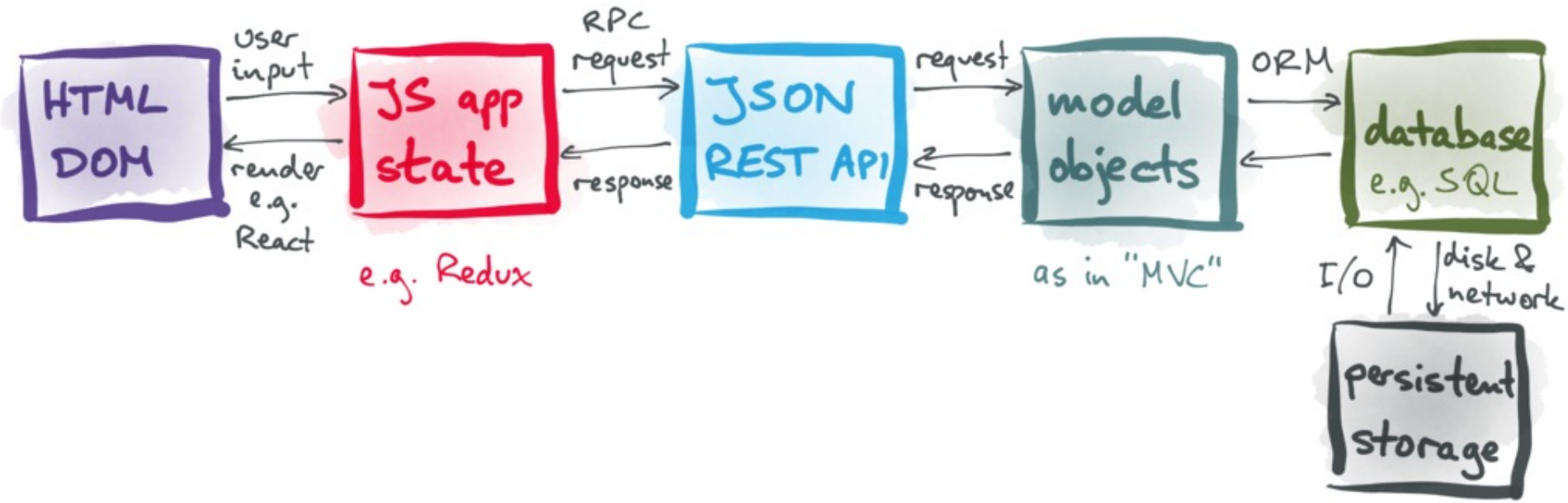
"The client's local storage is what matters — the server is just for multi-user sync and backup"

Don't care if we're online or offline right now!

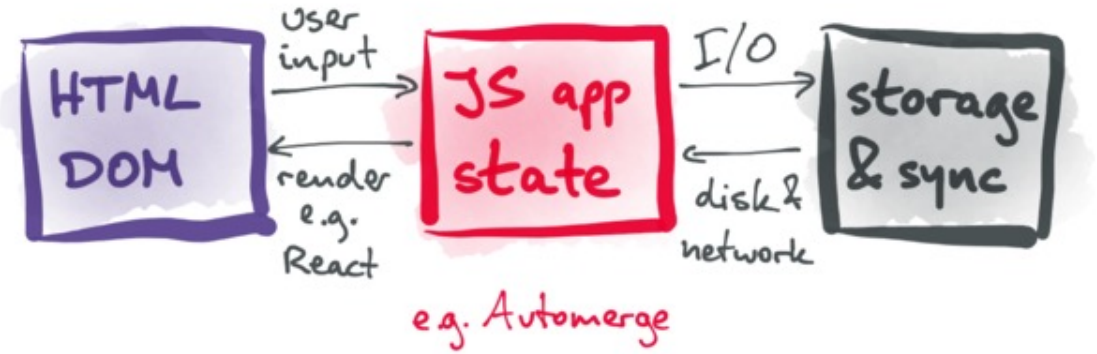
TRADITIONAL WEB APP MODEL:



TRADITIONAL WEB APP MODEL:



LOCAL-FIRST MODEL:



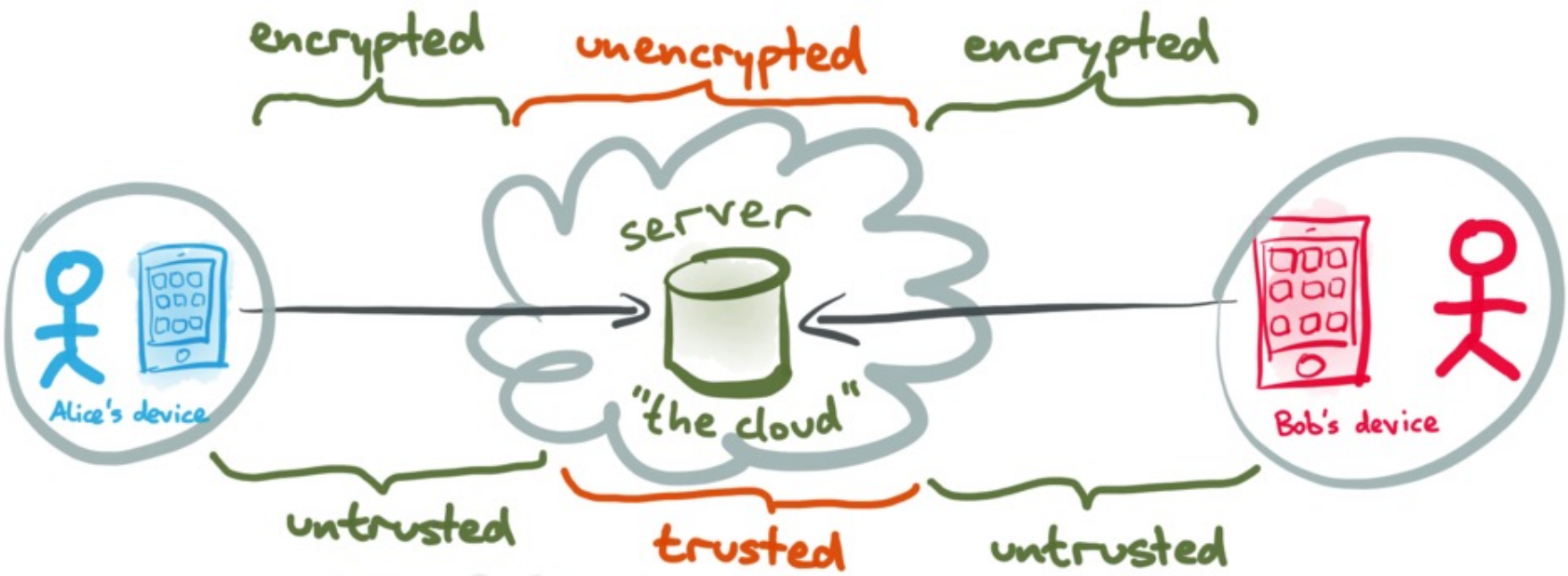
Self-contained!

