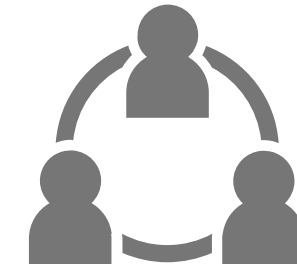
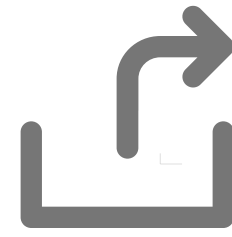
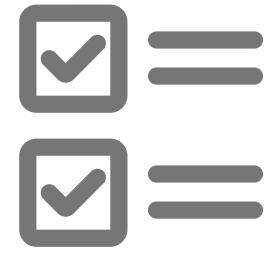
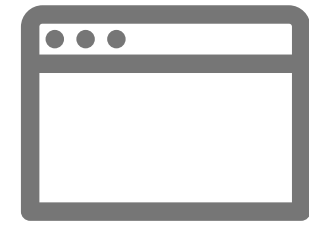




**Every Node.js app should have
real-time collaboration features.**





Vladi Stevanovic

Director of Community
Multiplayer



@vladistevanovic



Vladi Stevanovic



vladi@multiplayer.app





We make it easier for teams to design, develop, and manage distributed software

The screenshot displays the Multiplayer interface for a project named "web-app" in a "v2/new-feature" branch. The main workspace shows a hierarchical architecture diagram with a "web-app" node on the left, branching into "auth-service", "web-api-service", "search-service", and "messaging-service". The "web-api-service" node further branches into "RabbitMQ" and a "data-platform" container. The "data-platform" container includes "clone-service", "sync-service", and "polling-service". A "You" cursor is positioned near the "web-app" node, and a "Tom" cursor is pointing at the "RabbitMQ" node. The interface includes a top navigation bar with "Merge" and "Draft" buttons, and a sidebar with various tool icons. An "AI Assist" chat window is open on the right, with the following content:

Multiplayer AI Assist

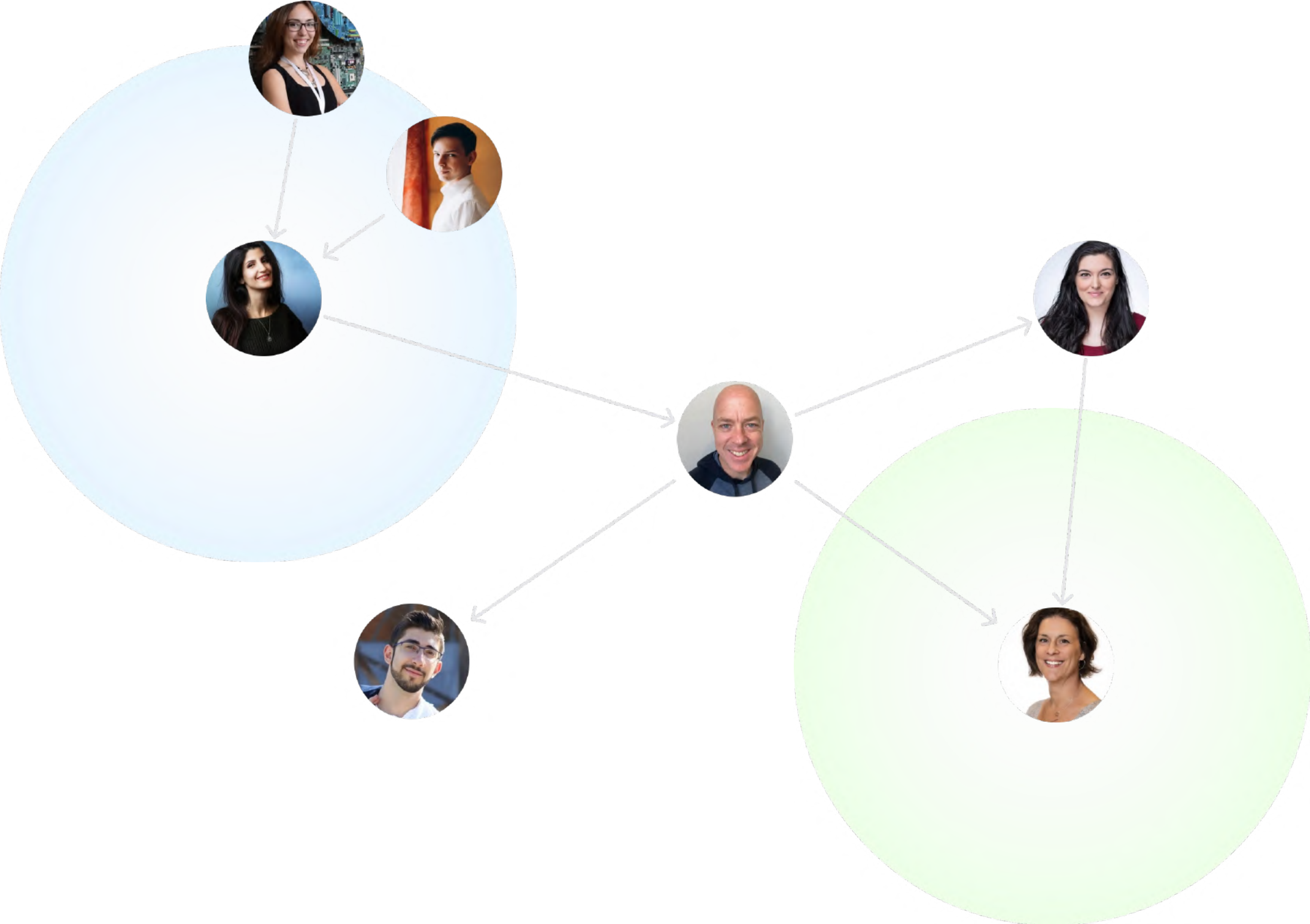
AI Assist from Multiplayer
Hi there 🙌
How can I help you today?

- ✓ Create a README for my feature branch
- Generate code for the new API methods in this feature branch
- Why does the platform feel slow now?
- Generate tests for this feature branch.

You
Generate the release notes for this feature branch and send them to the team on Slack.

