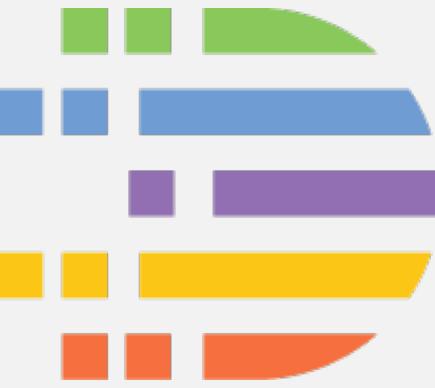




# Dashbit

@josevalim



# Dashbit

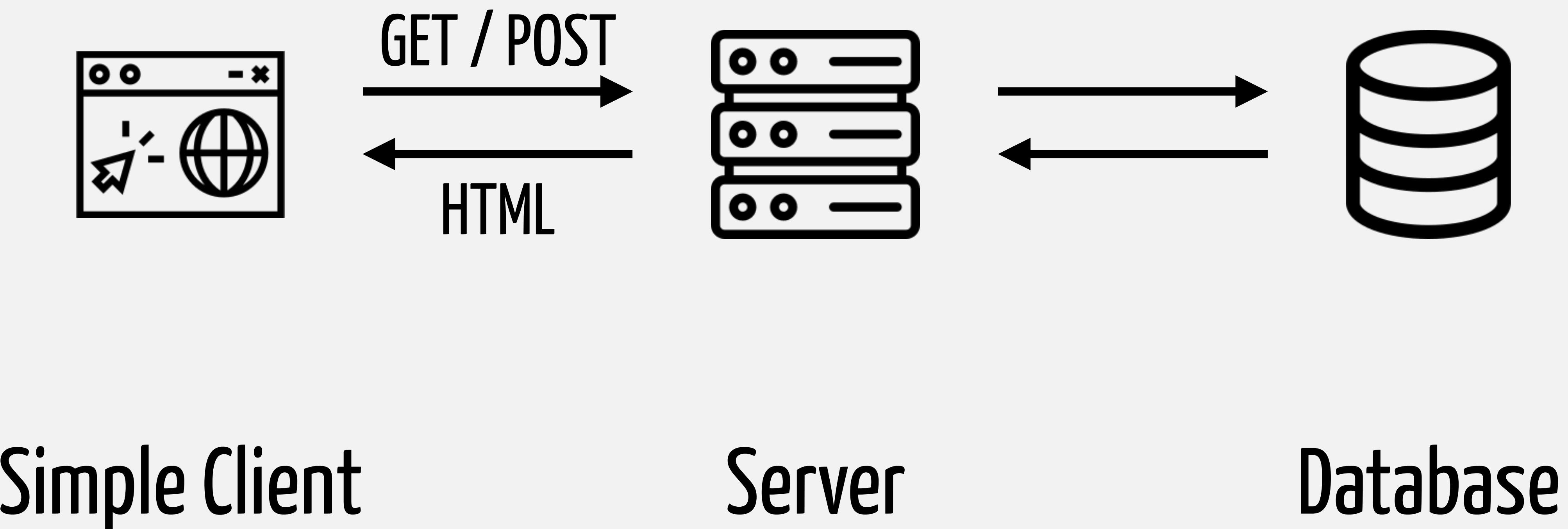
- Elixir
- Ecto / Postgrex / MyXQL
- Phoenix / LiveView
- Plug
- Broadway
- Livebook
- Req
- Nimble\*

# The projects that never were (part 1)



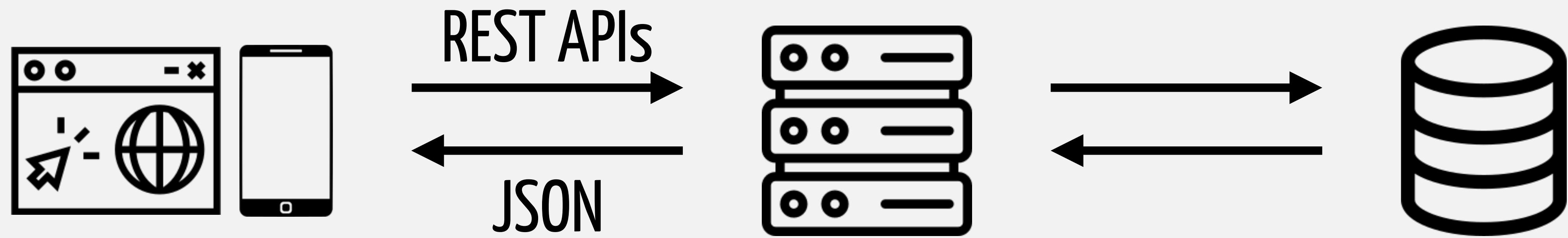
**Dashbit**

# Web apps: 20 years ago



A FEW  
MOMENTS LATER

# Web apps: ~15 years ago



Rich Clients / SPAs

Server

Database

# SPAs + REST APIs

- Batching: 

