

Exploding Chickens

A full-stack card game

Radison Akerman, April 2023

Hi there, my name is

Radison Akerman.

Incoming SWE Intern @ SpaceX
Project Manager & SWE @ UIC CoE
Computer Science + Business @ UIC





Draw Deck x52



Discard Deck x0



Exploding Chicken

Avoid this little guy at all costs. Once this card is drawn, you must use a defuse card to stop the ticking time bomb. If you don't have a defuse card, it looks like your time is up.



Exploding Chicken

if (card is drawn)
 if (player has defuse)
 plays defuse, places
 chicken back in draw deck
 else
 player explodes, removed
 from game permanently



Defuse Card

if (card is drawn)
place card in players hand

if (card is played &&
player is exploding)
discard card
&& prompt player to place
chicken back in draw deck
&& advance turn



Shuffle Card

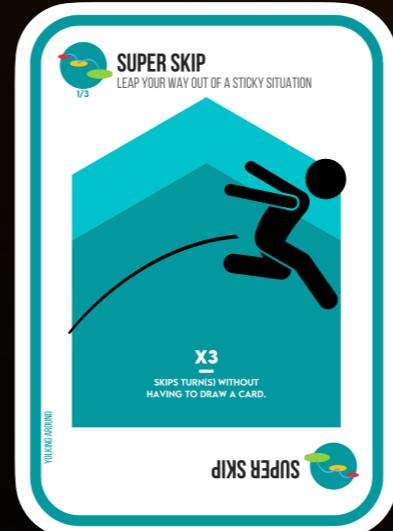
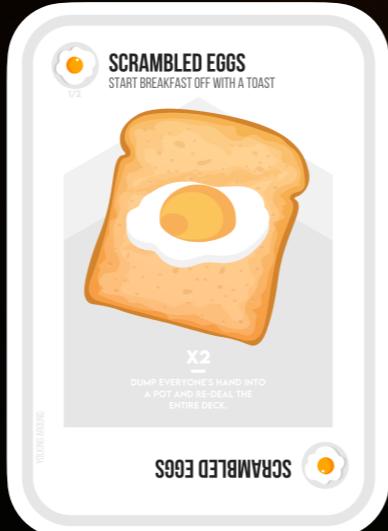
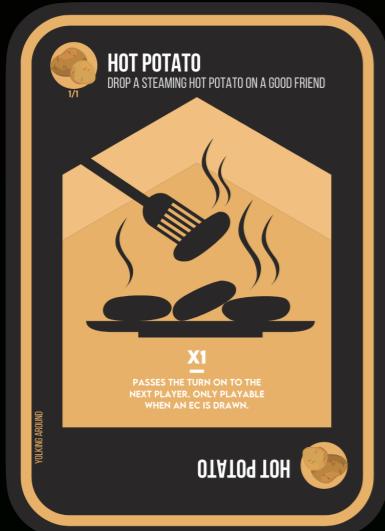
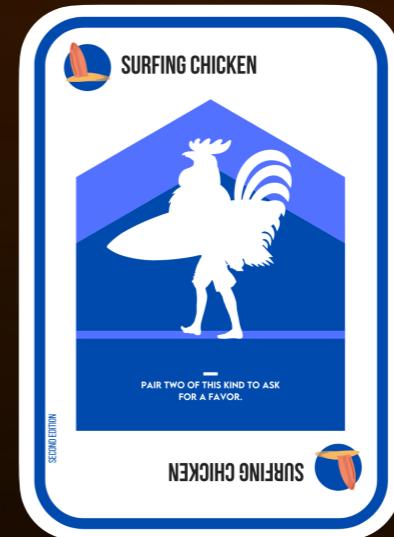
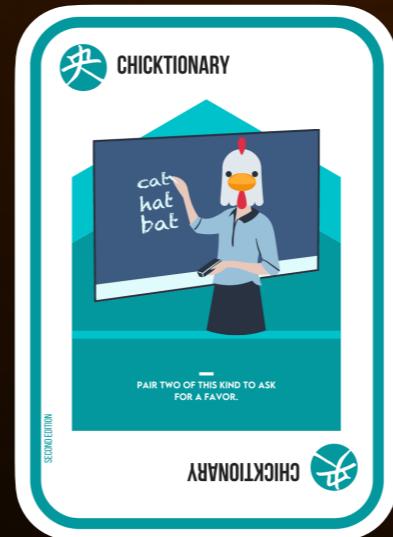
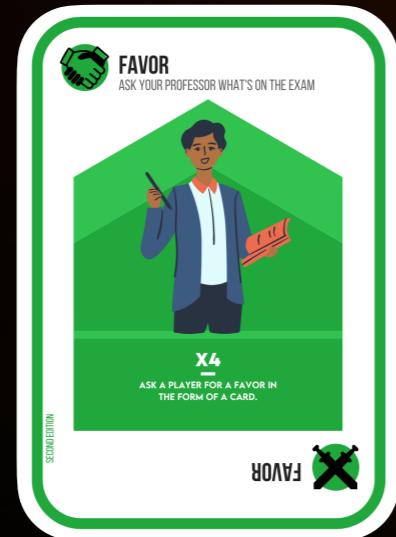
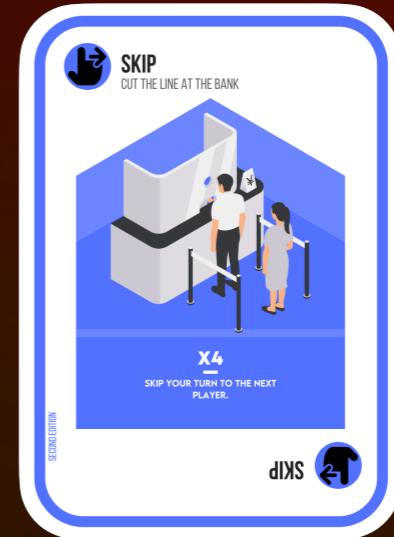
if (card is drawn)
place card in players hand