End-to-end encryption for billions of users

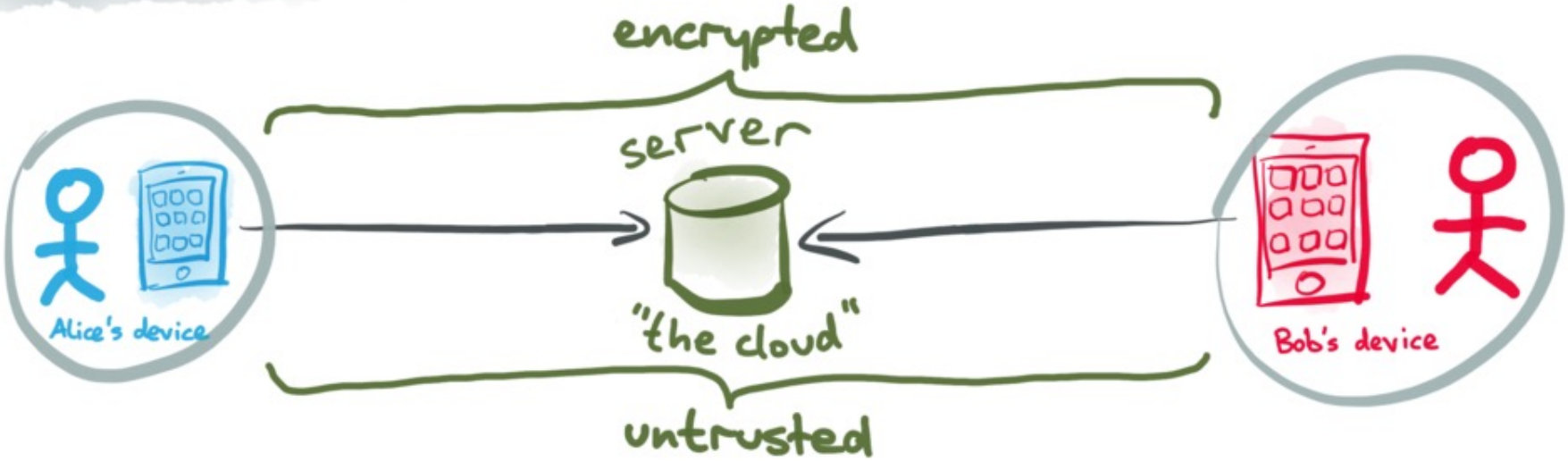
WhatsApp

Signal

iMessage



End-to-end encryption:



CLOUD SOFTWARE	LOCAL-FIRST SOFTWARE
 Real-time collaboration is hard to implement	 Built for real-time sync

CLOUD SOFTWARE	LOCAL-FIRST SOFTWARE
 Real-time collaboration is hard to implement	 Built for real-time sync
 Does not work offline	 Works offline

CLOUD SOFTWARE	LOCAL-FIRST SOFTWARE
 Real-time collaboration is hard to implement	 Built for real-time sync
 Does not work offline	 Works offline
 Service shuts down? Lose everything!	 Users can continue using local copy of software + data

CLOUD SOFTWARE	LOCAL-FIRST SOFTWARE
☹ Real-time collaboration is hard to implement	☺ Built for real-time sync
☹ Does not work offline	☺ Works offline
☹ Service shuts down? Lose everything!	☺ Users can continue using local copy of software + data
☹ Custom service for each app (infra, ops, on-call rotation, ...)	☺ Sync service is generic ⇒ outsource to cloud vendor

Local-first is a good fit for:

"File editing" software

(text editor, word processor, spreadsheet, presentation slides, graphics editor, video editing, music production, CAD software for engineering, Jupyter notebooks, ...)

Local-first is a bad fit for:

Local-first is a good fit for:

"File editing" software

(text editor, word processor, spreadsheet, presentation slides, graphics editor, video editing, music production, CAD software for engineering, Jupyter notebooks, ...)

Productivity software

(notes, to-do lists, issue trackers, calendar, time tracking, group messaging, bookkeeping...)

Basically, apps where the user can edit the data however they like.

Local-first is a bad fit for:

Local-first is a good fit for:

"File editing" software

(text editor, word processor, spreadsheet, presentation slides, graphics editor, video editing, music production, CAD software for engineering, Jupyter notebooks, ...)

Productivity software

(notes, to-do lists, issue trackers, calendar, time tracking, group messaging, bookkeeping...)

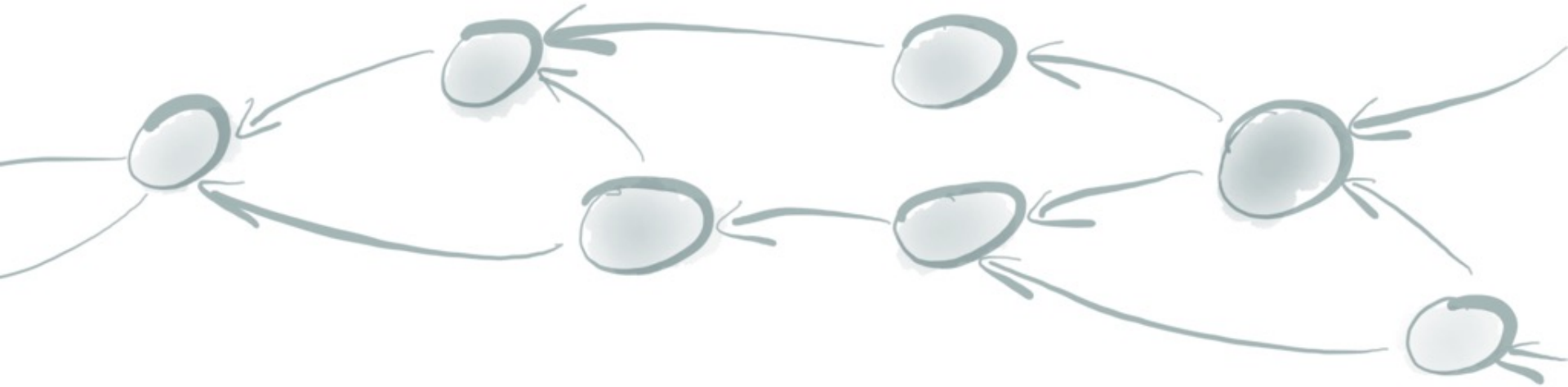
Basically, apps where the user can edit the data however they like.