Welcome,  
John Doe

GENERAL

Home

Dashboard

Dashboard2

Dashboard3

Forms

UI Elements

Tables

Data Presentation

Layouts

LIVE ON

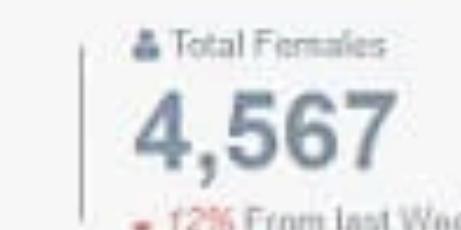
Additional Pages

Extras

Multilevel Menu

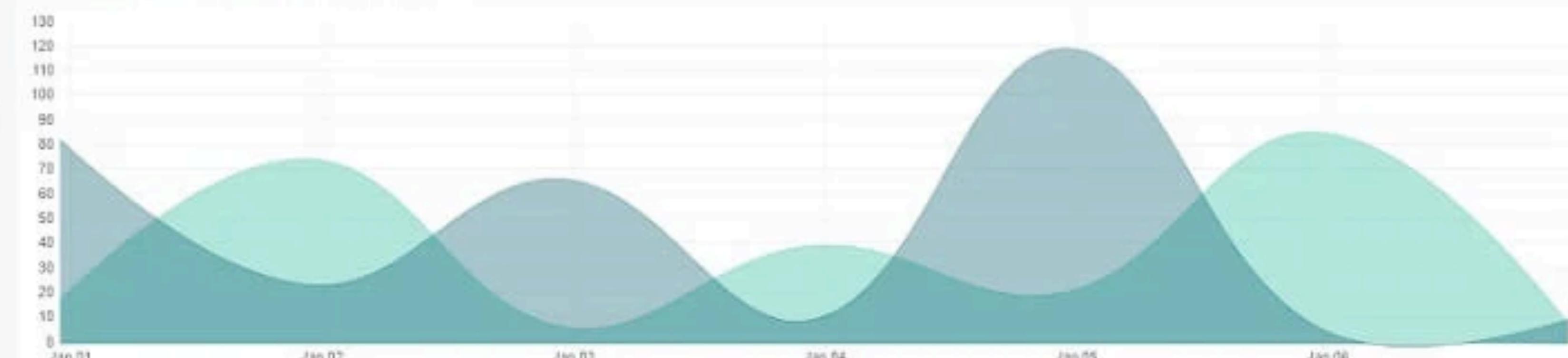
Landing Page

Coming Soon



## Network Activities Graph title sub-title

November 15, 2017 - December 14, 2017



## Top Campaign Performance

Facebook Campaign

Twitter Campaign

Conventional Media

Bill boards

## App Versions

App Usage across versions

0.1.5.2		123k
0.1.5.3		53k
0.1.5.4		23k
0.1.5.5		3k
0.1.5.6		1k

## Device Usage

Top 5



Device	Progress
iOS	30%
Android	10%
Blackberry	20%
Symbian	15%
Others	30%

## Quick Settings

Settings

Subscription

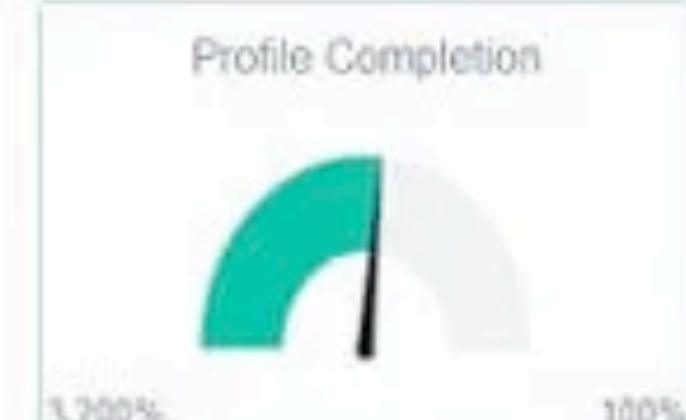
Auto Renewal

Achievements

Auto Renewal

Achievements

Logout



## Recent Activities Sessions

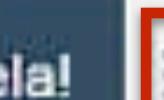
- Who Needs Sundance When You've Got Crowdfunding?  
12 hours ago by Jane Smith  
Film festivals used to be do-or-die moments for movie makers. They were where you met the producers that could fund your project, and if the buyers liked your flick, they'd pay to Fast-forward and... Read More

## Visitors location geo-presentation

125.7k Views from 60 countries

United States	33%
France	27%



Welcome,  
John Doe

## GENERAL

- Home
- Dashboard
- Dashboard2
- Dashboard3
- Forms
- UI Elements
- Tables
- Data Presentation

- Layouts
- LIVE ON
- Additional Pages
- Extras
- Multilevel Menu
- Landing Page Coding Soon



Network Activities Graph title sub-title

\$http.get(' /api/network' )

November 15, 2017 - December 14, 2017



## Top Campaign Performance

Facebook Campaign

Twitter Campaign

Conventional Media

Bill boards

## App Versions

App Usage across versions

0.1.5.2		123k
0.1.5.3		53k
0.1.5.4		23k
0.1.5.5		3k
0.1.5.6		1k

## Device Usage

Top 5



Device Progress

IOS	30%
Android	10%
Blackberry	20%
Symbian	15%
Others	30%

## Quick Settings

Settings

Subscription

Auto Renewal

Achievements

Auto Renewal

Achievements

Logout

## Profile Completion



## Recent Activities Sessions

Who Needs Sundance When You've Got Crowdfunding?

13 hours ago by Jane Smith

Film festivals used to be do-or-die moments for movie makers. They were where you met the producers that could fund your project, and if the buyers liked your flick, they'd pay to Fast-forward and... [Read More](#)