if (card is played)
shuffle draw deck
&& discard card

Shuffle Card



- Player wants to play card
- Sends “play-card” request
- Cascading validation phase
 - Is the player valid?
 - Is it their turn?
 - Do they have this card?
 - Can they play it now?
- Completes card action (randomize draw deck)
- Discard card
- Tell everyone what happened



1,050 Games
8,500 Minutes
40,000 Cards Played

Since April 2021

Demo

Type in your web browser...
rakerman.com/link/demo

Scalable and fast

Room for expansion

Easy to understand

Where do you start?

Maximize compatibility

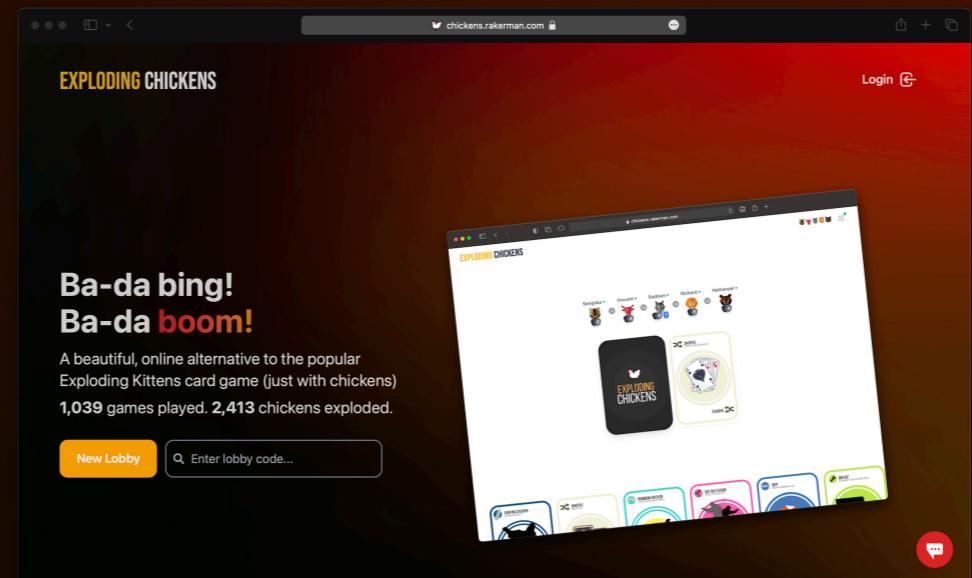
Quick to develop

Real-time communication

> **Web Browsers**

Functional Programming v1 & v2

Object Oriented Programming v3



User Interface

Browser Engine

Rendering Engine

Networking

Javascript
Interpreter

UI Backend

Data Persistence

User Interface

Browser Engine

Rendering Engine

Networking

Javascript
Interpreter

UI Backend

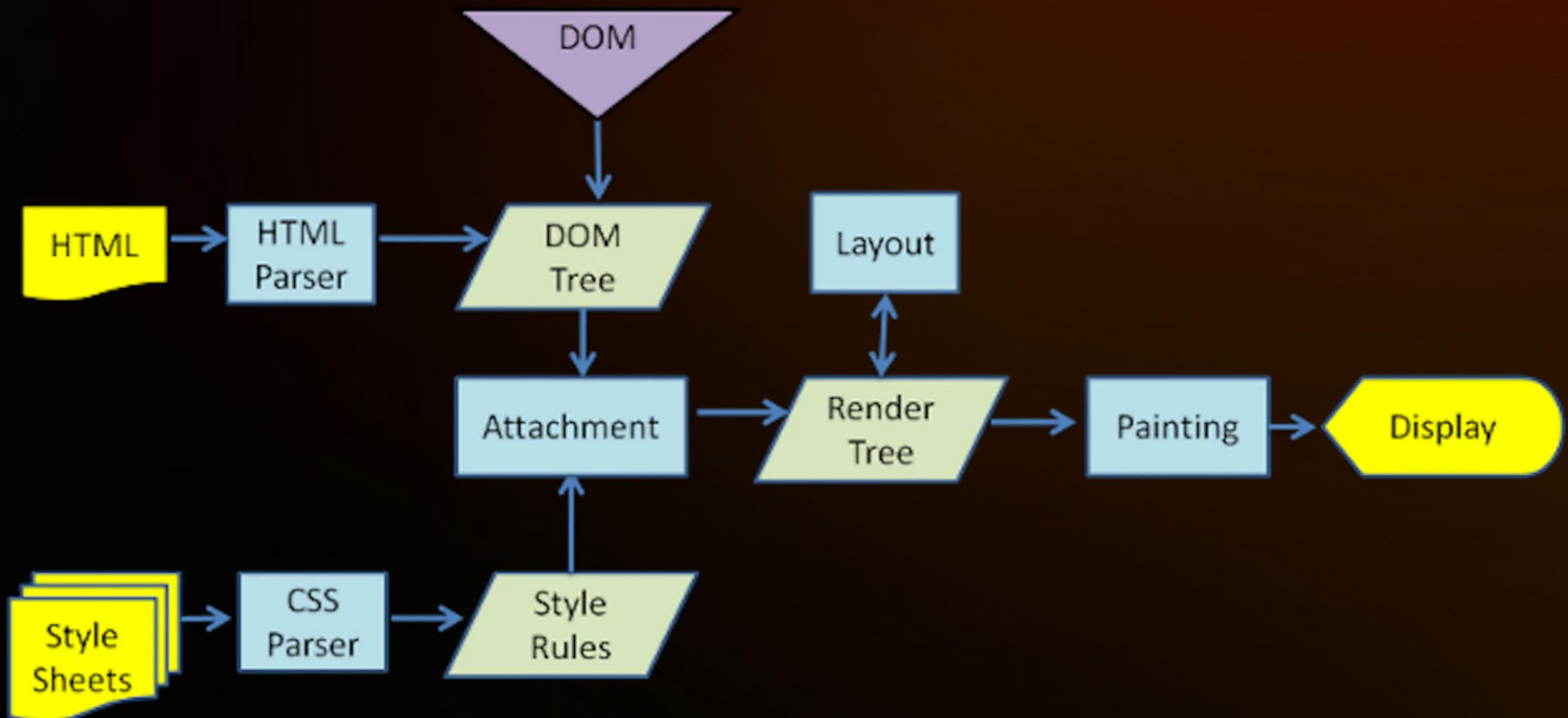
Data Persistence

Parsing HTML to Construct DOM Tree

Render Tree Construction

Layout of the Render Tree

Painting the Render Tree



```
<html>
  <body>
    <p> Hello World </p>
    <div>  </div>
  </body>
</html>
```