Try Pitch

Real-time, visual collaboration
is a core requirement for
Multiplayer





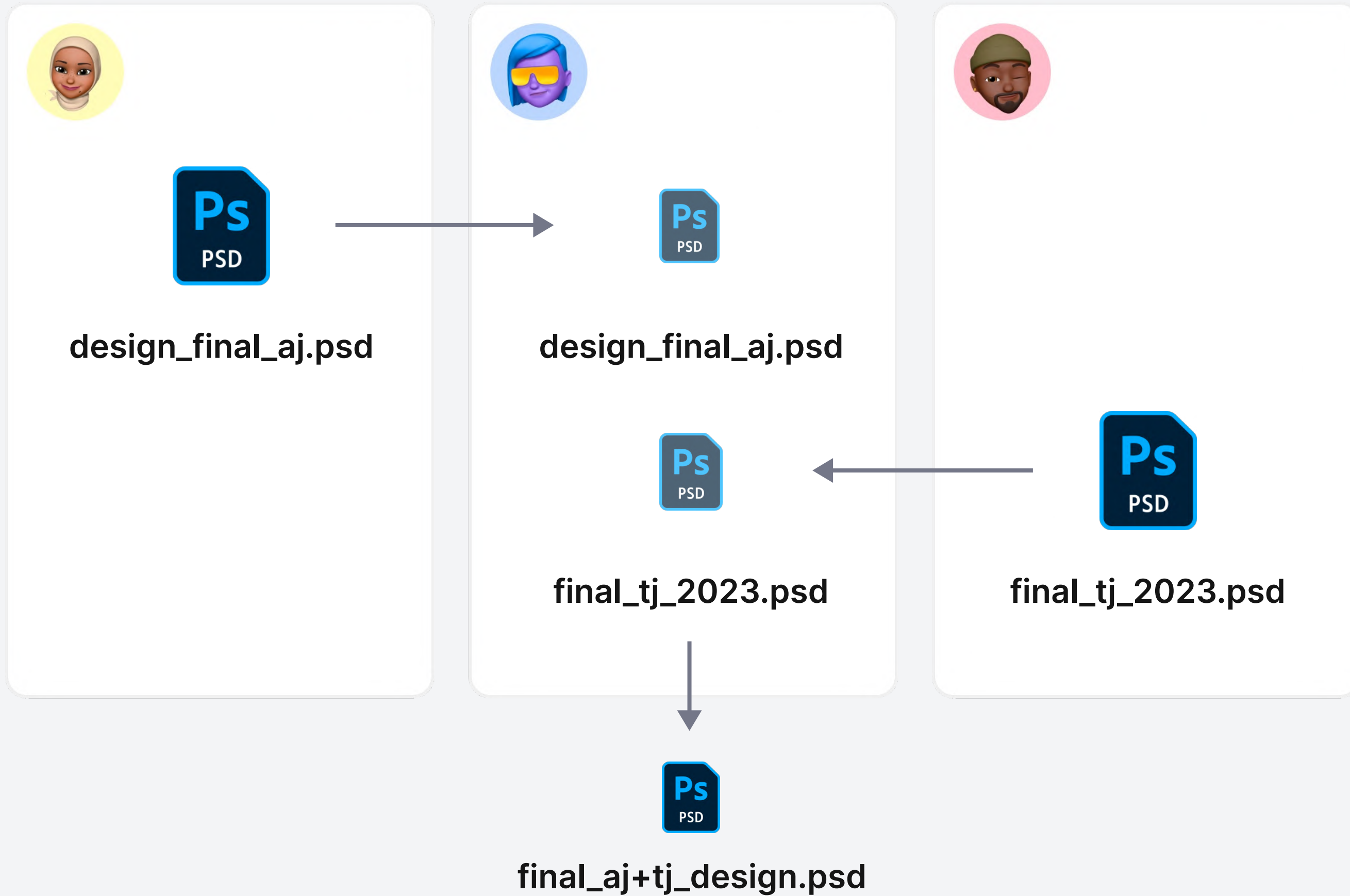
Real-time collaboration 



design.psd

design.psd

design.psd



Following Hermann X



Hermann



design

Carmen

Mike

Following Hermann X



Hermann



Carmen

Mike

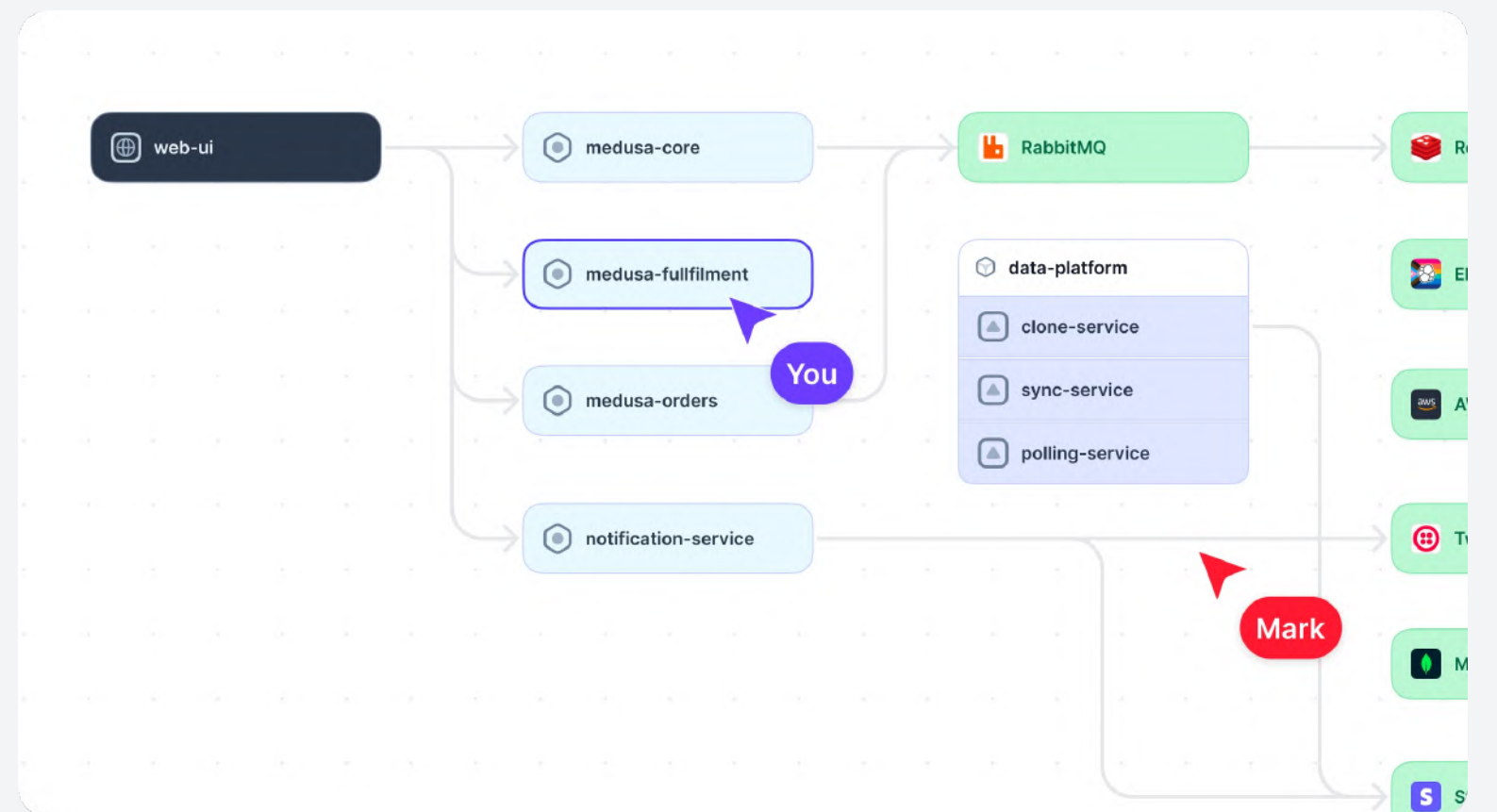
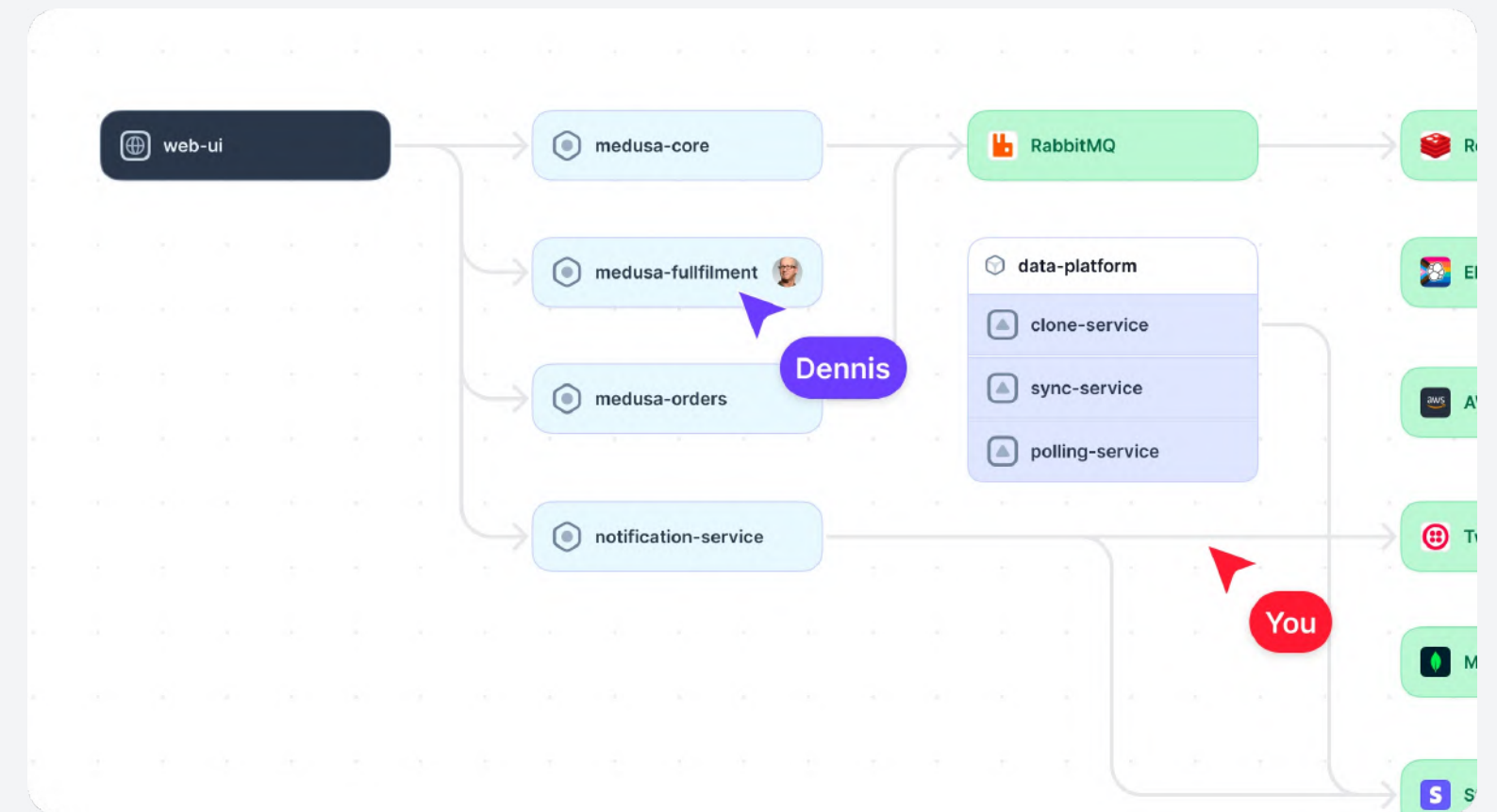
WHAT IS REAL-TIME COLLABORATION?