## Visitors location geo-presentation

125.7k Views from 60 countries

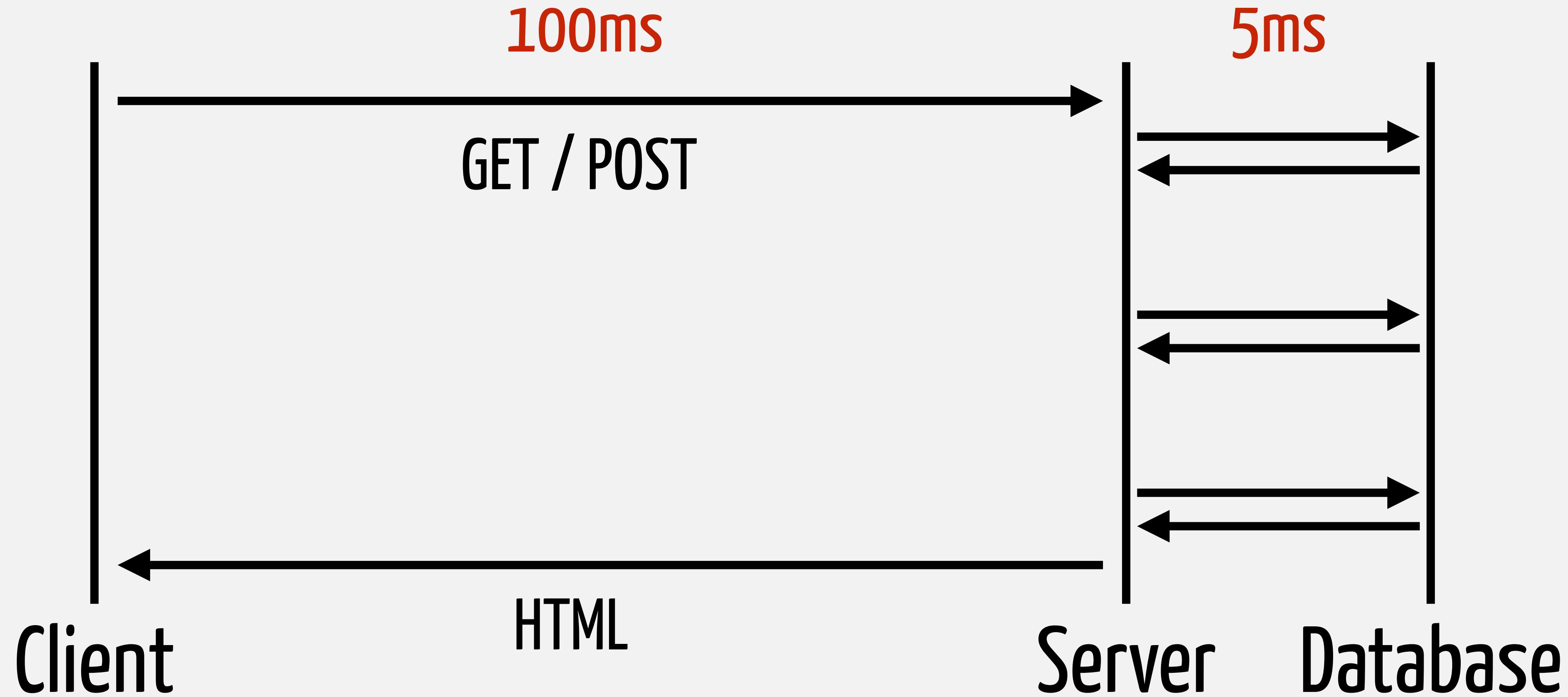
United States 33%

France 27%

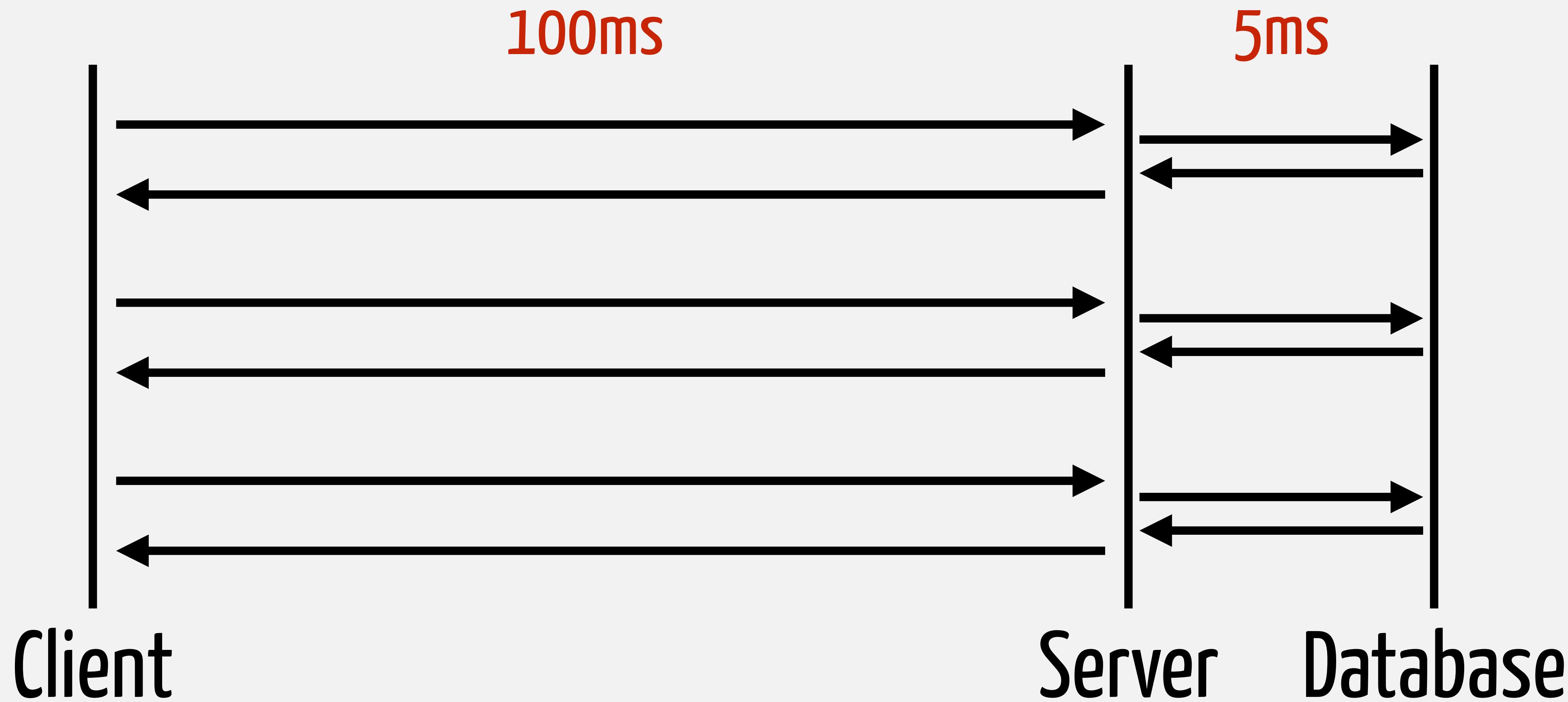
\$http.get(' /api/location' )



# Batching: server-side rendering



# Batching: SPAs + REST APIs



# SPAs + REST APIs

- Batching: 
- Reduced payloads: 

# SPAs + REST APIs: Payloads

```
GET /posts/123
{
  "id": 123,
  "title": "Understanding Single Page Applications",
  "author": "Jane Doe",
  "date_published": "2024-08-22T14:30:00Z",
  "tags": ["SPA", "Web Development"],
  "body": "[...]",
  "comments": [
    ...
  ]
}
```

# SPAs + REST APIs: Payloads

```
GET /posts/123/preview
```

```
{  
  "id": 123,  
  "title": "Understanding Single Page Applications",  
  "author": "Jane Doe",  
  "date_published": "2024-08-22T14:30:00Z",  
  "tags": ["SPA", "Web Development"],  
}
```

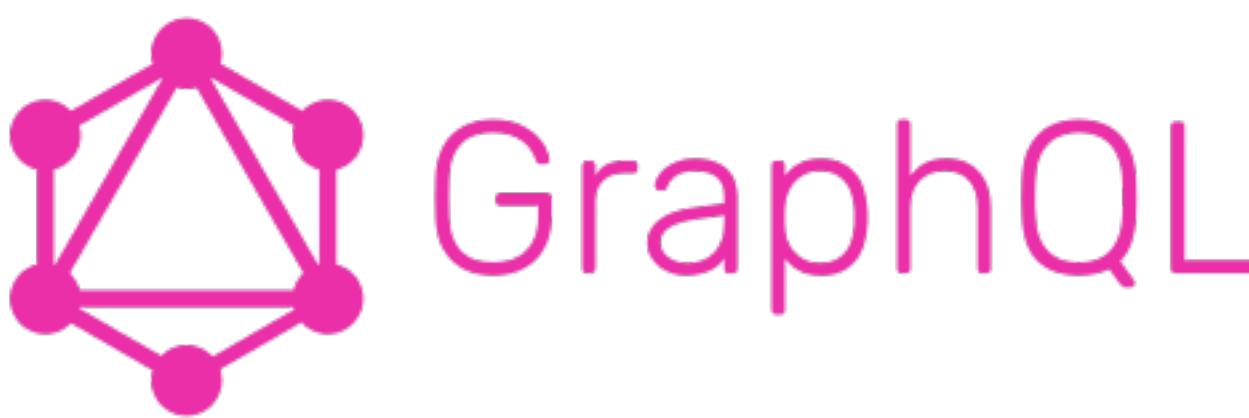
# SPAs + REST APIs: Payloads

```
GET /posts/123?exclude=body, comments
{
  "id": 123,
  "title": "Understanding Single Page Applications",
  "author": "Jane Doe",
  "date_published": "2024-08-22T14:30:00Z",
  "tags": ["SPA", "Web Development"],
}
```