HTMLHtmlElement

HTMLBodyElement

HTMLParagraphElement

HTMLDivElement

Text

HTMLImageElement

User Interface

Browser Engine

Rendering Engine

Networking

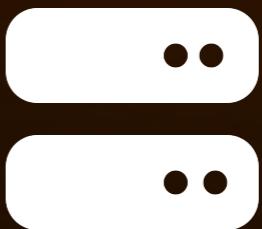
Javascript
Interpreter

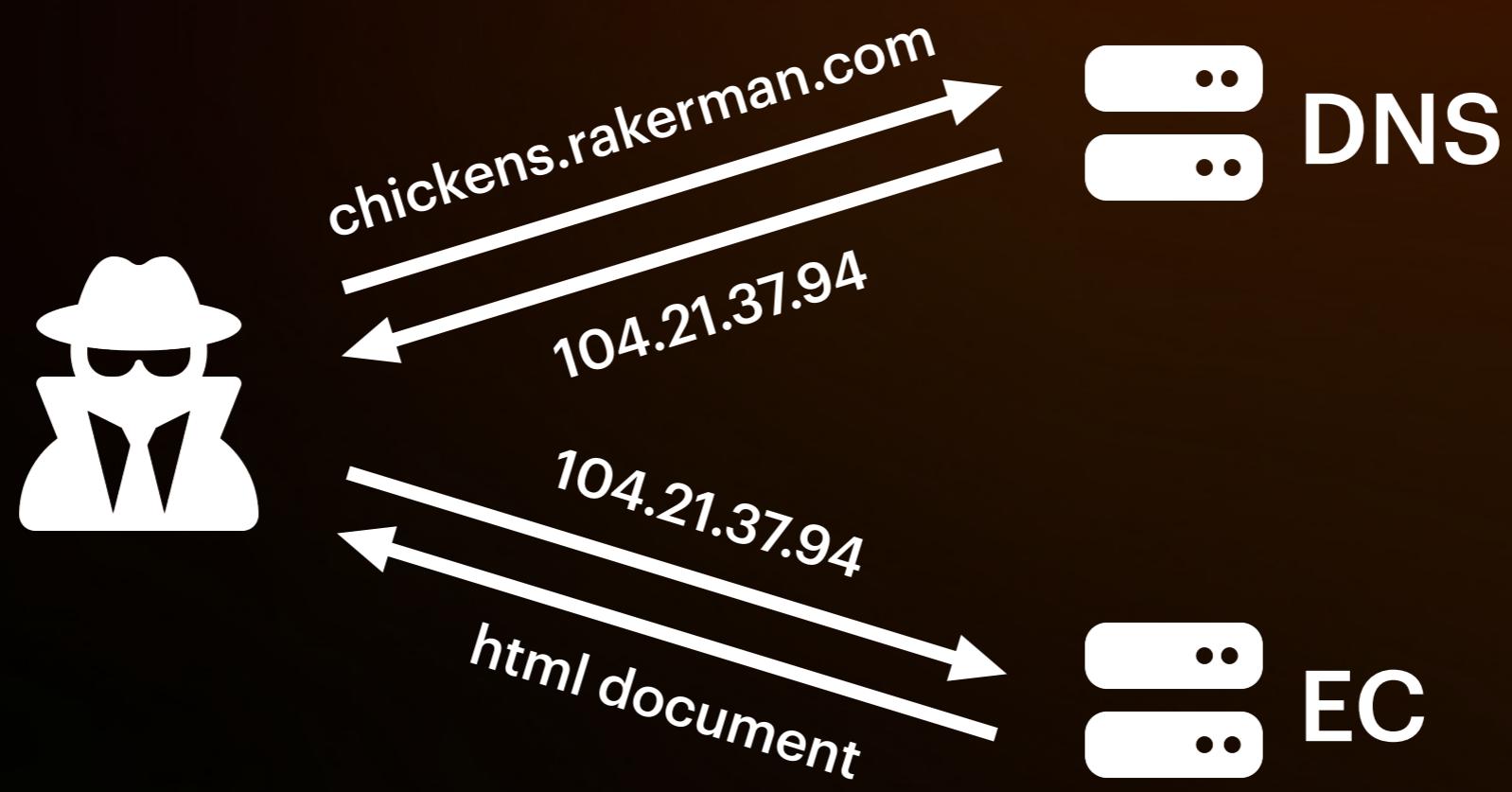
UI Backend

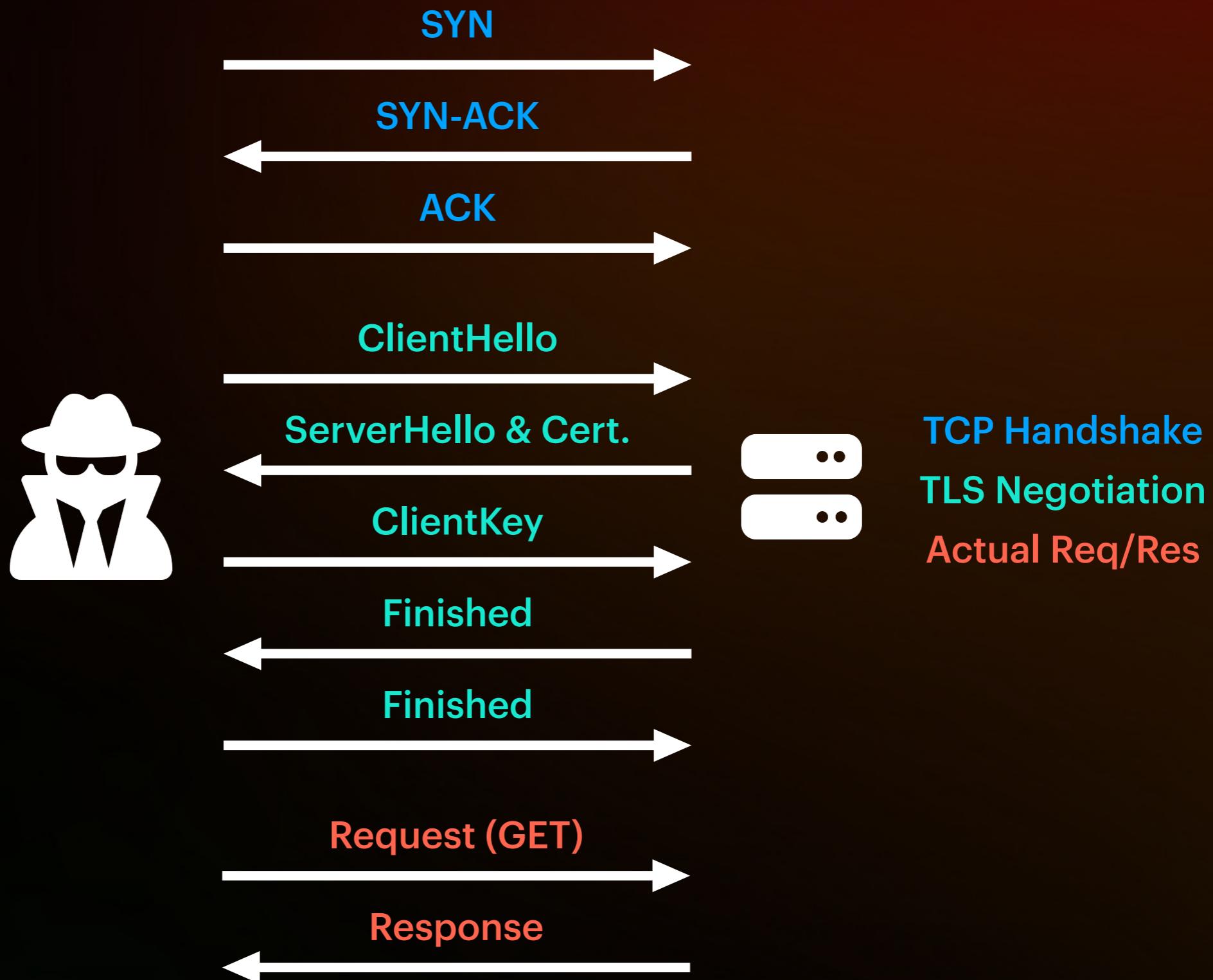
Data Persistence



chickens.rakerman.com









```
<html>
  <body>
    <p> Hello World </p>
    <div>
      
    </div>
  </body>
</html>
<script src="ex.js"/>
```