One
Real-time updates

Two
Live playback

Three
Presence and status

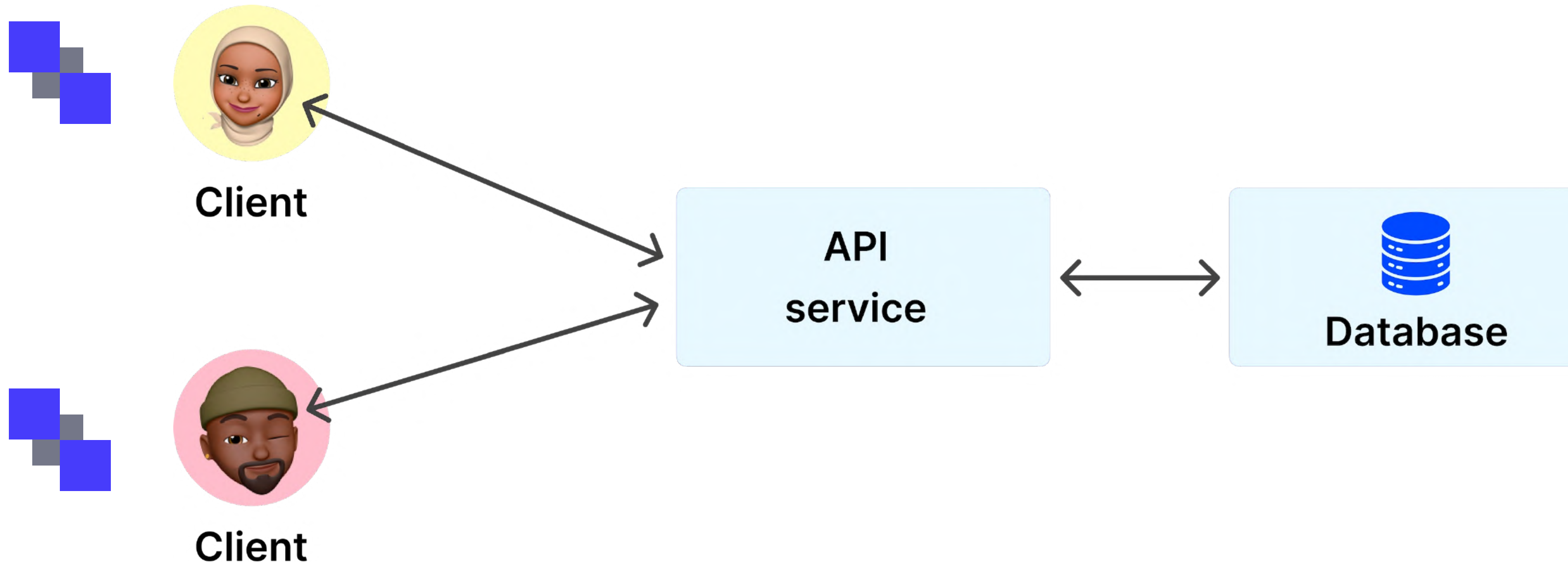
Four
Data integrity



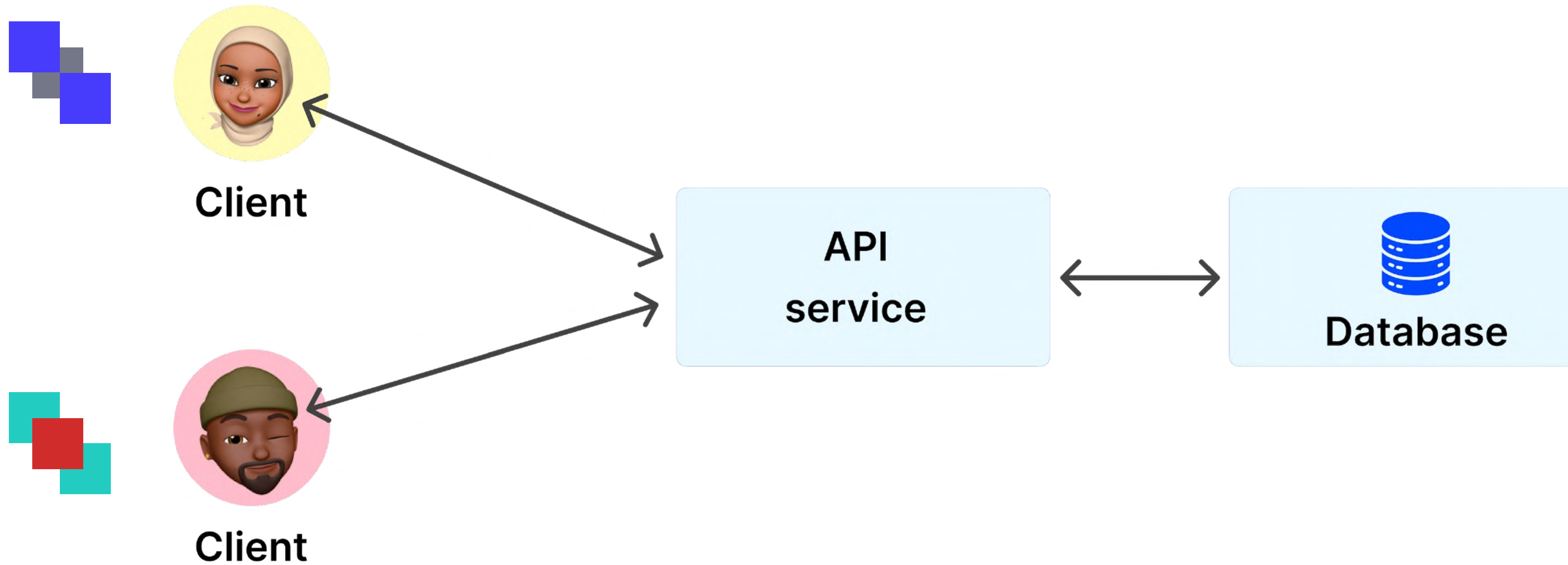


How do you implement it?

The hard way.



The hard way.