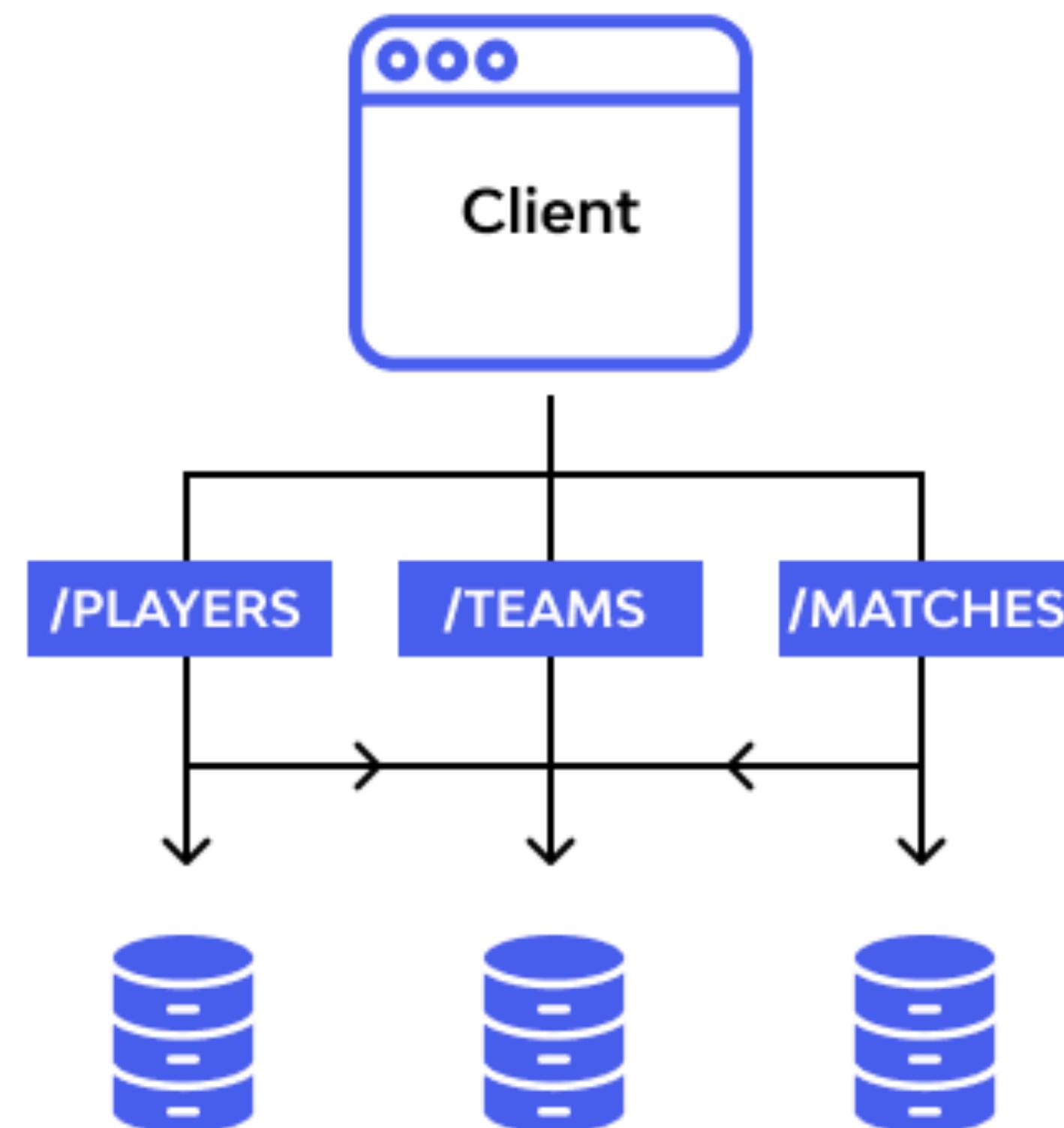
# SPAs + REST APIs

- Batching: 
- Reduced payloads: 
- Streaming: 