User Interface

Browser Engine

Rendering Engine

Networking

Javascript
Interpreter

UI Backend

Data Persistence

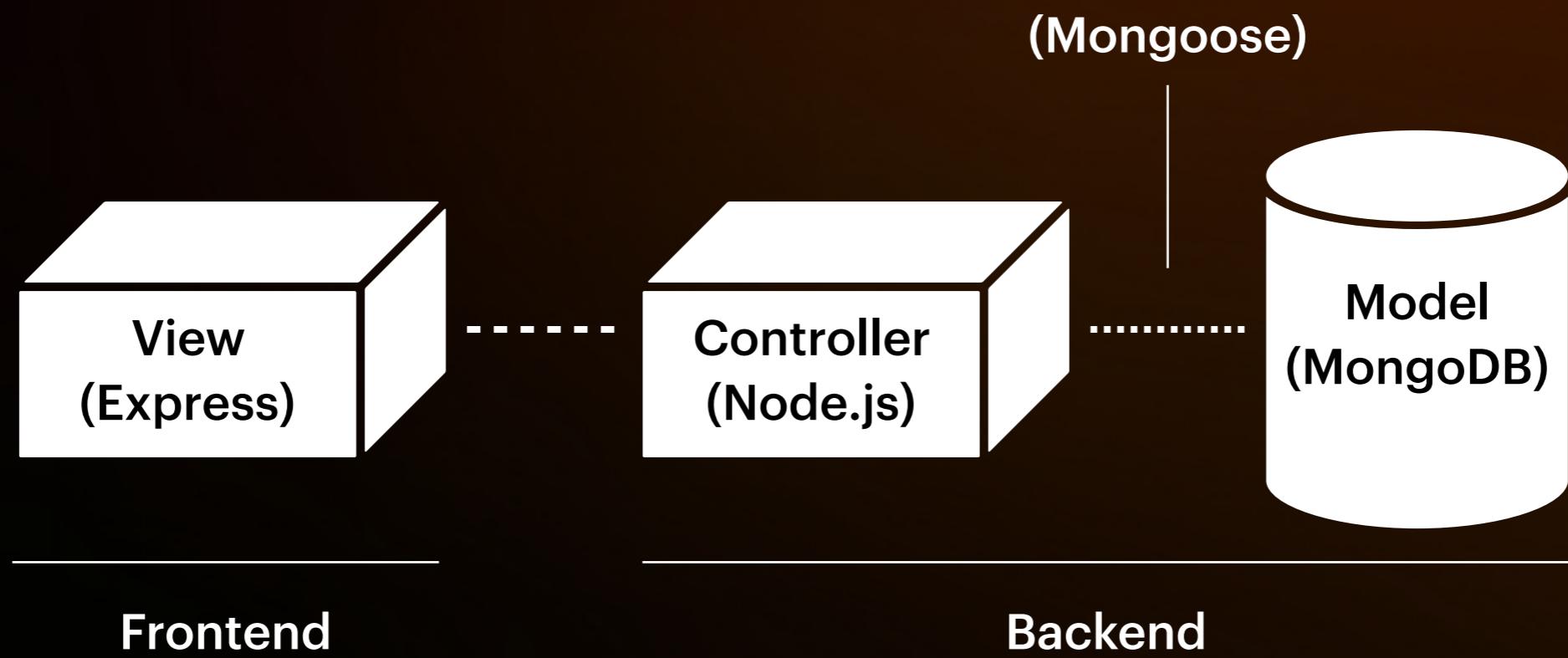
Web Browsers

- > Functional Programming v1 & v2
- Object Oriented Programming v3

v1.0.0

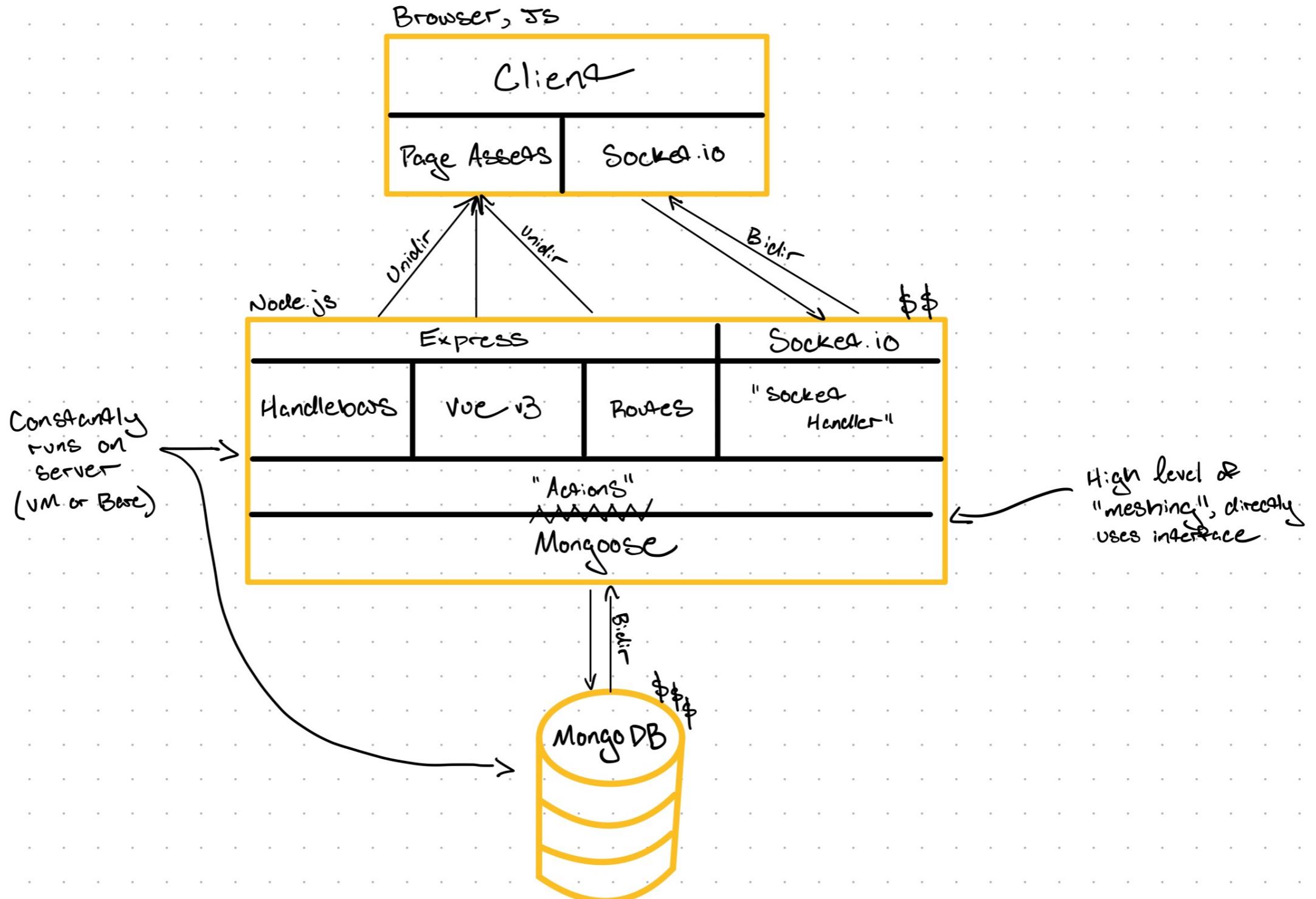
An unorganized first attempt (that works)

- Full-stack web application
 - Node.js — an asynchronous event-driven JavaScript runtime geared towards scalable network apps
 - MongoDB — a NoSQL document-oriented database
 - Mongoose as an ODM (Object Data Modeling)
 - Handlebars — HTML templating language
 - Socket.io — real-time, bidirectional communication



Exploding Chickens

Stack v2.0



Game

Cards

- _id
- action
- assignment
- position
- pack

Players

- _id
- nickname
- avatar
- seat_position
- wins
- sockets_open
- is_host
- is_dead

Events

- _id
- tag
- req_player
- target_plyr
- related_key
- related_value

Misc data...