Local-first is a bad fit for:

Managing a real-world resource, e.g.

- **money** (bank account, payments, ad impressions)
- **physical products** (e-commerce, warehouse inventory)
- **vehicles** (car-sharing/rental, freight/logistics)

For these apps, a centralised cloud/server model works best.



Automerge

https://automerge.org

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

```
after = Automerge.change(before, "mark item as done", doc => {
    doc.todos[1].done = true;
});
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

true

```
after = Automerge.change(before, "mark item as done", doc => {
    doc.todos[1].done = true;
});
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

true

reflects updated state

immutable

```
after = Automerge.change(before, "mark item as done", doc => {
    doc.todos[1].done = true;
});
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

true

reflects updated state

immutable

"commit message" (optional)

```
after = Automerge.change(before, "mark item as done", doc => {
    doc.todos[1].done = true;
});
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

true

reflects updated state

immutable

"commit message" (optional)

```
after = Automerge.change(before, "mark item as done", doc => {
    doc.todos[1].done = true;
});
```

record exactly what changed

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk", "done": false},
    {"title": "water plants", "done": false}
]}
```

```
after = Automerge.change(before, "add new item", doc => {
    doc.todos.push({title: "do laundry", done: false});
});
```

AUTOMERGE: "Git for your app's data"

```
{"todos": [
    {"title": "buy milk",    "done": false},
    {"title": "water plants", "done": false},
    {"title": "do laundry",   "done": false}
]}
```

← added