OTs

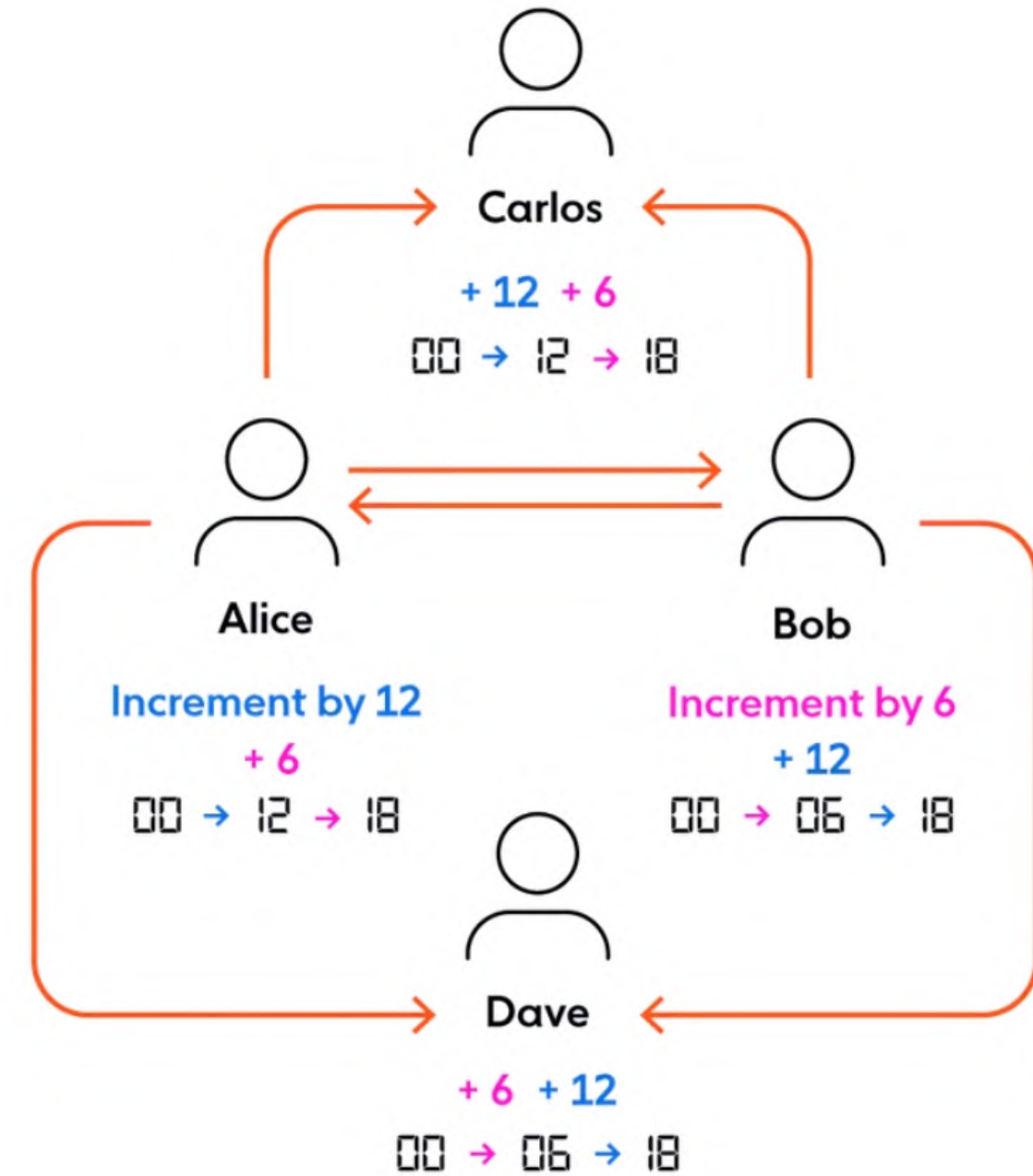
Operational Transformations

CRDTs

Conflict-Free Replicated Data Types

CRDTs

Conflict-Free Replicated Data Types





Ready open source integrations

Support for any data structure

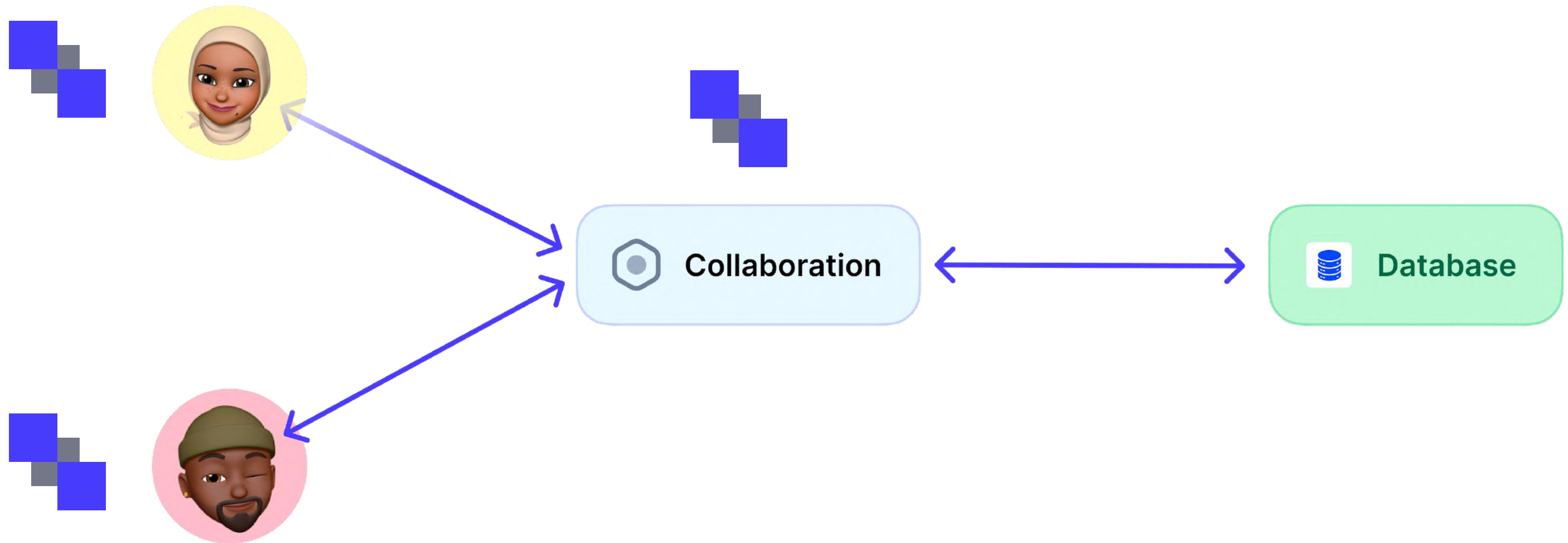