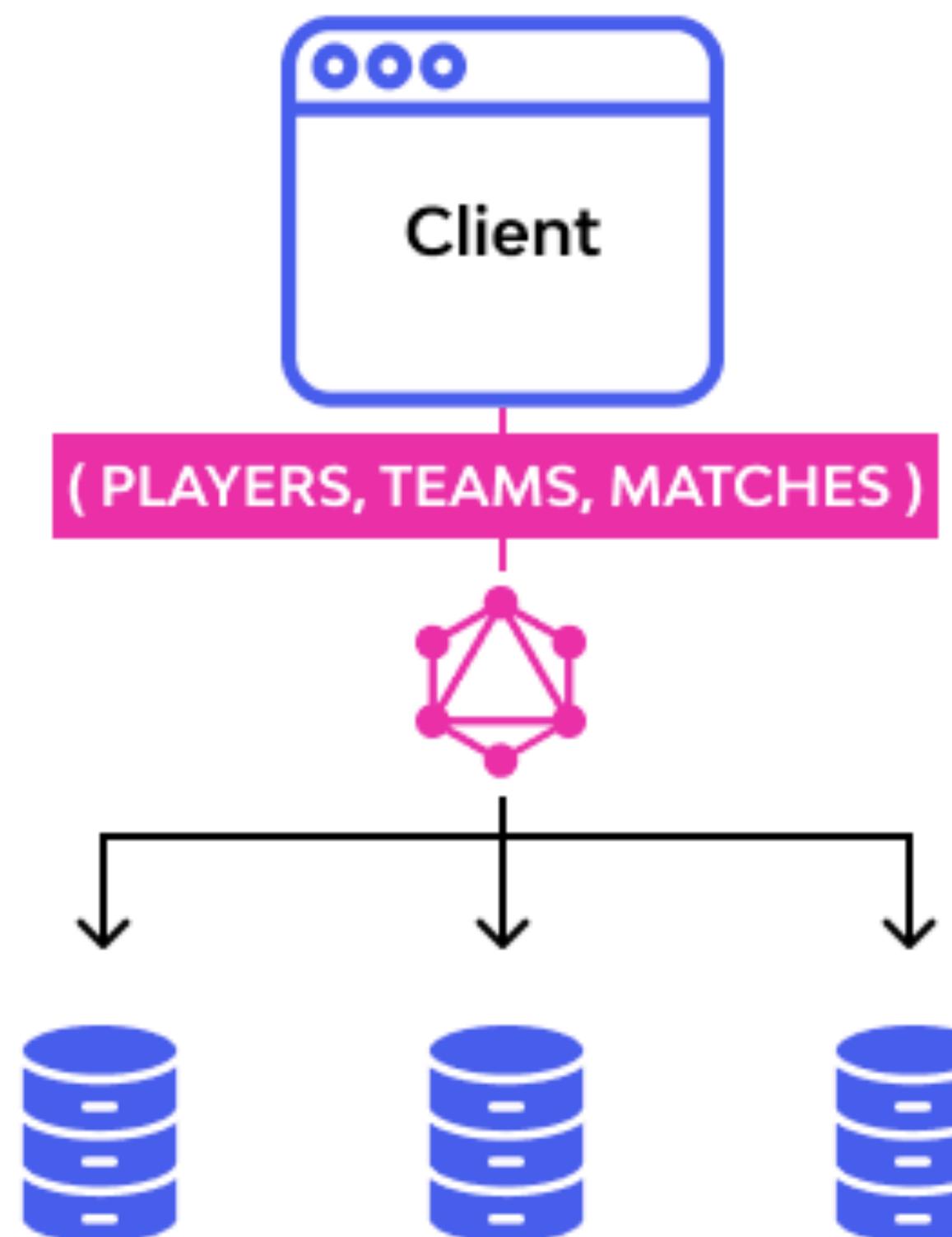
A FEW  
MOMENTS LATER



Rest API



GraphQL API

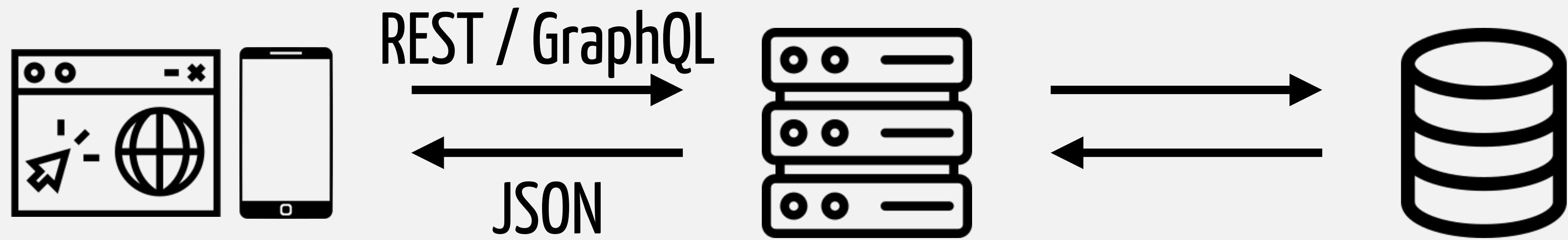


# SPAs + GraphQL

- Batching: 
- Reduced payloads: 
- Streaming: 

**However, every other week,  
it seems to be a new article  
bashing GraphQL...**

# Web apps: ~10 years ago



Rich Clients / SPAs

Server

Database

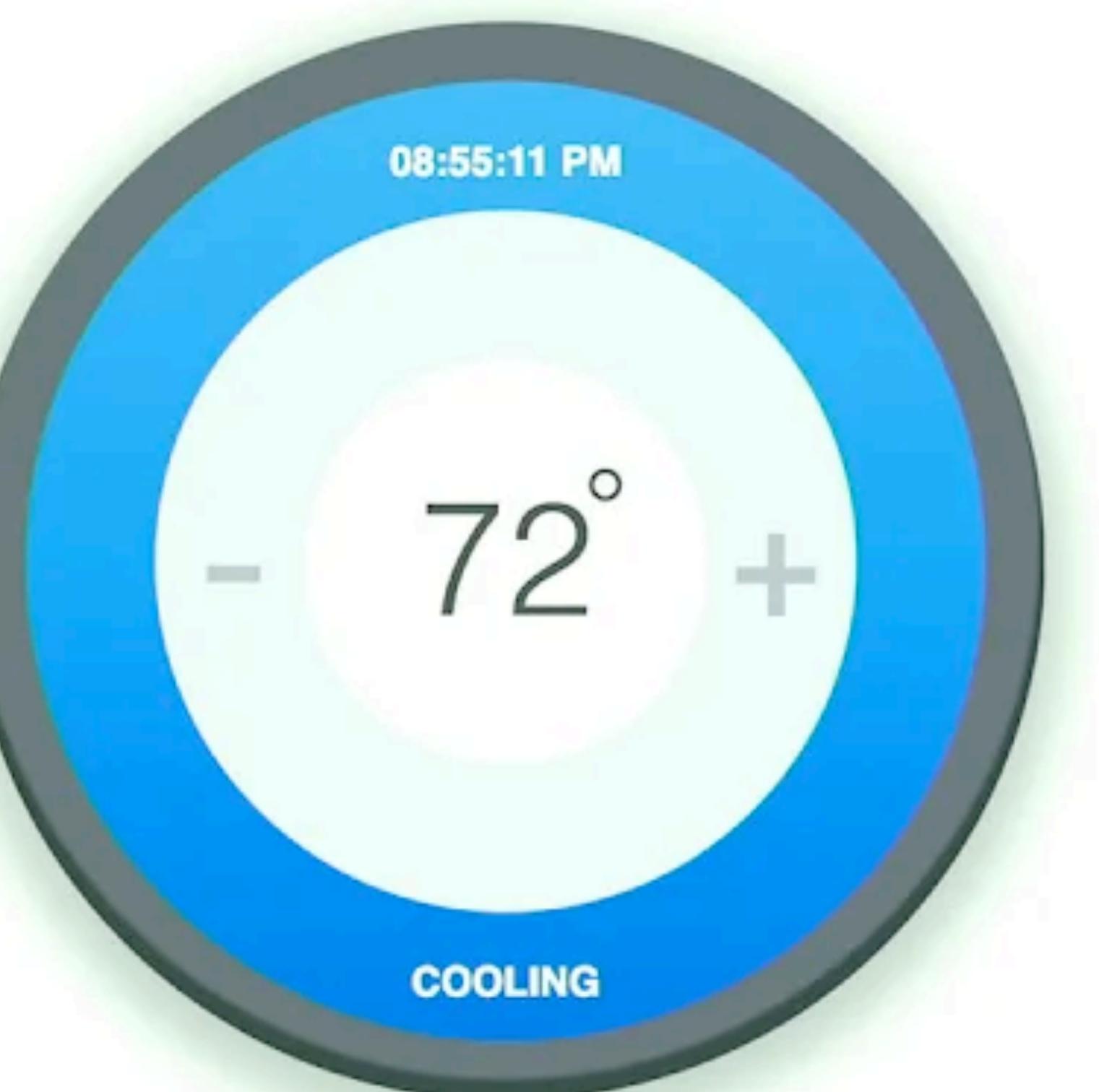
# What can we do better?