```
after = Automerge.change(before, "add new item", doc => {
    doc.todos.push({title: "do laundry", done: false});
});
```

AUTOMERGE: "Git for your app's data"

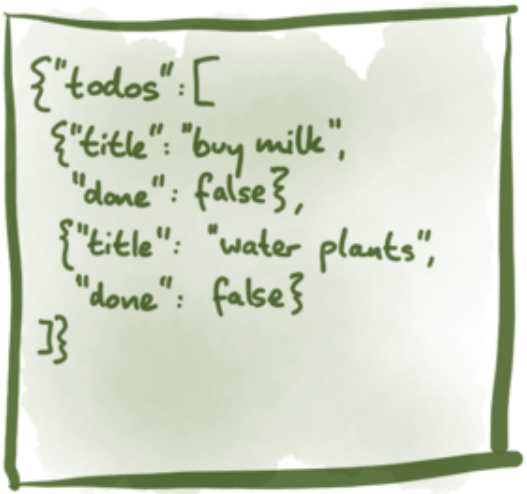
```
{"todos": [
    {"title": "buy milk",    "done": false},
    {"title": "water plants", "done": false},
    {"title": "do laundry",  "done": false}
]}
```

← added

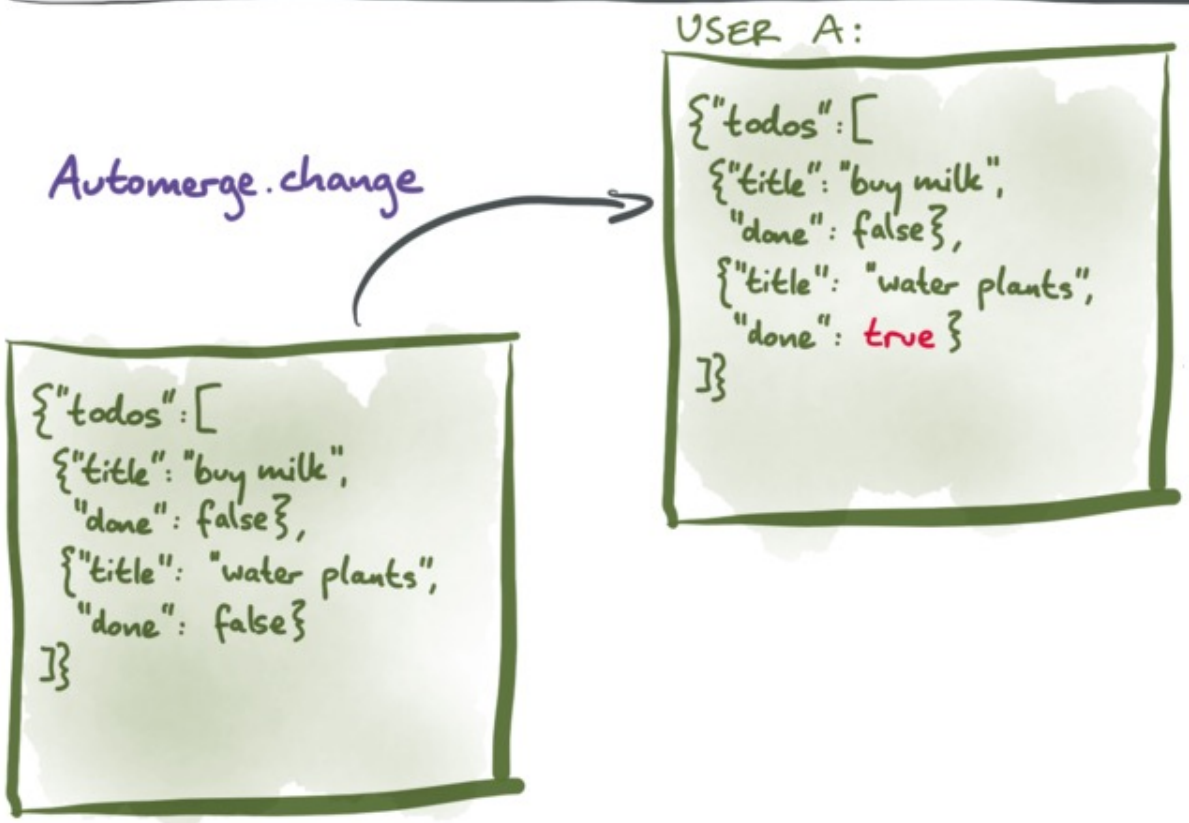
```
after = Automerge.change(before, "add new item", doc => {
    doc.todos.push({title: "do laundry", done: false});
});
```

append item to list

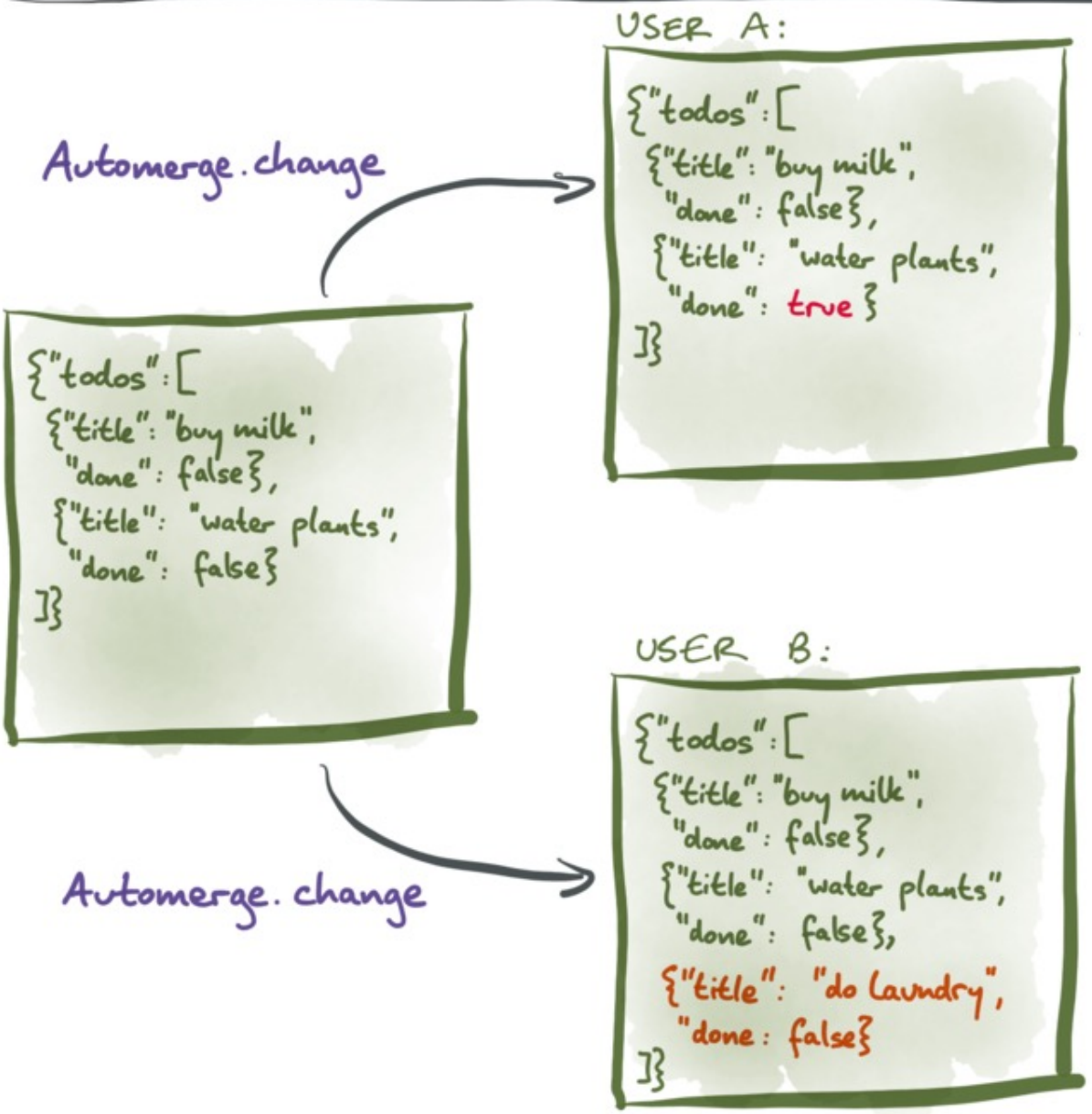