Painless client reconnect

Network-agnostic

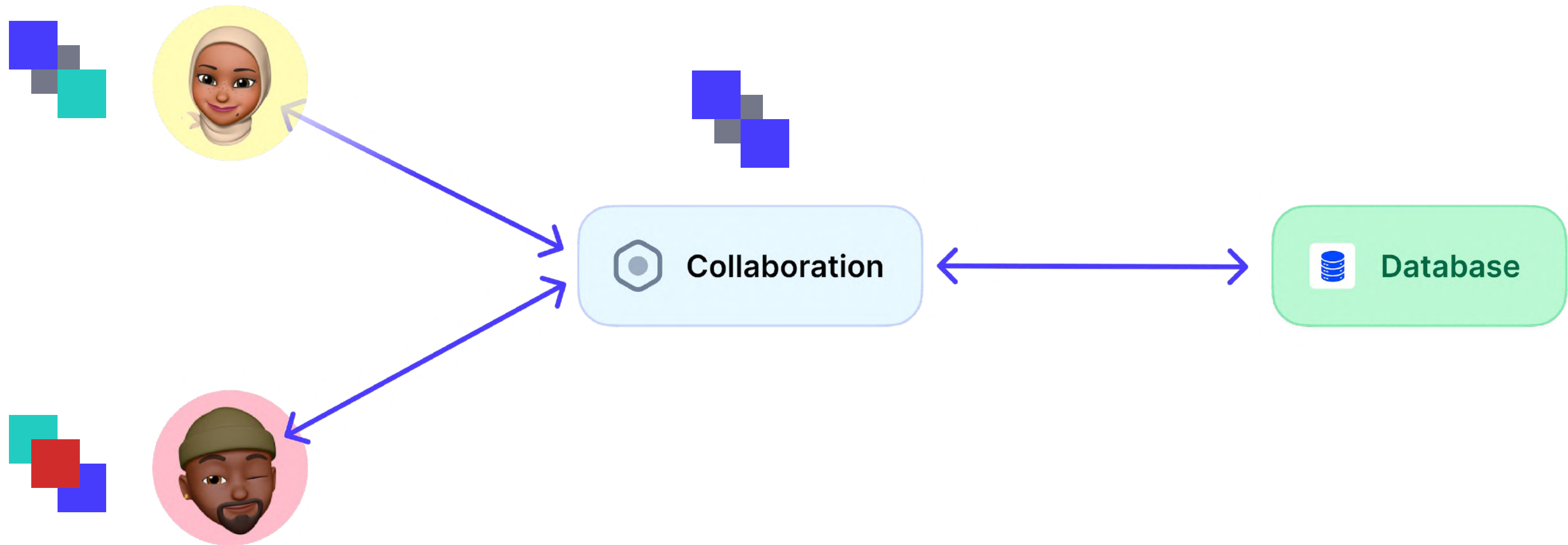
Awareness features out of the box

Large OSS community

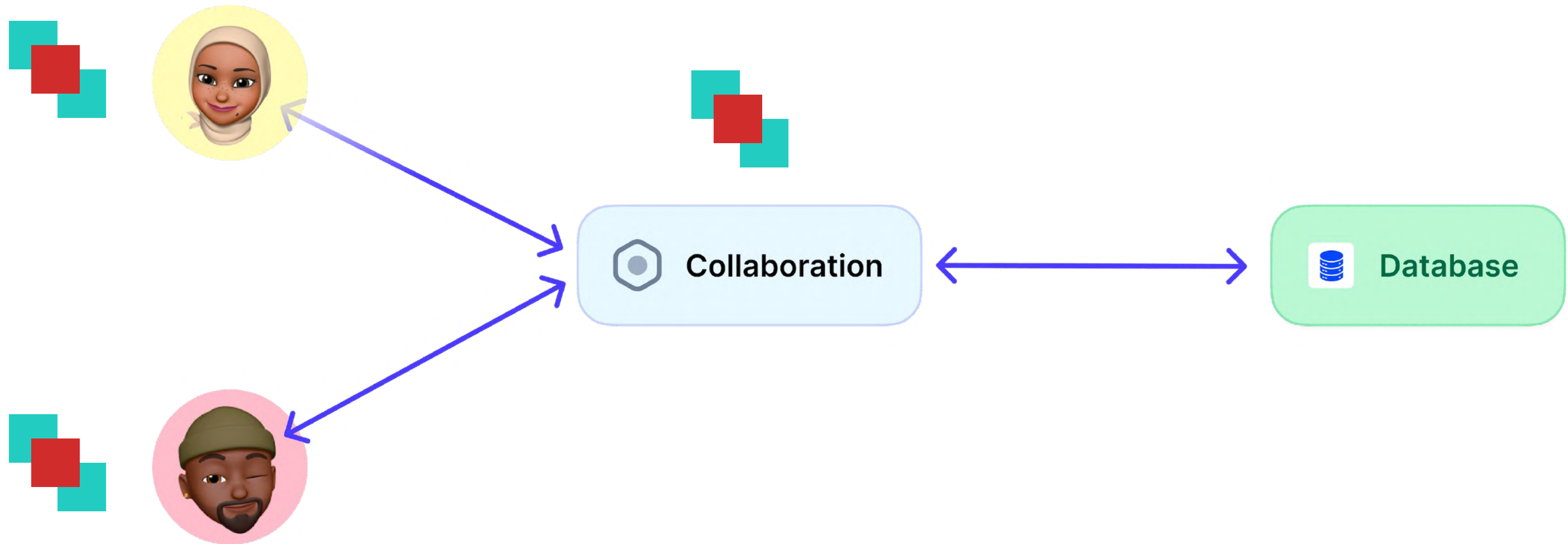
The easy way.



The easy way.

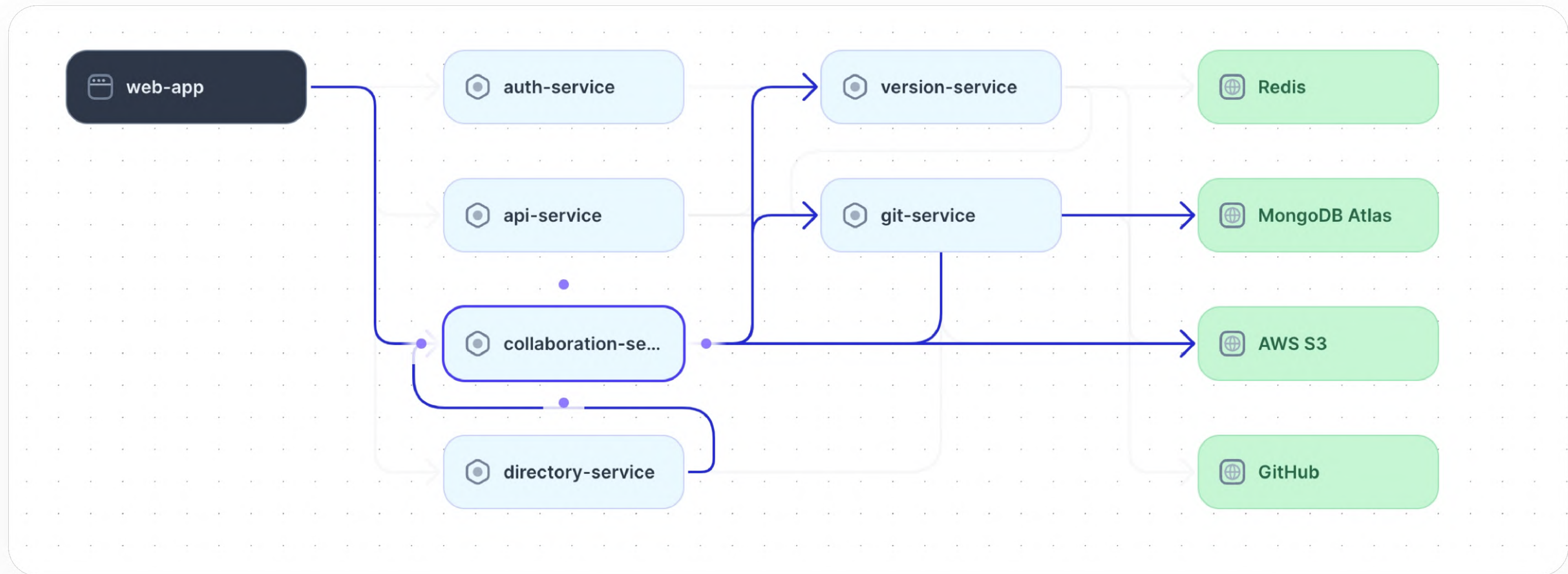


The easy way.



Here's what we do

with Y.js



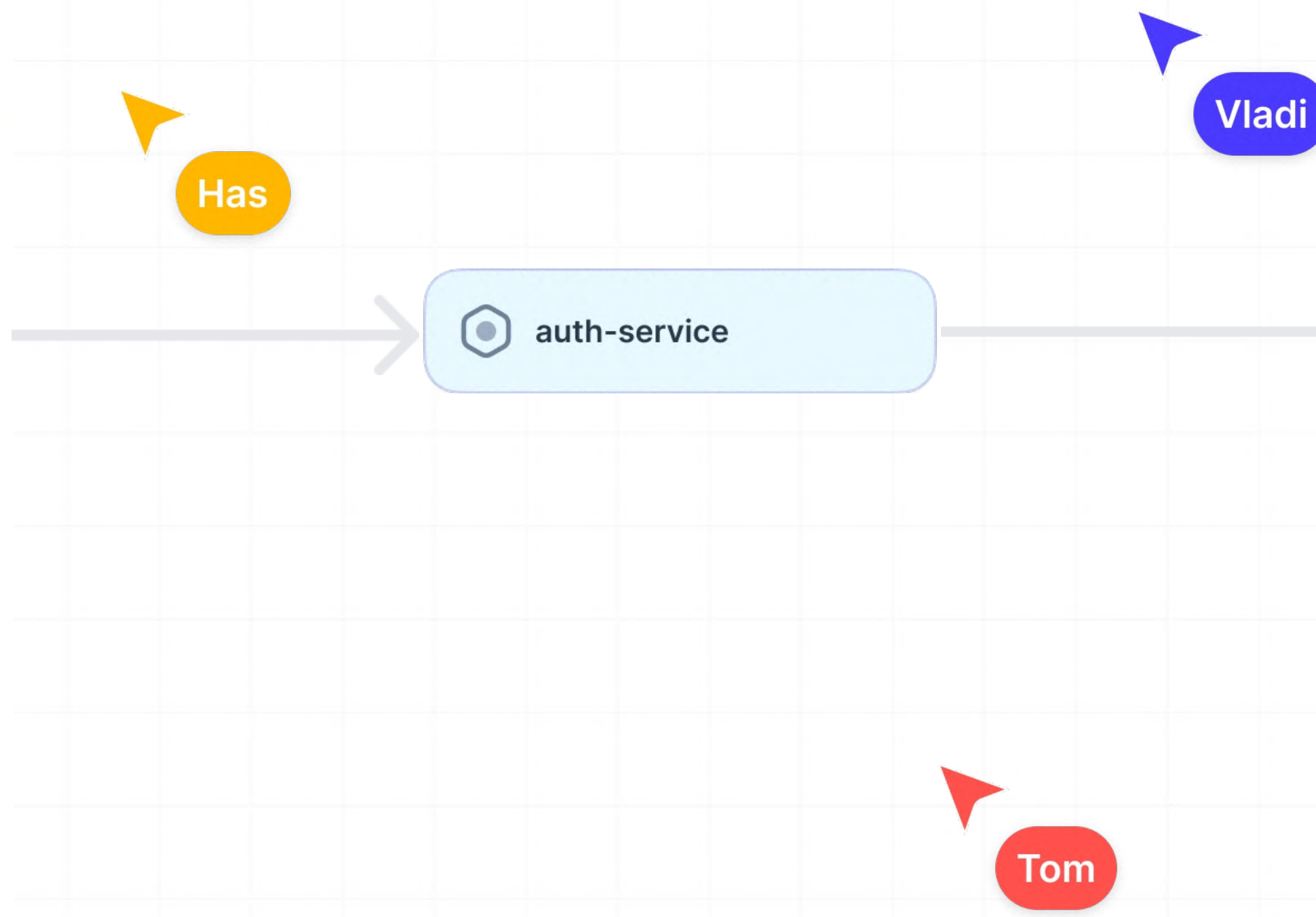


**Which real-time
collaborative features did we implement?**


```
1 // All of Y.js providers implement the awareness CRDT
2 const awareness = provider.awareness
3
4 // Set info about current user
5 awareness.setLocalStateField('user', {
6   id: '1',
7   name: 'Vladi Stevanovic',
8   avatar: '#linkToAvatar',
9 })
10
11 // Get list of all connected users
12 // List can include duplicates if app allows multiple connections per user
13 const users = Array.from(provider.awareness.getStates().values())
14   .map((state: any) => state.user)
```



```
1 // All of yjs providers implement the awareness CRDT
2 const awareness = provider.awareness
3
4 // Update user's "cursor" field to propagate relevant curso
5 awareness.setLocalStateField('cursor', {
6   x: '100',
7   y: '100',
8 })
9
10 // Event is called on any changes in the awareness states
11 awareness.on('change', (
12   changes: {
13     added: Array,
14     updated: Array,
15     removed: Array
16   }) => {
17   // Users' cursors and in-app locations can be redrawn
18   // using information from awareness.getStates()
19 })
20
21
22
23
24
25
26
27
28
29
```



STATE SYNCHRONIZATION

One
Live updates

Two
Co-editing

Three
Undo / Redo

Try Pitch

The screenshot shows a web application interface with a dark theme. At the top, there's a navigation bar with a star icon, a 'web-app' label, and a 'v2/new-feature Draft' indicator. Below the navigation bar, there are three tabs: 'readme', 'file', and 'file'. The main content area displays a chat conversation. The chat starts with a heading 'What is Next.js?'. The first message is from a user with a profile picture, stating: 'Next.js is a framework for building web applications. With Next.js, you can build user interfaces using React components. Then, Next.js provides additional structure, features, and optimizations for your application. Under the hood, Next.js also abstracts and automatically configures tooling for you, like bundling, compiling, and more. This allows you to focus on building your application instead of spending time setting up tooling.' The second message is from another user with a profile picture, stating: 'Whether you're an individual developer or part of a larger team, Next.js can help you build interactive, dynamic, and fast web applications.' Below the second message, there is a placeholder text: 'Enter text or type "/" for commands'. On the left side of the chat area, there is a vertical sidebar with several icons representing different views or actions.


```

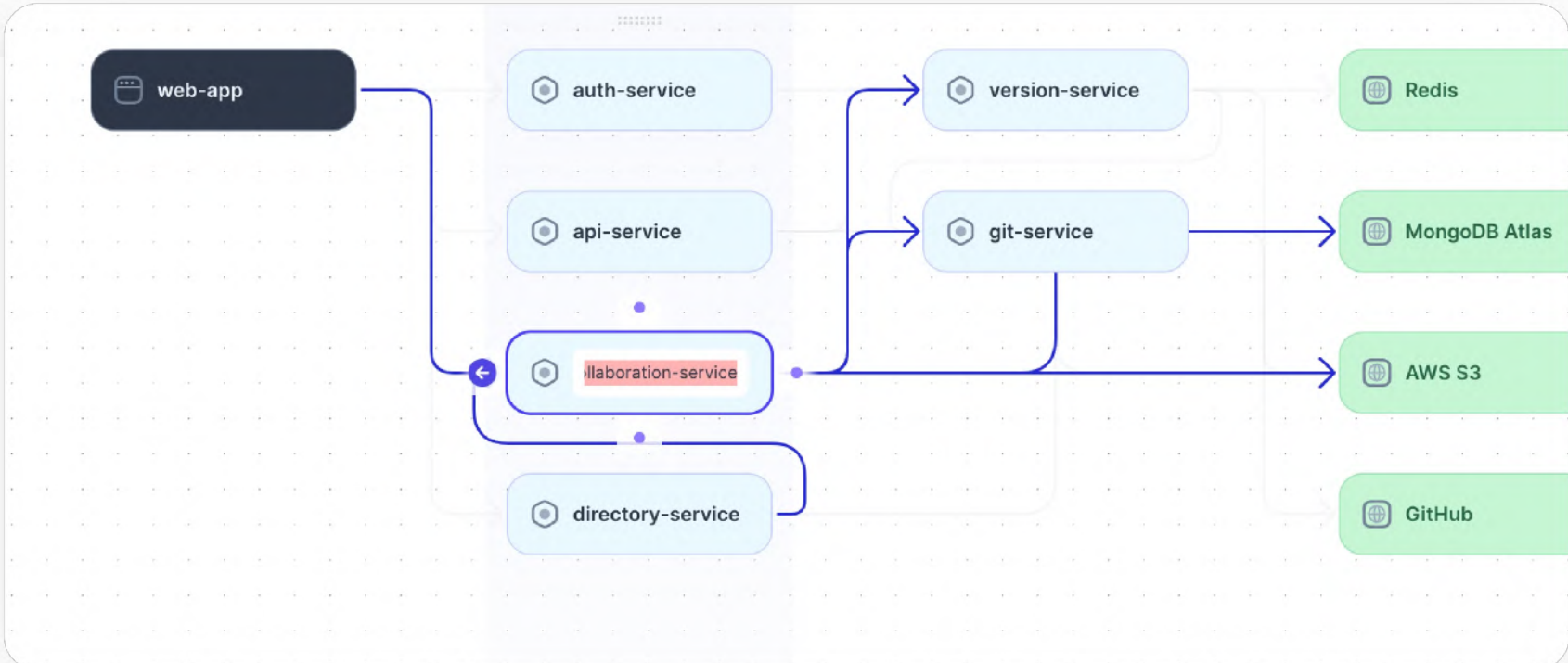
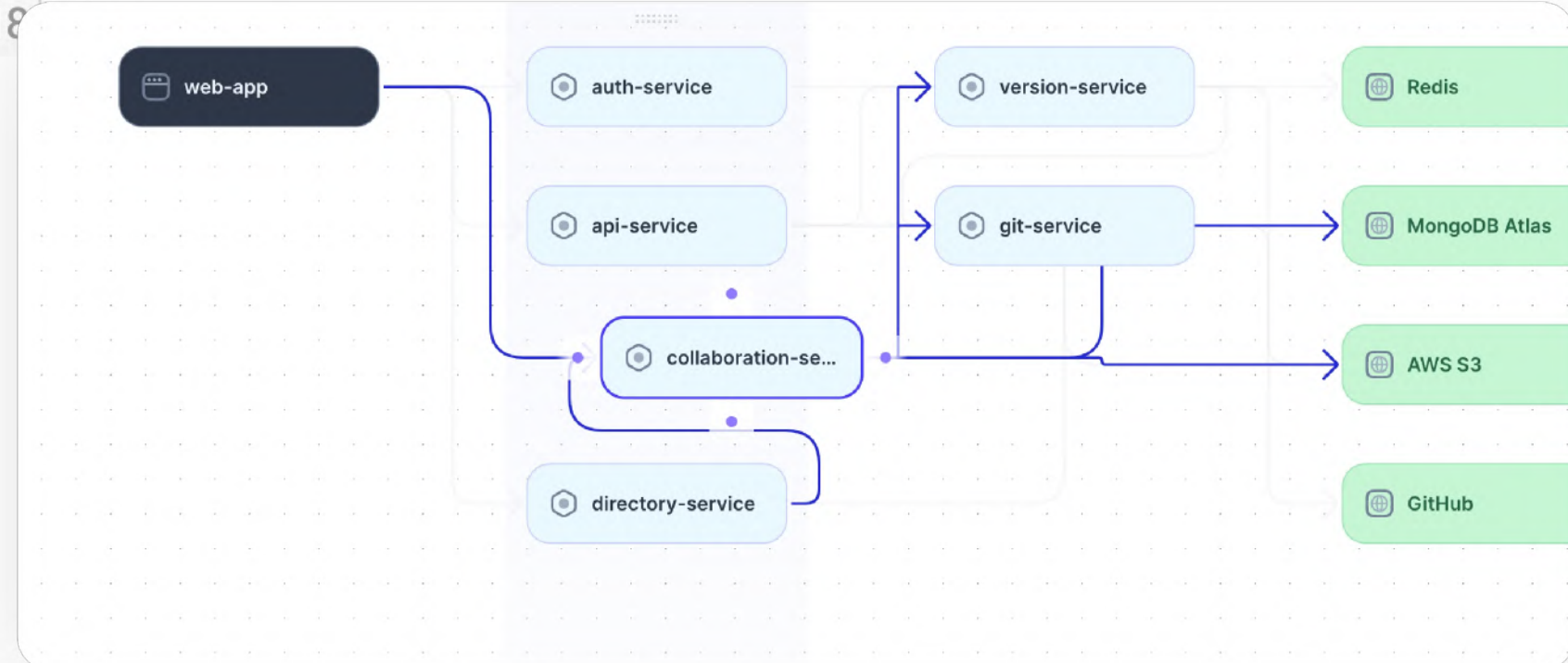
1 // ### Client 1
2 const doc = new Y.Doc()
3 const wsProvider = new WebSocketProvider(
4   'ws://localhost:1234', 'room', doc)
5
6 const nodeMap = new Y.Map()
7 nodeMap.name = 'Node name'
8 nodeMap.position = { x: 0, y: 0 }
9
10 doc.getMap('nodes').set('node-id', nodeMap)
11
12 wsProvider.on('sync', (isSynced) => {
13   if (!isSynced) return
14   doc.getMap('nodes')
15     .get('node-id')
16     .set('position', { x: 100, y: 100 })
17 })
18

```

```

19 // ### Client 2
20 const doc = new Y.Doc()
21 const wsProvider = new WebSocketProvider(
22   'ws://localhost:1234', 'room', doc)
23
24 wsProvider.on('sync', (isSynced) => {
25   if (!isSynced) return
26   doc.getMap('nodes')
27     .get('node-id')
28     .set('name', 'New name')
29 })
30
31 // both docs will include summed up changes from both clients
32
33
34

```




```

1 // Socket.IO example
2
3 enum CommentsEvents {
4   COMMENT_CREATE='comment/create'
5 }
6
7
8 const io = new Server({/*options*/});
9
10 io.on('connection', (socket) => {
11   const onCommentCreate = (data, callback) => {
12     try {
13       // validate and write comment to database
14       const comment = CommentService.createComment(data)
15       callback(comment, undefined)
16
17       // notify other users about change
18       socket.broadcast.emit(CommentsEvents.COMMENT_CREATE, comment)
19     } catch (err) {
20       // notify user that request failed
21       callback(undefined, err)
22     }
23   }
24
25   socket.on(CommentsEvents.COMMENT_CREATE, onCommentCreate)
26 })
27

```

Try Pitch

Comments in this file

web-platform

Pedro Pessoa 1h ago

Let's review this together in a while.

Pedro is typing...

web-platform

Pedro Pessoa 1h ago

I'm not that sure about this one, since we'll be using redis heavily and the whole architecture might become unscalable for the time being. I'm going to ping you later on Slack, maybe huddle this one out.

Thanks @Mike

8 replies

web-platform

Pedro Pessoa 1h ago

Let's review this together in a while.

What we learned ✨

Learning #1 - How to keep track of order

asset-service

auth-service

web-api-service

search-service

messaging-service

1

2

3

4

What we learned ✨

Learning #1 - How to keep track of order

asset-service

1
auth-service

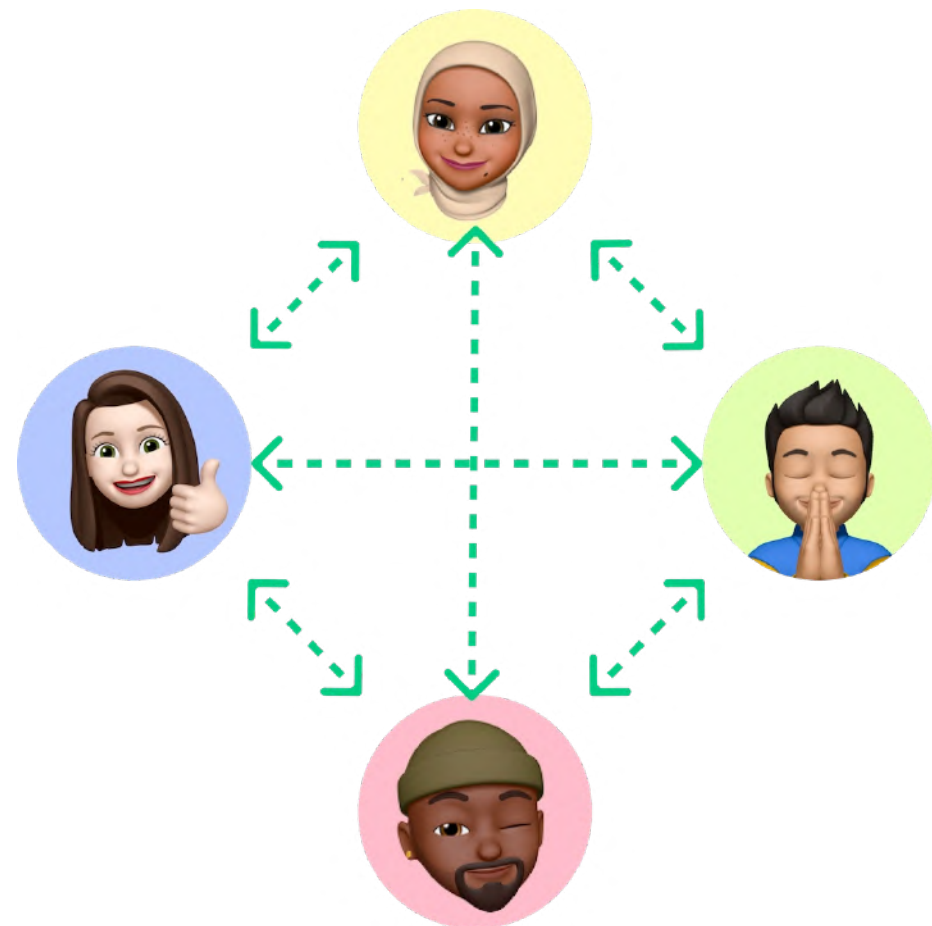
2
web-api-service

3
search-service

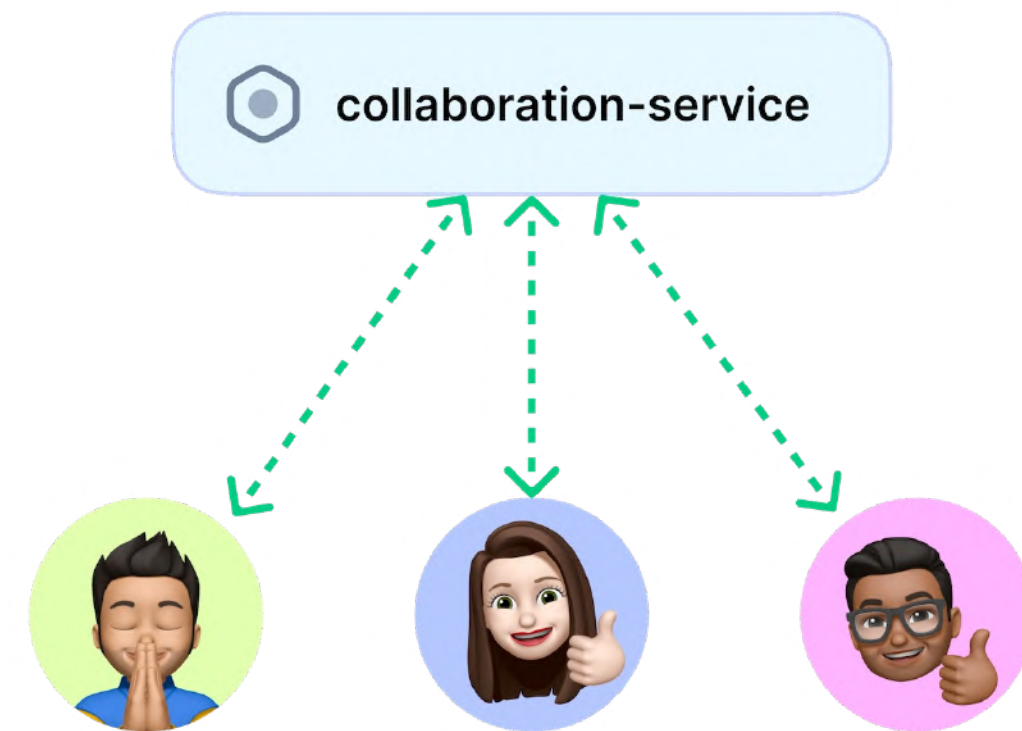
4
messaging-service

What we learned ✨

Learning #2 - Architecture options



peer-to-peer

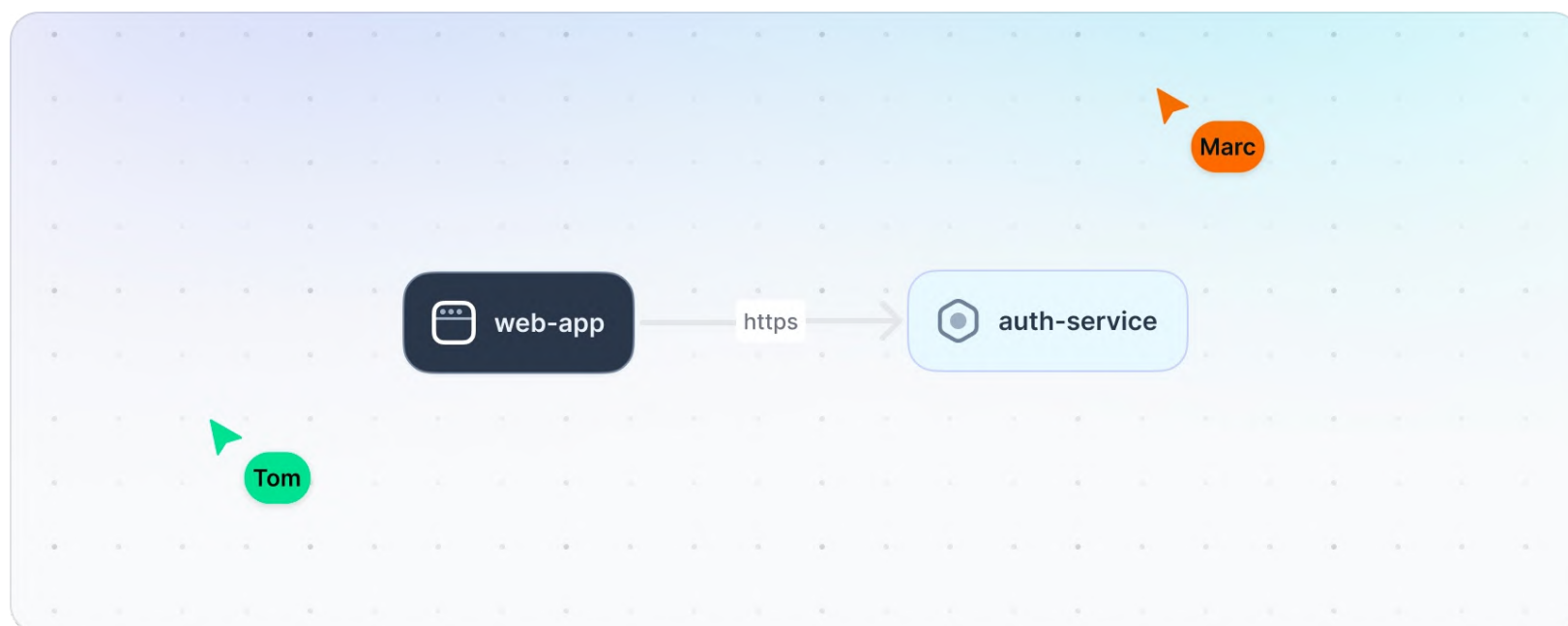


central service

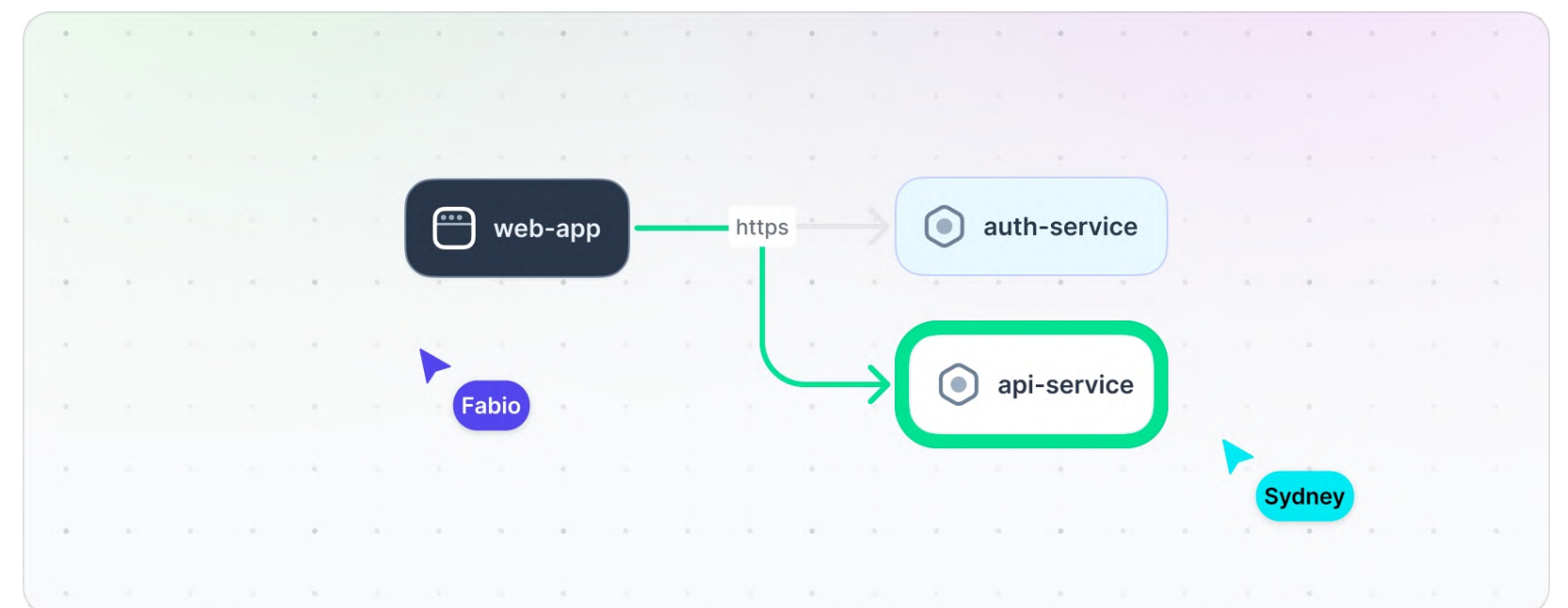
What we learned ✨

Learning #3 - Scope of collaboration

🔍 feature/ai-assist Draft



🔍 feature/slack-integration Draft



What we learned ✨

Learning #4 - Changes to dependent views


web-api

Mike

+ Changes + 4 ▲ 12

- web-platform
- web-api
- web-api
- requirements
- v2-requirements
- Stripe
- schema-users
- schema-companies
- schema-types

Try Pitch



web-api


Multiplayer

About | Comments | History


Information

Name	Mike	Type	Service Type
web-api		Service	Cloud Provider
Visibility	Owner	Version	
Private	@multiplayer	@latest	

TBD Client API Write more tags..



web-api

 web-api

A better user experience.



Shorter production cycles.



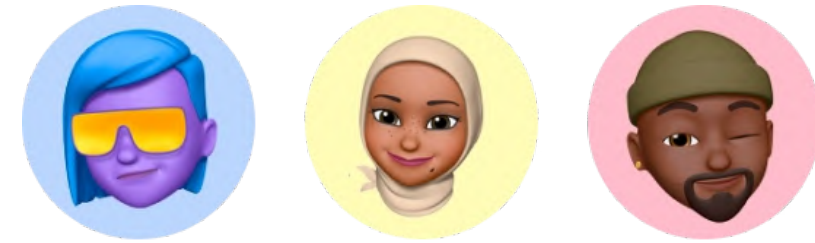
Increased business revenue.



Final notes



Don't reinvent the wheel.



Embrace collaboration from the start.



Thank you

Join our waitlist

multiplayer.app

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

[Create a presentation \(It's free\)](#)