Socket.io Client



socket-handler.js

game-actions.js

card-actions.js

Shuffle Card

- Player wants to play card
- Sends “play-card” request
- Cascading validation phase
 - Is the player valid?
 - Is it their turn?
 - Do they have this card?
 - Can they play it now?
- Completes card action (randomize draw deck)
- Discard card
- Tell everyone what happened

v1.1.7 Code Review

v2.0.0

A segmented second attempt (much, much cleaner)

- Same web stack (Node.js, MongoDB, Handlebars, Socket.io) + Vue.js + Auth0
- Mocha + Istanbul — Backend tests, code coverage
- Shared repository model
- Issues and features request tracking in Github
- New Lobby abstraction

Lobby

Games

Cards

Events

Misc data...

Players

- _id
- game_assign
- nickname
- avatar
- seat_position
- wins
- sockets_open
- is_host
- is_dead

Events

- _id
- tag
- req_player
- target_plyr
- related_key
- related_value

v2.0.6 Code Review

Web Browsers

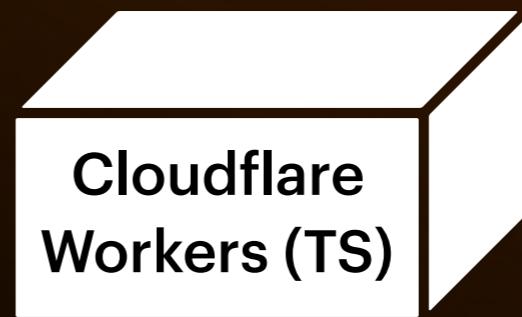
Functional Programming v1 & v2

> **Object Oriented Programming v3**

v3.0.0

A globally-distributed 3rd rework (here we go again)

- Completely new web stack: Typescript (OOP) + Cloudflare Workers + Websockets + Nuxt + Auth0
 - State-of-the-art, lightweight & serverless = cheap
 - Ability to run 'on the edge', decentralized (worldwide)
 - Same test coverage, lobby abstraction, and user interface

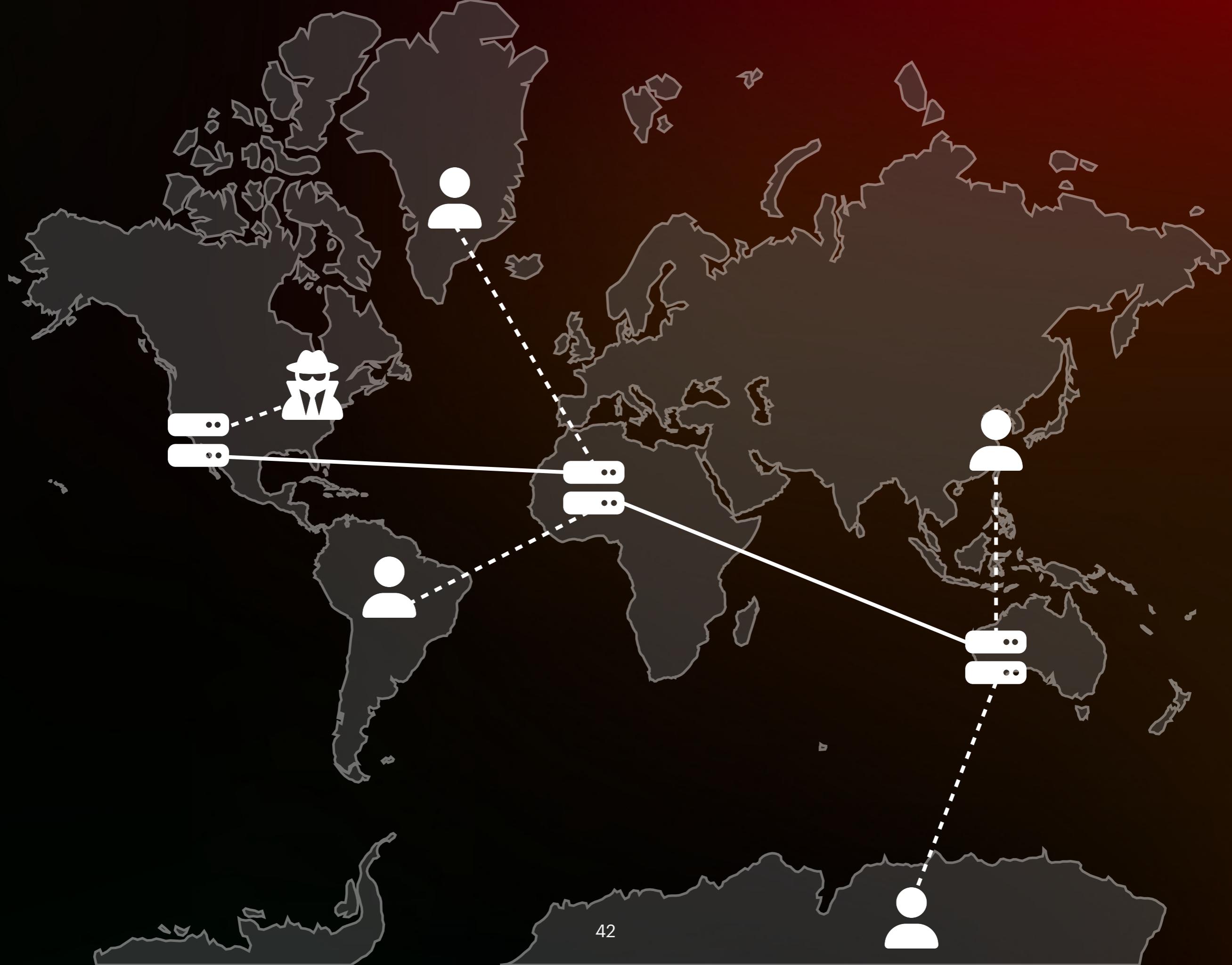


Frontend

Backend

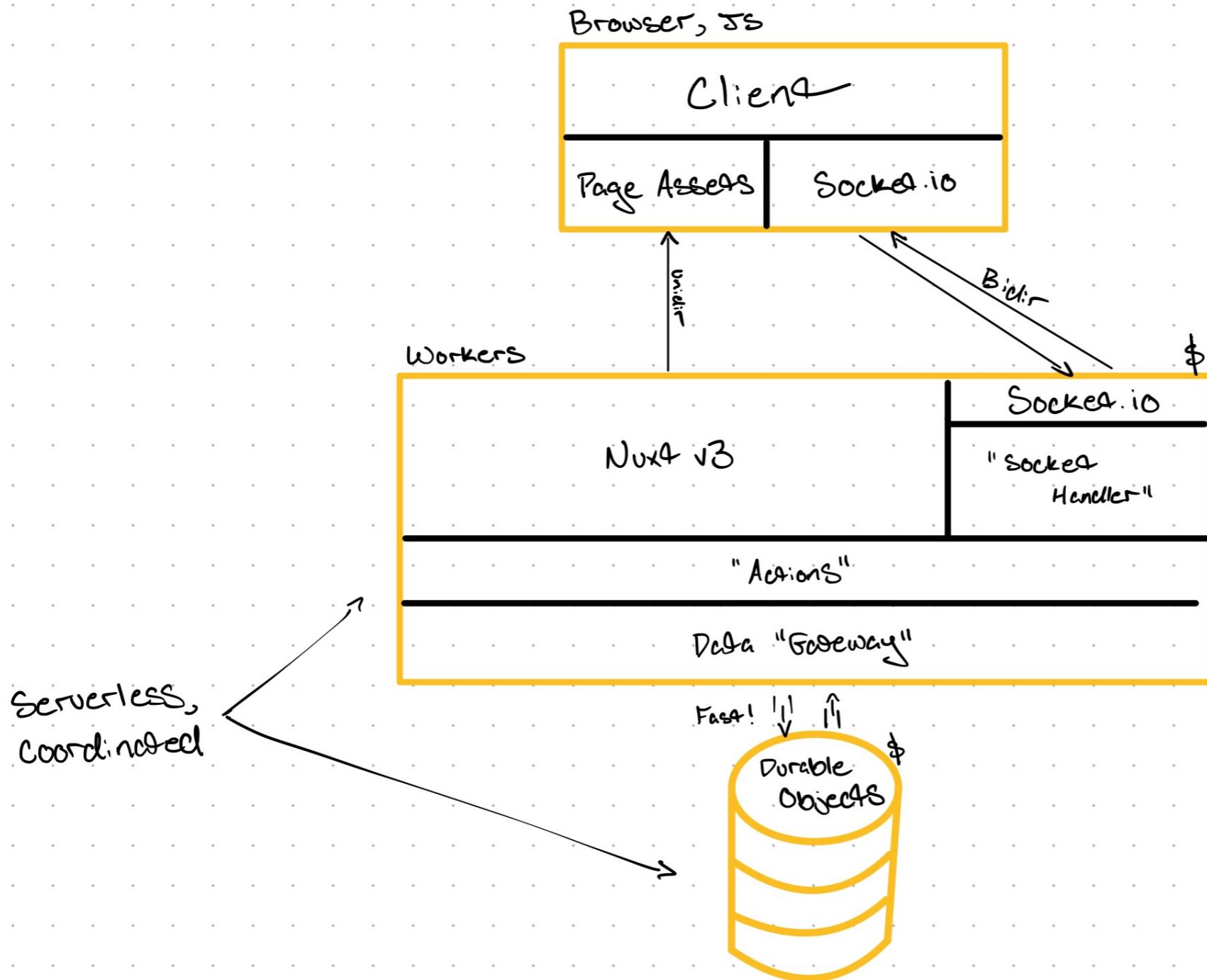






Exploding Chickens

Stack v3.0



v3.0.0-beta Code Review

Key Takeaways

- Do your research when picking a language
- Build a solid data structure
- Clearly define boundaries between parts of your project
- Give yourself room to build something better
- Visualize everything upfront

Questions

chickens.rakerman.com