AUTOMERGE: Branching and merging



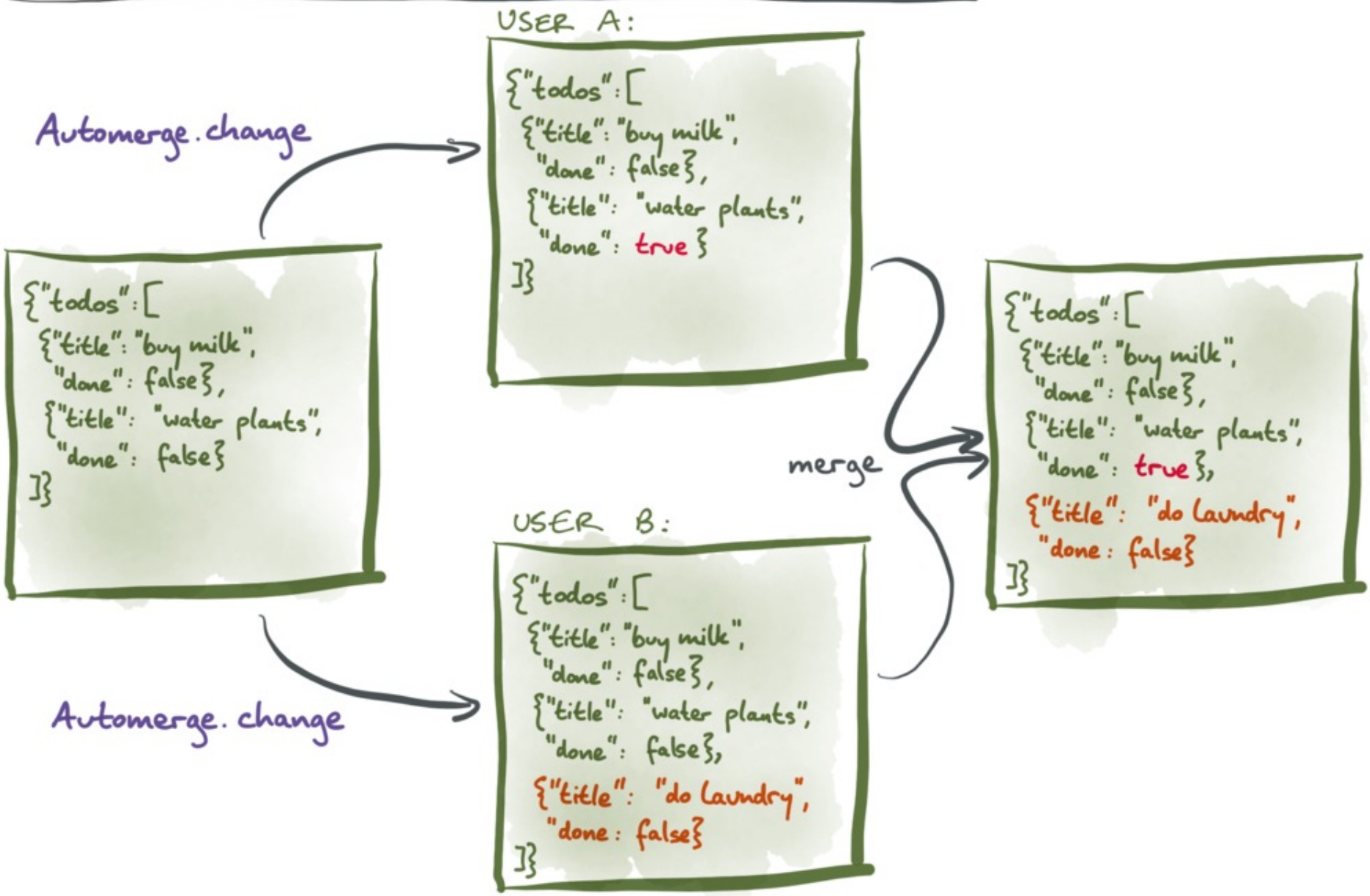
AUTOMERGE: Branching and merging



AUTOMERGE: Branching and merging



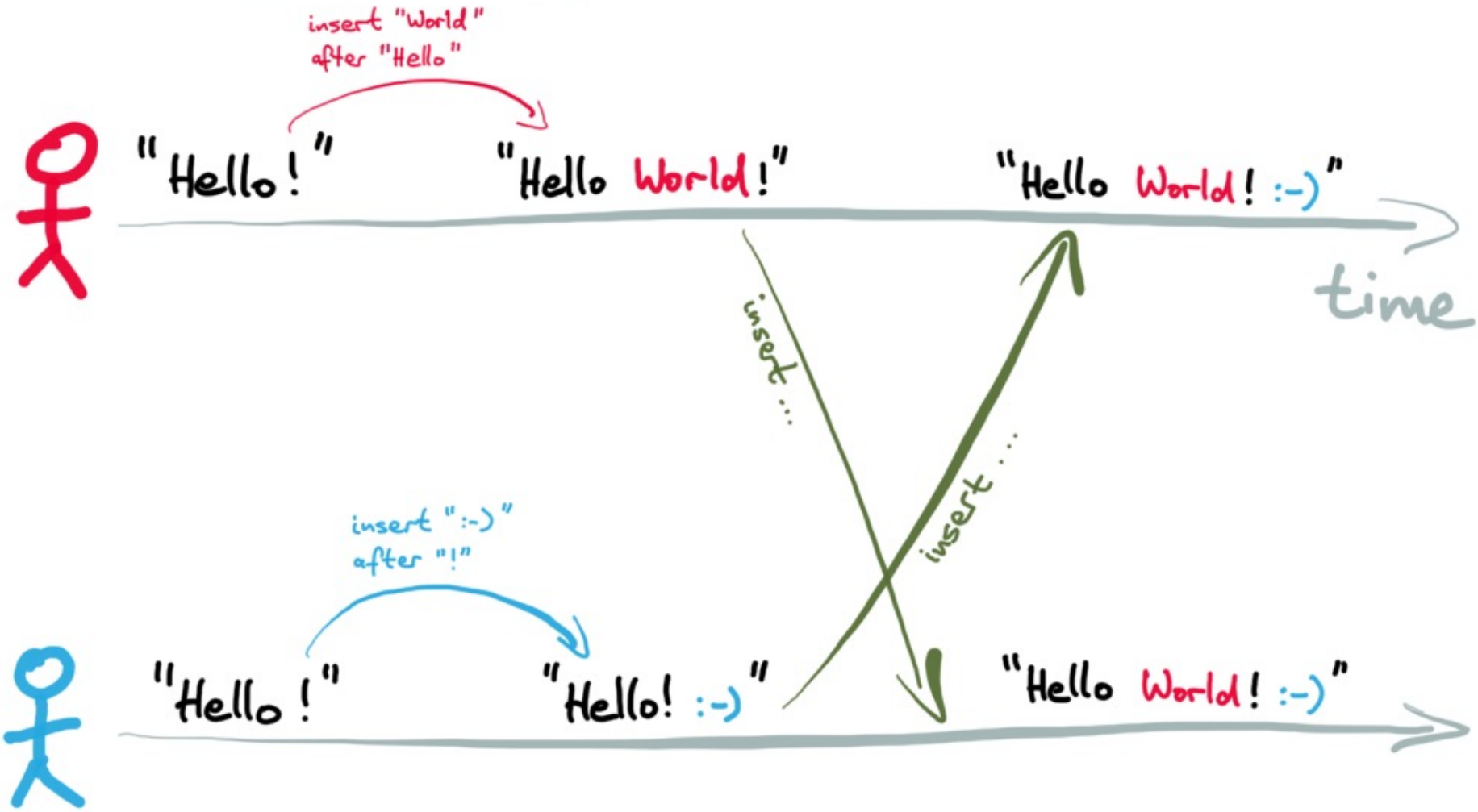
AUTOMERGE: Branching and merging



Example: Text editing

Example: Text editing

Example: Text editing



Automerge guarantees:

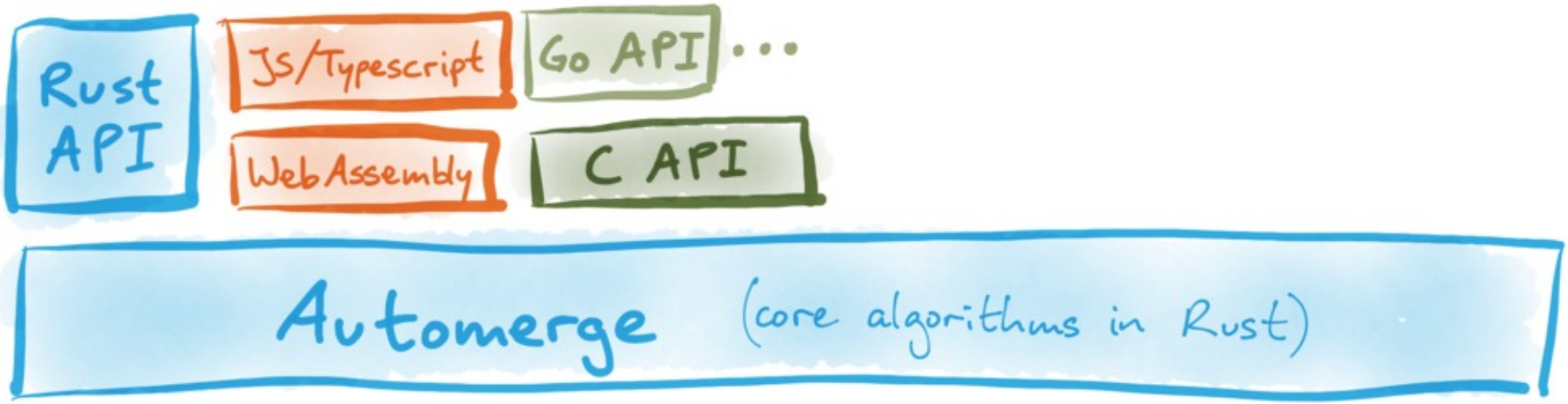
- All changes are preserved
- If two users have seen the same set of changes (in any order), then they are in the same state
- Branches (=concurrent updates) can be merged automatically
- Can branch & merge arbitrarily, can inspect/compare versions