- No more serialization
- Squash the layers

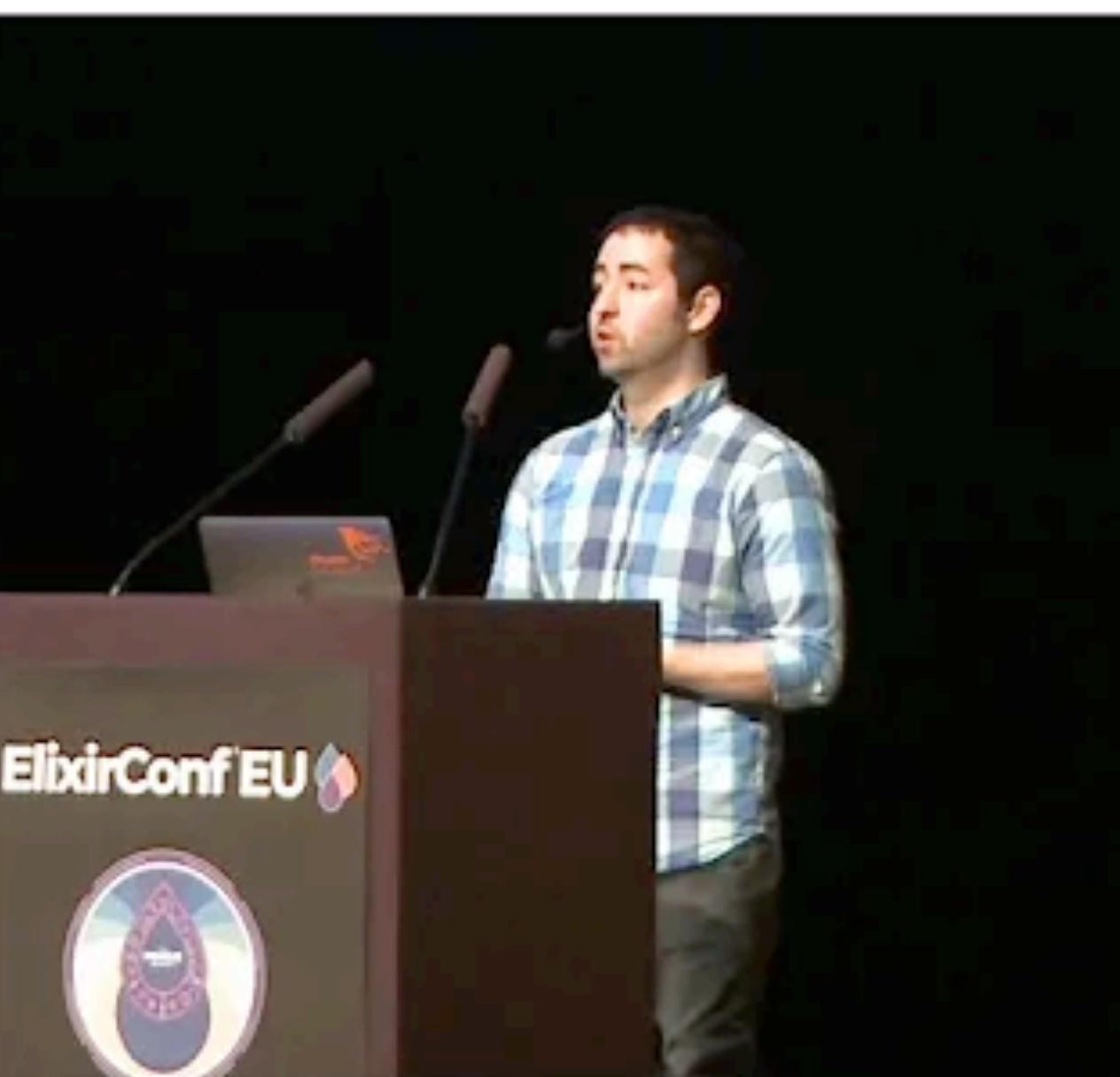
```
<div>
  <h1><%= @val %></h1>
  {<a href="#">-</a>
   <a href="#">+</a>
   <span><%= @mode %></span>
</div>
```

incButton.onClick(() => ...ajax/channels...)

- **Server**
  - Define routes
  - Create Controller / Channel
  - Define JSON/payload contracts
- **Client**
  - Handle events and syncing state
  - Handle client actions
  - Handle error recovery
  - *Navigate the JS library labyrinth*



ElixirConf EU



Keynote: Phoenix LiveView - Interactive, Real-Time apps without JavaScript  
Chris McCord



**Build in a weekend.  
Scale to millions.**

app.supabase.xyz

Copple Staging > staging-testing

countries X

Filter Sort 1 + Add Row

	id	name	iso2
1	Bonaire, Sint Eustatius and Saba	BQ	
2	Curaçao	CW	
3	Guernsey	GG	
4	Isle of Man	IM	
5	Jersey	JE	
6	Åland Islands	AX	
7	Montenegro	ME	
8	Saint Barthélemy	BL	
9	Saint Martin (French part)	MF	
10	Serbia	RS	
11	Sint Maarten (Dutch part)	SX	
12	South Sudan	SS	
13	Timor-Leste	TL	
14	American Samoa	AS	
15	Andorra	AD	

1 - 100 of 249 records

**SQUASH  
THE  
LAYERS**

The mural is set against a backdrop of a brick building with arched windows and doors. A ladder and some debris are visible on the ground in front of the wall.

# Modern server-centric architectures

- Server-rendered HTML
- Server-rendered Props
- Server-rendered Diffs

# Modern server-centric criteria

- Batching:   
- Reduced payloads:   
- Streaming:   
- Handling of client / server state:   

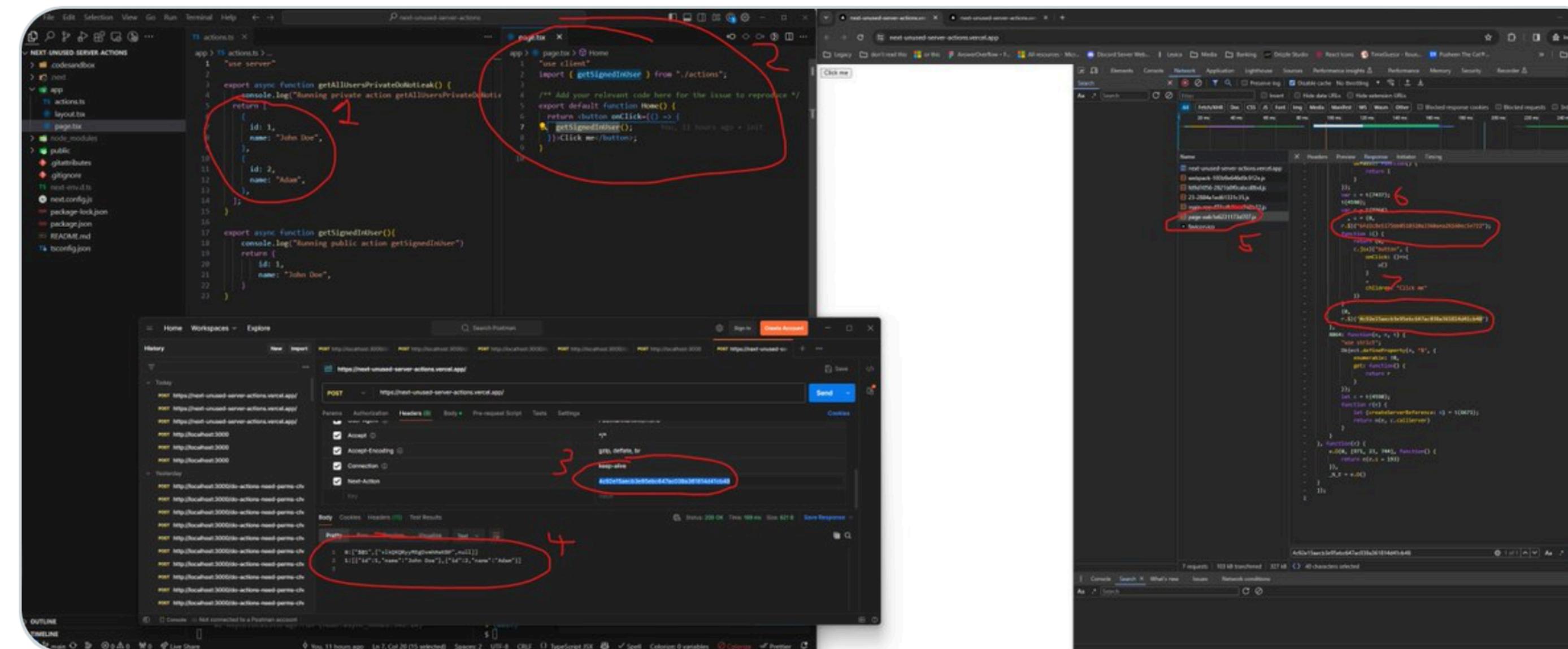
Rhys ✅  
@RhysSullivan

...

If you care about security for your NextJS app, **stop using top level "use server"** - it's way too easy to leak data

Top level "use server" creates endpoints for all exported functions, even if they are never used on the client

One accidental export can cause a ton of damage



9:27 AM · Jul 2, 2024 · 283.3K Views

# Sources of client state (pt1)

- Cookies
- Local storage, session storage
- URL anchors
- Browser history

# Sources of client state (pt2)

- Inputs, forms
- Scroll position
- Focused elements
- Drag and drop
- etc

# Mind the latency

foo@example.com

Client

Server

already taken

foo@example.com.br



# Modern **server-centric** architectures

<https://dashbit.co/blog/web-apps-have-client-and-server-state>

	Squash	Batching	Reduced payloads	Streaming	Client/server state
<b>Server Rendered HTML</b> HTMX, Hotwire					
<b>Server Rendered Props</b> Inertia.js, Remix					
<b>Server Rendered Diffs</b> LiveView					

[Subscribe](#)[Product News](#)[Speed & Reliability](#)[Security](#)[Zero Trust](#)[Developers](#)[AI](#)[Policy](#)[Partners](#)[Life at Cloudflare](#)

# Cloudflare acquires PartyKit to allow developers to build real-time multi-user applications

2024-04-05



Sunil Pai



Rita Kozlov

4 min read

We're thrilled to announce that PartyKit, an open source platform for deploying real-time, collaborative, multiplayer applications, is now a part of Cloudflare. This acquisition marks a significant milestone in our journey to redefine the boundaries of serverless computing, making it more dynamic, interactive, and, importantly, stateful.

## Defining the future of serverless compute around state

Building real-time applications on the web have always been difficult. Not only is it a distributed systems problem, but you need to provision and manage infrastructure, databases, and other services to maintain state across multiple clients. This complexity

"WebSockets have revolutionized how we think about bidirectional communication on the web.

Yet, the challenge has always been about scaling these interactions to millions without a hitch.

Cloudflare Workers step in as the hero [...]"

**"Building experiences with real-time components  
was previously only accessible to multi-billion  
dollar companies"**

**Phoenix + PostgreSQL  
is the one stop shop  
for front-end developers**



# Build single-page apps, **without** **building an API.**

Create **modern single-page React, Vue, and Svelte apps** using classic server-side routing.

Works with any backend – tuned for Laravel.

```
UsersController.php          Users.vue

class UsersController
{
    public function index()
    {
        $users = User::active()
            ->orderByName()
            ->get(['id', 'name',
                    ]);

        return Inertia::render('U
            'users' => $users
        ]);
    }
}
```

☰  Valian / live\_vue

🔍 Type / to search | + ⚙️ ⌂ ⌂ ⌂ ⌂ ⌂

Code Issues 2 Pull requests 2 Actions Security Insights

 live\_vue Public Watch 8 Fork 7 Star 180

☰  woutdp / live\_svelte

🔍 Type / to search | + ⚙️ ⌂ ⌂ ⌂ ⌂ ⌂

Code Issues 18 Pull requests 2 Discussions Actions Projects Security Insights

 live\_svelte Public Sponsor Watch 28 Fork 43 Star 1.1k

☰  launchscout / live\_state

🔍 Type / to search | + ⚙️ ⌂ ⌂ ⌂ ⌂ ⌂

Code Issues 2 Pull requests 1 Actions Projects Security Insights

 live\_state Public Watch 16 Fork 11 Star 194

☰ main ▾ 12 Branches 0 Tags Go to file t + <> Code About

Chris Nelson bumpety bump ✓ 9664f81 · 4 months ago 83 Commits

The hex package for the server side of live state

# Client-centric LiveView?

	Squash	Batching	Reduced payloads	Streaming	Client/server state
<b>Server Rendered Diffs</b> LiveView					

 Topics  My Posts Review Admin More Categories Elixir News Chat / Discussions  Questions / Help  Suggestions & Propo...  Phoenix Forum  Ash Forum  Nerves Forum  Elixir Framework For...  Wikis & Polls  Learning Resources  Libraries  Elixir Jobs  Events List  Forum Announcements  All categories Tags community 

## Sequin - Elixir + Postgres for a feature-rich message queue

[Chat / Discussions](#)[postgres](#)[tools](#)

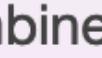
acco

1  Jul 19

Jul 19th

1 of 7  
Jul 19th

Hey everyone,

Excited to share a new open-source project built with Elixir. [Sequin](#)  combines Elixir and Postgres into a feature-rich message queue.

We (the maintainers) were searching for the goldilocks message stream/queue. Kafka's great when you need tremendous scale, but the trade-off is that it has few features and is hard to debug/configure/operate. SQS also has limited features, and we don't like the dev/test story. RabbitMQ was closest to what we were seeking, but was the most unfamiliar to the team, and looked like it would be a black box.

Databases and messaging systems are at the heart of most applications. But having both adds complexity. We already know and love Postgres – can't it do both?

Teams pass on using Postgres for streaming/queuing use cases for two primary reasons:

1. They underestimate Postgres' ability to scale to their requirements/needs
2. Building a stream/queue on Postgres is DIY/open-ended

Getting DIY right requires some diligence to build a performant system that doesn't drop messages (MVCC can work against you). We also wanted a ton of features from other streams/queues that weren't available off-the-shelf, namely:

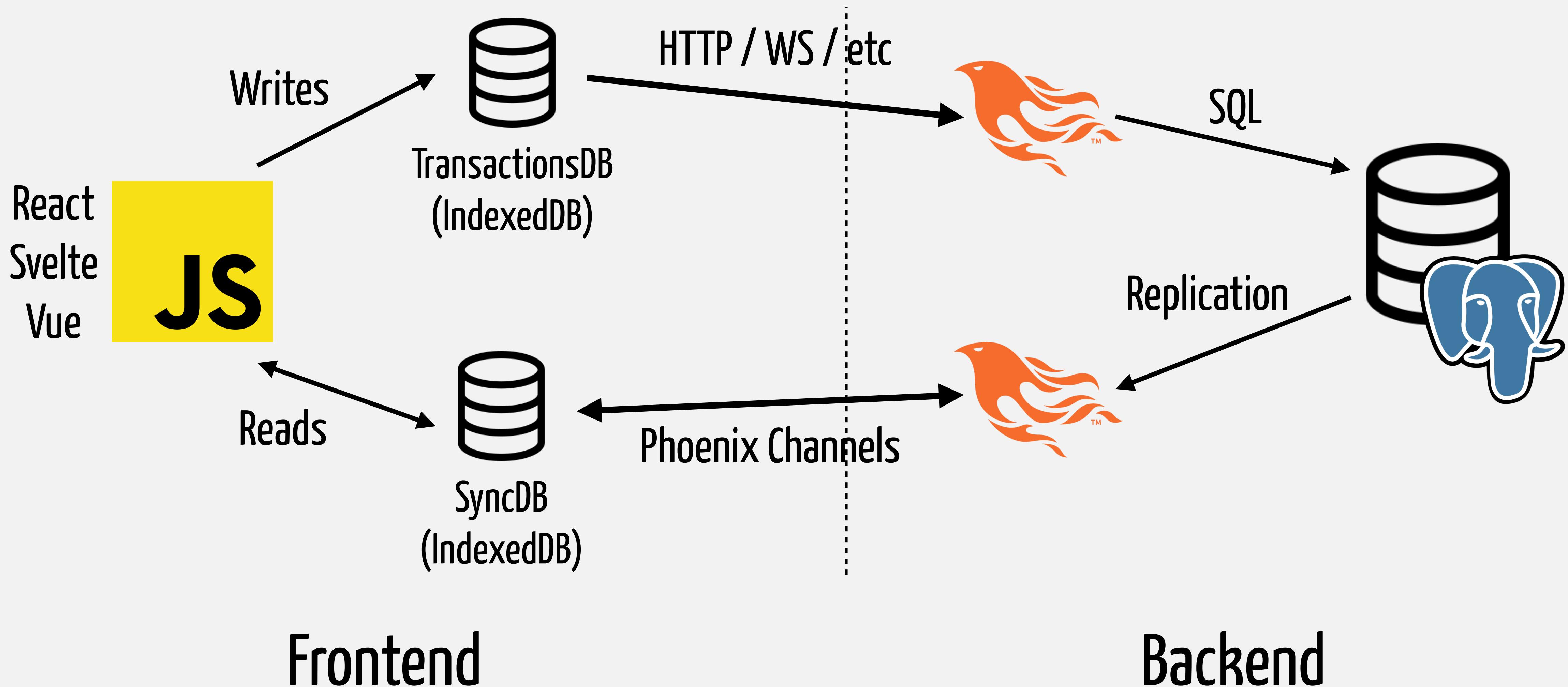
- message routing with subjects

24d ago



# Demo time!

# "Phoenix Sync"



# "Phoenix Sync"

- Data sync with transactional consistency  
(no serialization, no endpoints, no queries)
- Offline mode and realtime (streaming)
- Phoenix and stock PostgreSQL only

# Phoenix and stock PostgreSQL

- No additional services
- No additional copies of the data
- No additional programming models
- Only requires one column per table (+ soft-deletes)
- Thanks to PostgreSQL, Phoenix, and Erlang VM

# Don't drink the kool aid

- It is just a proof of concept:  
[github.com/josevalim-sync](https://github.com/josevalim-sync) (beware! bugs galore!)
- Does it generalize? What to sync, when to sync?
- What is the data model on the client (and server)?
- What other patterns can we discover/explore?

# Postgres everywhere

Your data, in sync, wherever you need it.

[Quickstart](#)[About](#)

- [electric-sql.com](#)
- [Phoenix + PostgreSQL](#)
- [More mature](#)



## Sync Engine

Sync partial replicas of your data into



## Data Delivery Network

Load data faster than you can query



## PGlite

Embed a lightweight client Postgres with

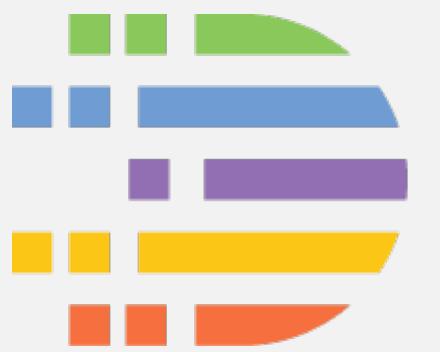
"What if we use this to synchronise data  
across Elixir nodes? Durable PubSub?"

- Chris McCord

# Thank yous

- Anthony Accomazzo ([sequin.io](http://sequin.io))  
[blog.sequinstream.com/build-your-own-sqs-or-kafka-with-postgres/](http://blog.sequinstream.com/build-your-own-sqs-or-kafka-with-postgres/)
- Tuomas Artman ([linear.app](http://linear.app))  
[linear.app/blog/scaling-the-linear-sync-engine](http://linear.app/blog/scaling-the-linear-sync-engine)
- James Arthur ([electric-sql.com](http://electric-sql.com))
- Chris McCord and Steffen Deusch (LiveView)

# The projects that never were?



**Dashbit**