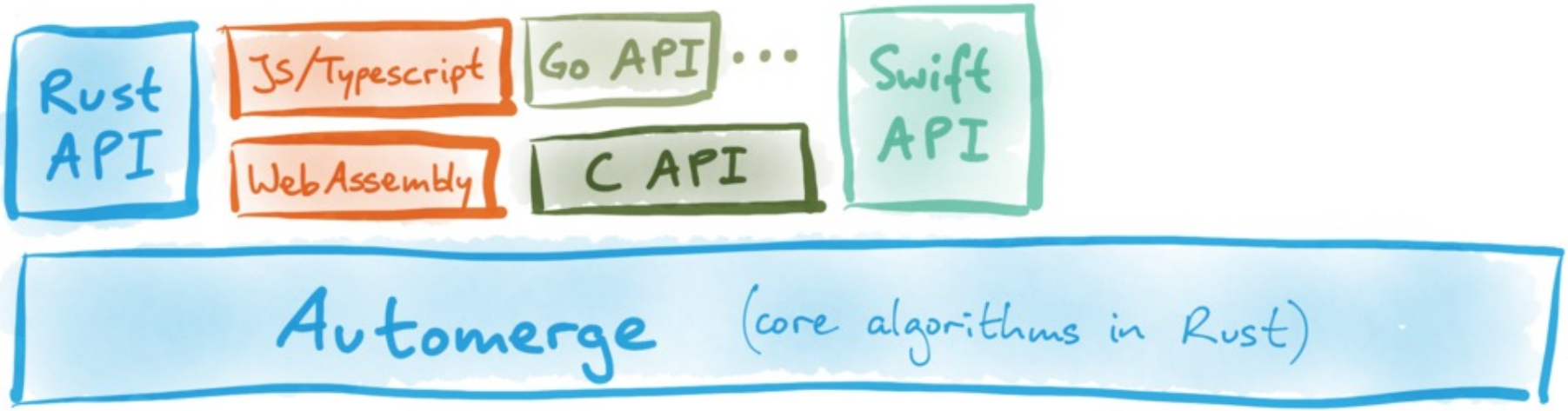
Automerge is a CRDT

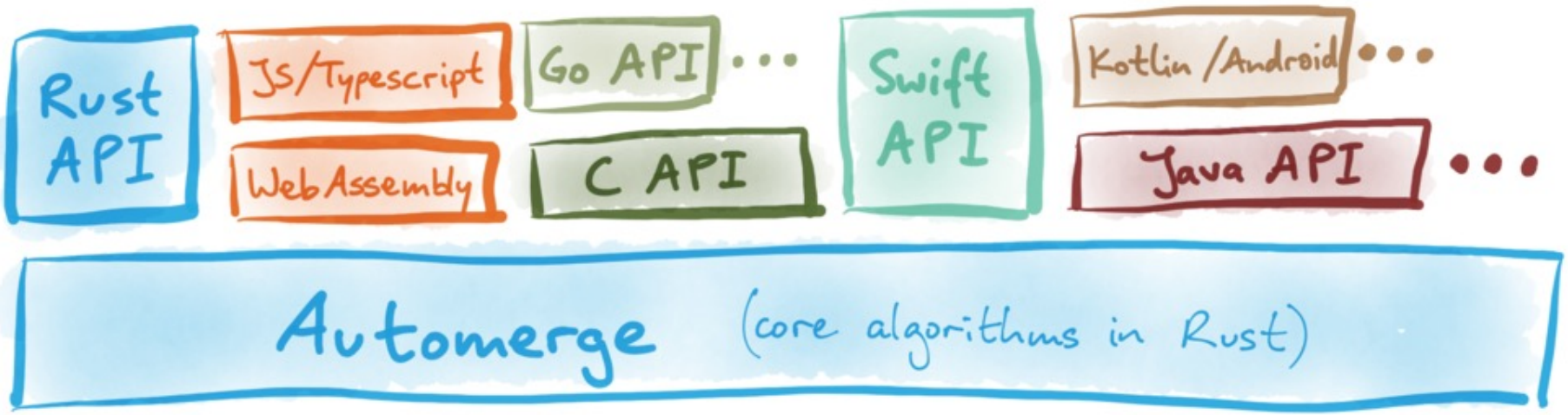
(conflict-free replicated data type)

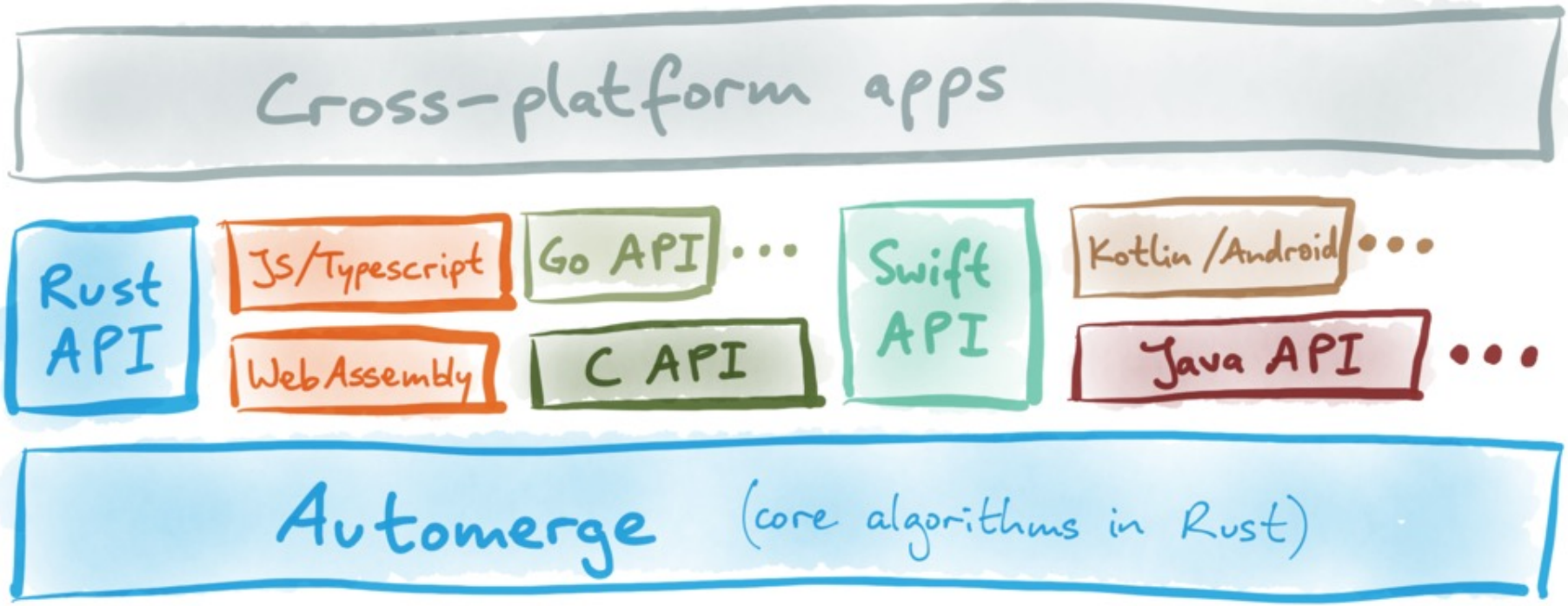

Rust
API
Automerge (core algorithms in Rust)

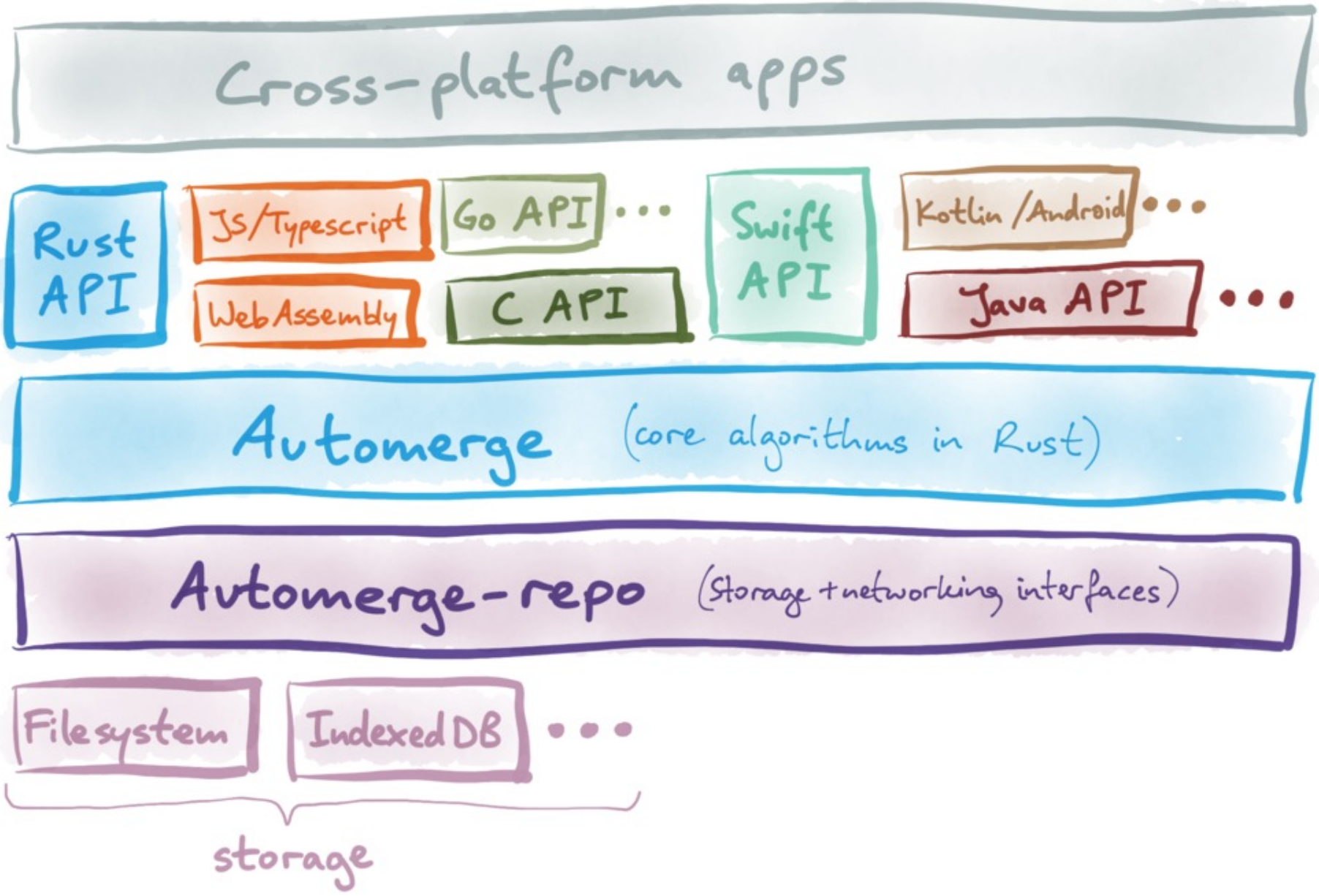

Rust API
JS/Typescript
WebAssembly
Automerge (core algorithms in Rust)

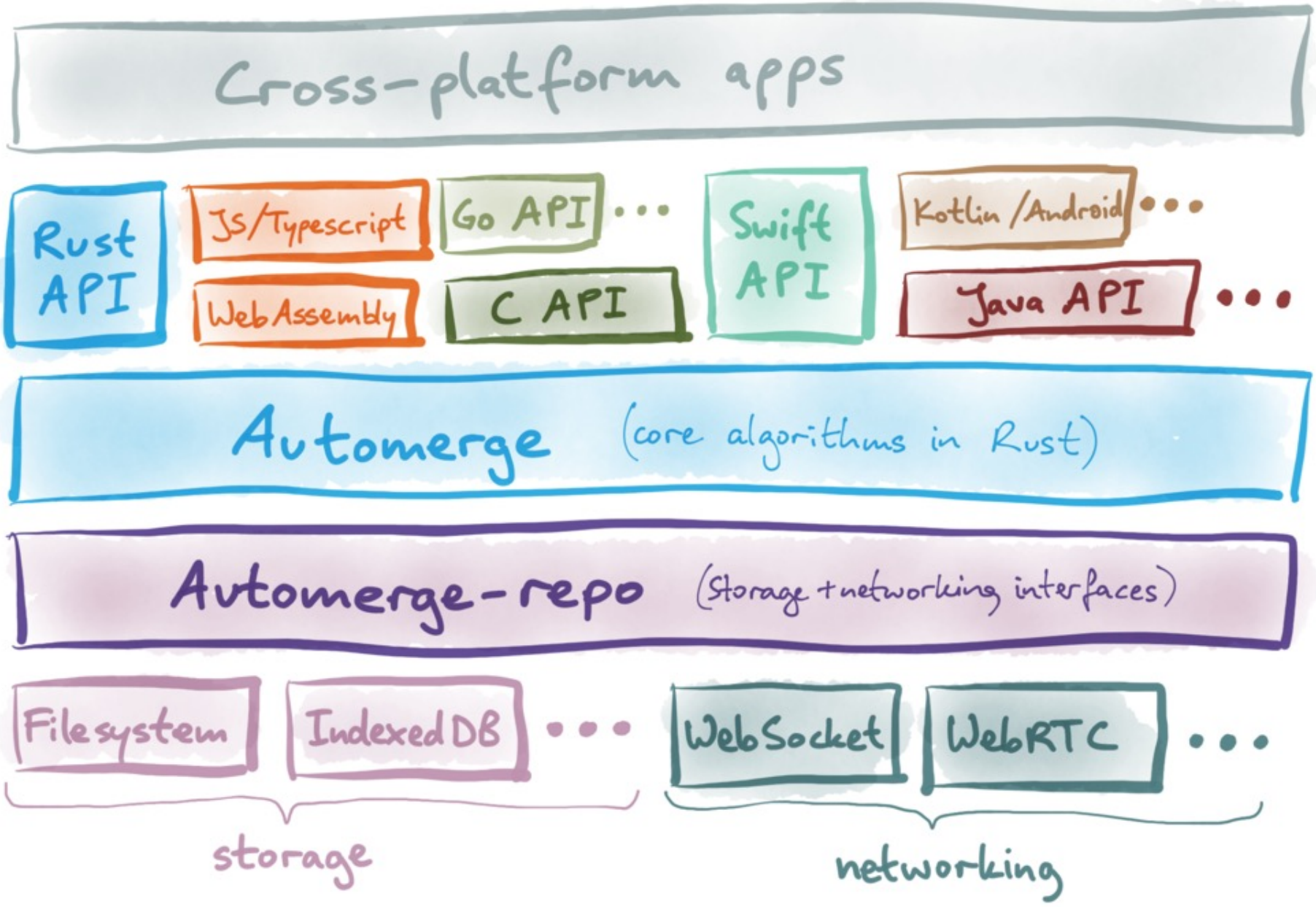

Rust API
JS/Typescript
Go API
...
WebAssembly
C API
Automerge
(core algorithms in Rust)


Rust API
JS/Typescript
Web Assembly
Go API
...
C API
Swift API
Automerge
(core algorithms in Rust)


Rust API
JS/Typescript
Web Assembly
Go API
C API
Swift API
Kotlin /Android
Java API
Automerge
(core algorithms in Rust)


Cross-platform apps
Rust API
JS/Typescript
WebAssembly
Go API
C API
Swift API
Kotlin/Android
Java API
Automerge
(core algorithms in Rust)


Cross-platform apps
Rust API
JS/Typescript
Web Assembly
Go API
C API
Swift API
Kotlin /Android
Java API
Automerge
(core algorithms in Rust)
Automerge-repo
(Storage + networking interfaces)
Filesystem
IndexedDB
storage


Cross-platform apps
Rust API
JS/Typescript
Web Assembly
Go API
C API
Swift API
Kotlin /Android
Java API
Automerge
(core algorithms in Rust)
Automerge-repo
(Storage + networking interfaces)
Filesystem
IndexedDB
WebSocket
WebRTC
storage
networking

Resources

Local-first	https://www.inkandswitch.com/local-first/
Automerge	https://automerge.org/
My work	https://martin.kleppmann.com/
Email	martin@kleppmann.com
Bluesky	@martin.kleppmann.com
Mastodon	@martin@nondeterministic.computer

Huge thanks to the Automerge community and contributors, especially Alex Good, Peter van Hardenberg, Orion Henry, Andrew Jeffery, Herb Caudill, Alex Currie-Clark, Jason Kankiewicz, Conrad Irwin, and many others!

Thank you to my Patreon backers and institutional supporters: