



# ARE YOU BEING SERVERED?

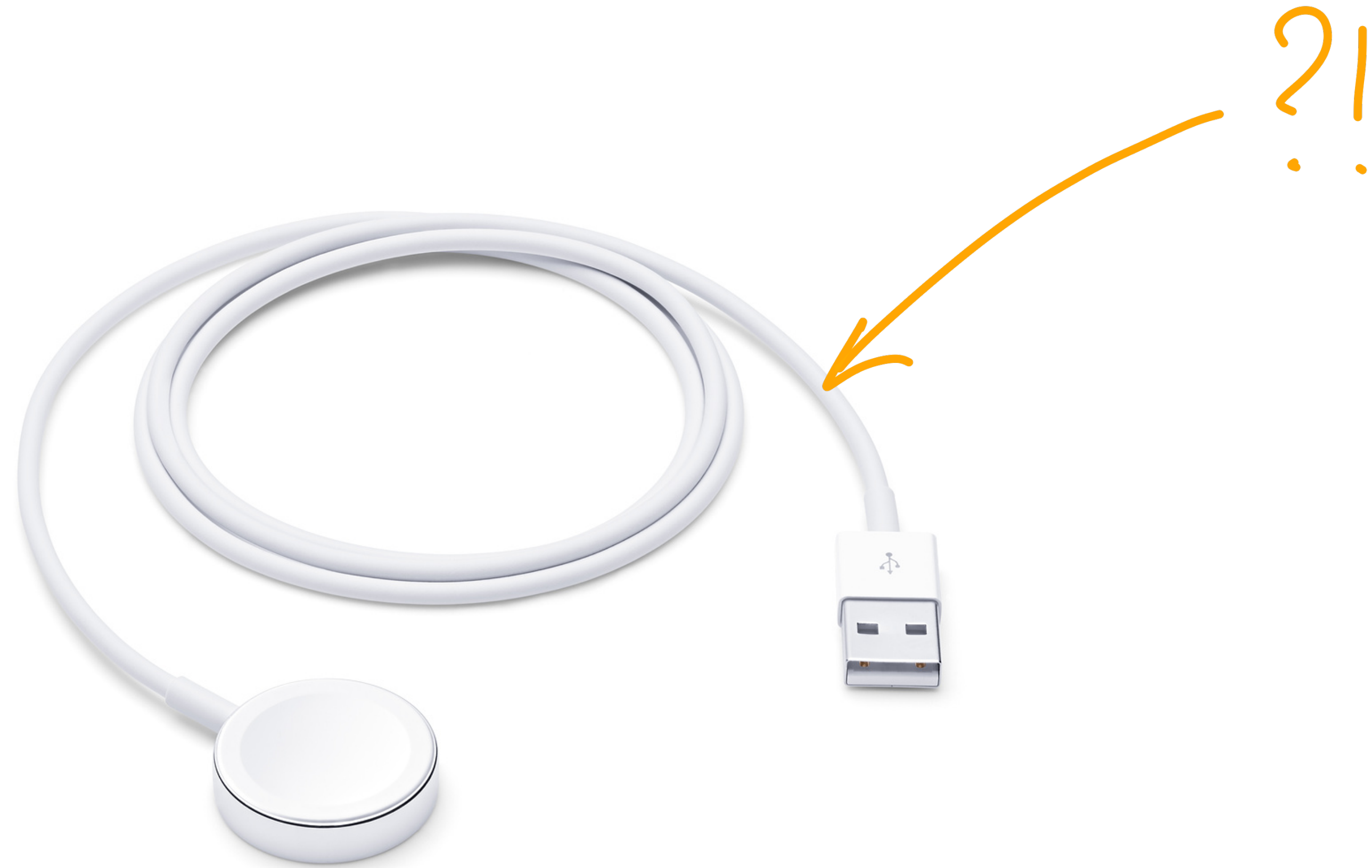
## EXPLORING A SERVERLESS WEB

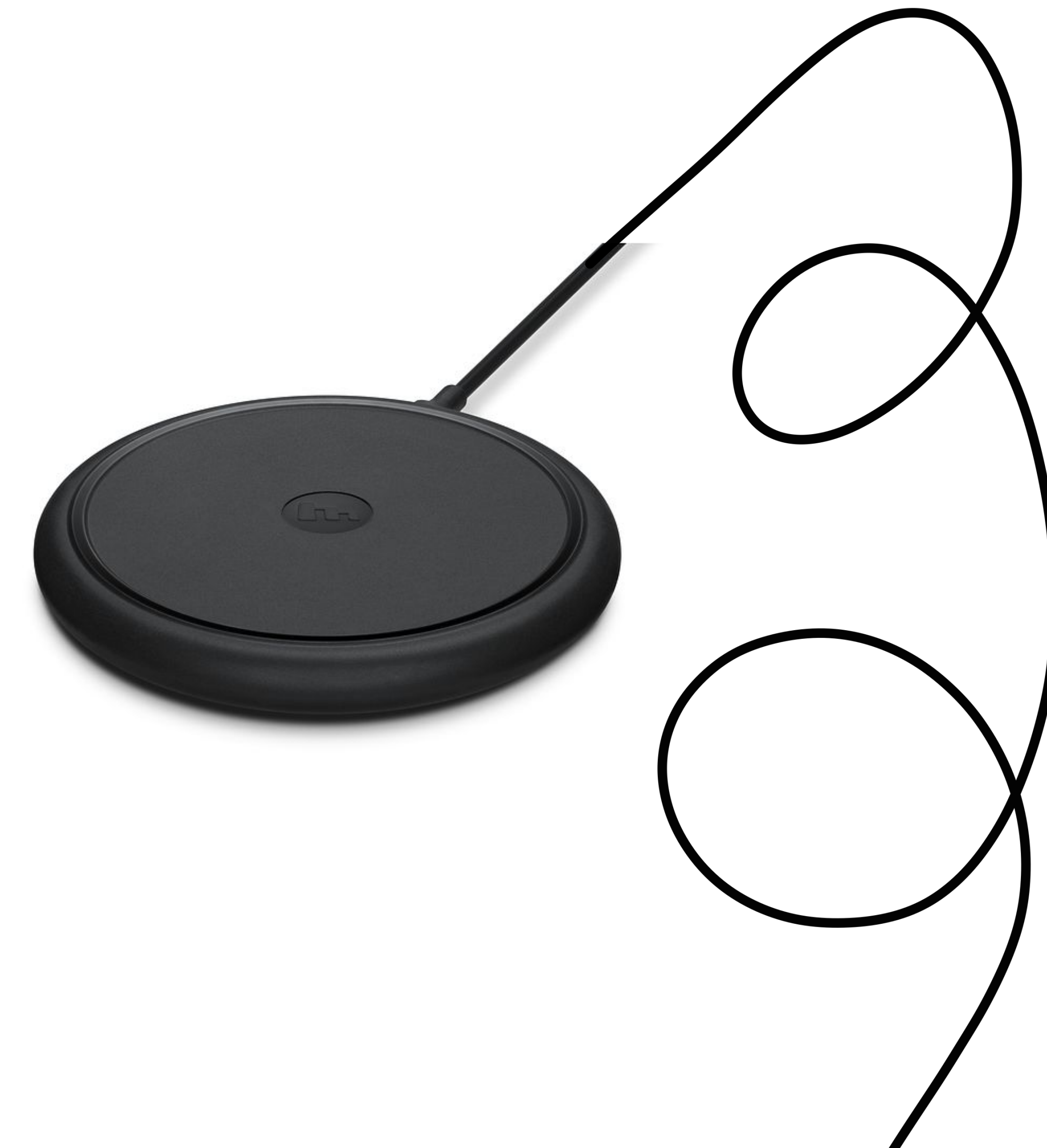
PHIL HAWKSWORTH













Are you being served?

@philhawksworth



# WIRELESS



# SERVERLESS

**FOCUSSING ON  
THE THINGS YOU  
CARE ABOUT**

**FOCUSSING ON  
THE THINGS YOU  
*have to*  
CARE ABOUT**

*liberated*  
**TO FOCUS ON  
WHAT MATTERS**

*the problem with*

# DOING BIG PROJECTS

# CONSTRAINTS AND COMPLEXITY AND INFRASTRUCTURE

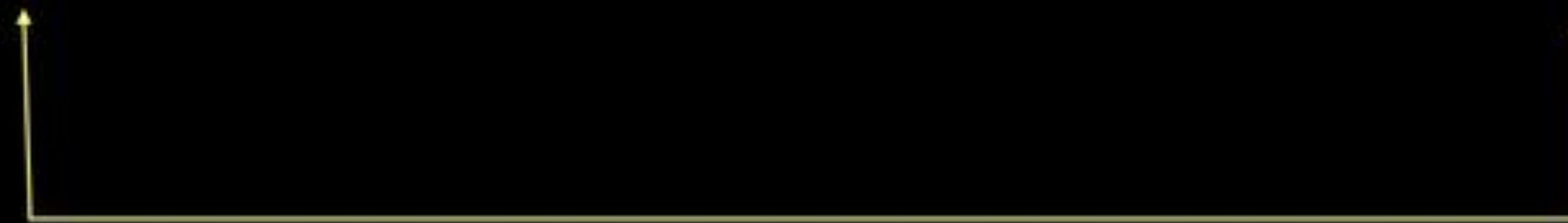
*oh my!*

# WHERE TO FOCUS OUR EFFORTS?

Back End

Front End

OS Database Servers Server Side Languages APIs Frameworks JavaScript CSS HTML



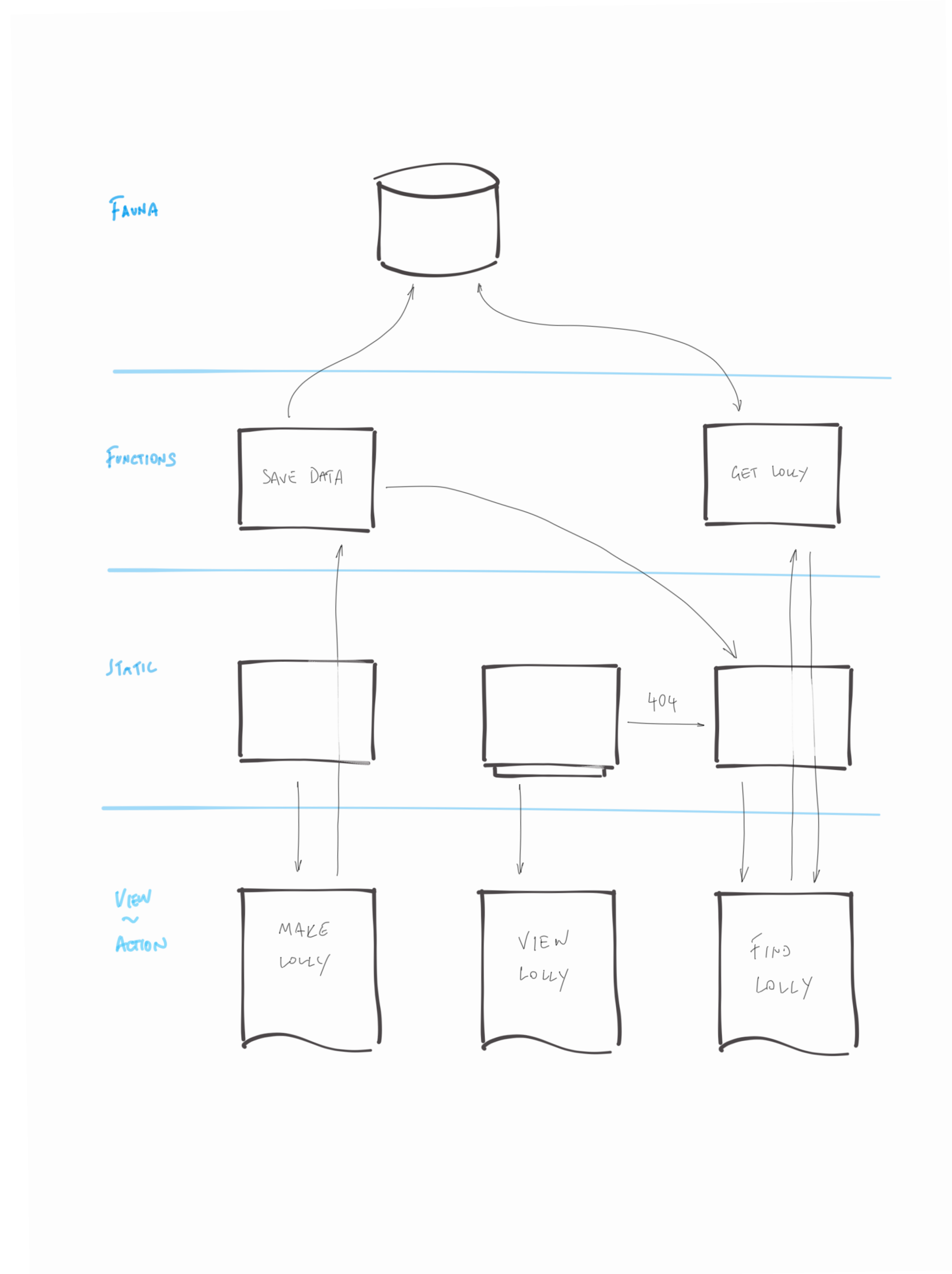
# LAMP

Linux Apache MySQL PHP





# LINES AND BOXES



**@PHILHAWKSWORTH**

HE / HIM

DEVELOPER EXPERIENCE,  
NETLIFY



**FINDTHAT.AT/JAMSTACK-BOOK**

FINDTHAT.AT/SOMETHINGSOMETHING

FINDTHAT.AT/SERVERED

FINDTHAT.AT



# ARE YOU BEING SERVERED?

## EXPLORING A SERVERLESS WEB

PHIL HAWKSWORTH



*Are You Being Served?*

BBC

# Are You Being Served?



The Complete First Series  
Including pilot episode

DVD  
VIDEO

PG

EMI

NAT COHEN presents

# "Are You Being Served?"

By JEREMY LLOYD  
and DAVID CROFT



Starring

MOLLIE SUGDEN | JOHN INMAN

FRANK THORNTON  
TREVOR BANNISTER

Also Starring

WENDY RICHARD · ARTHUR BROUGH  
NICHOLAS SMITH · HAROLD BENNETT  
and ARTHUR ENGLISH

Produced by ANDREW MITCHELL

Directed by BOB KELLETT

Technicolor® · Distributed by EMI Films Limited





---

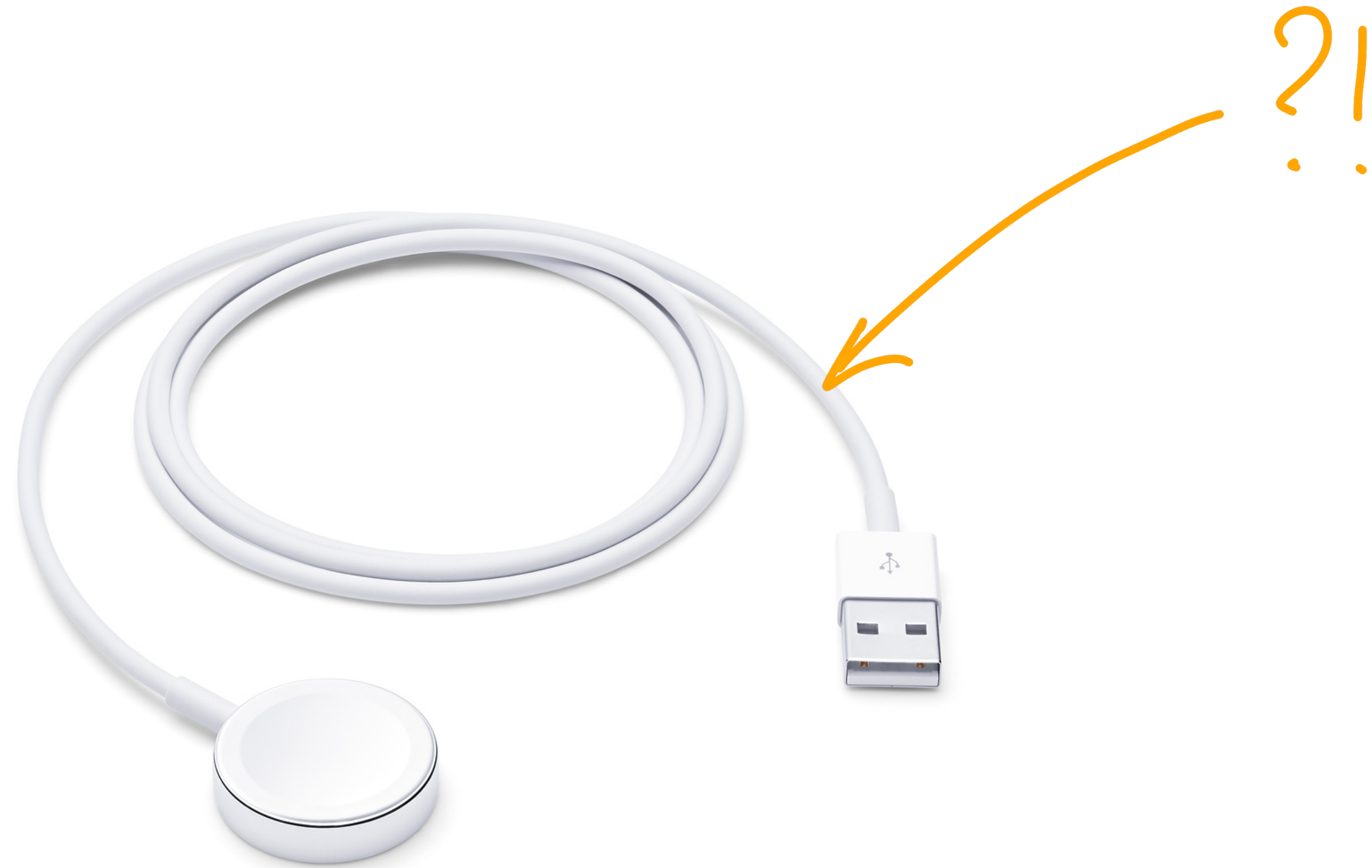
# ARE YOU BEING SERVERED?

## EXPLORING A SERVERLESS WEB

PHIL HAWKSWORTH

# FUNCTIONS AS A SERVICE

# SERVING WEBSITES WITHOUT WEBSERVERS



# *jamstack*

**JAVASCRIPT / API / MARKUP**

# STACKS FOR DAYZ

AN / LAMP / MAMP / XAMP / ROR / WISA / MEAN

WISA / MEAN / LAMP / MAMP / XAMP / ROR /

# jamstack

*A modern architecture —*

*Create fast and secure sites and dynamic apps  
with **J**avaScript, **A**PIs, and pre-rendered  
**M**arkup, served without web servers*

---

A black and white close-up portrait of Aaron Swartz, smiling and looking slightly to the left. He has dark, wavy hair and a light beard. The background is plain white.

# **BAKE, DON'T FRY**

*– Aaron Swartz*



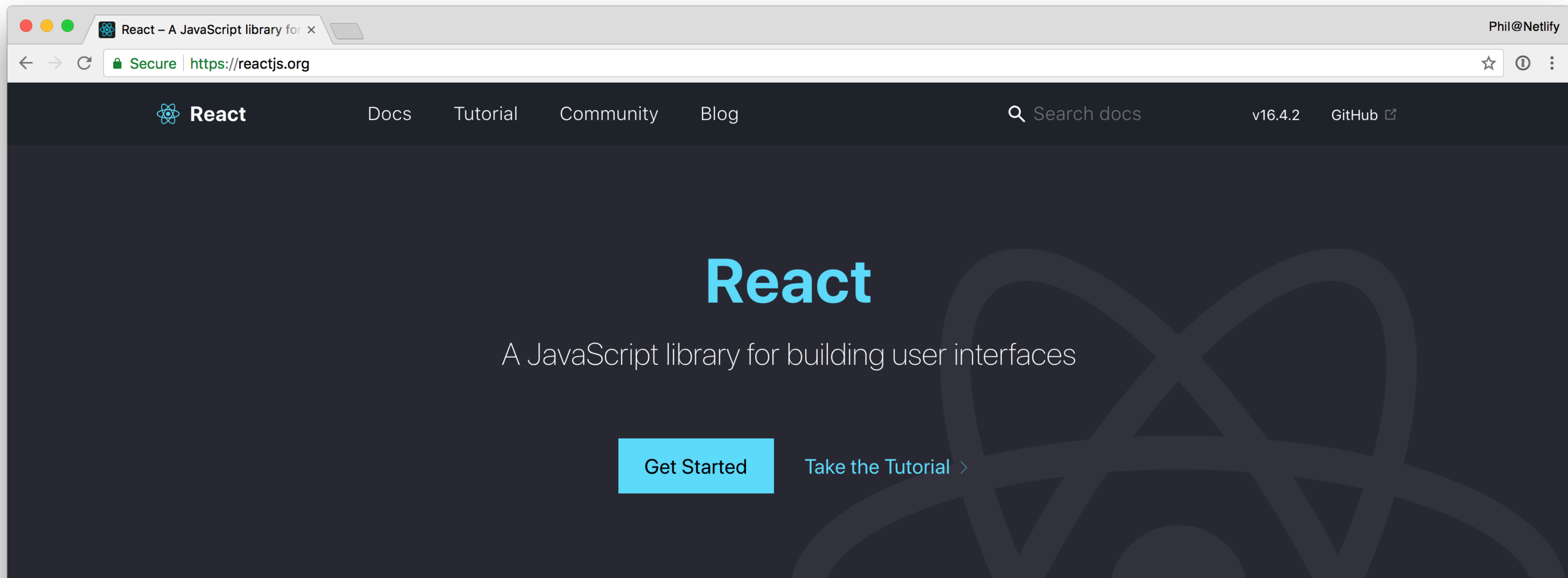
**I CARE ABOUT NOT HAVING TO  
MAINTAIN CRANKY AOLSERVER,  
POSTGRES AND ORACLE INSTALLS**

**AARON SWARTZ, 2002**

# TOOLS AND ECOSYSTEM

**SO... WHAT CAN  
A STATIC SITE  
EVEN BE?**

# REACTJS.ORG



## Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

## Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

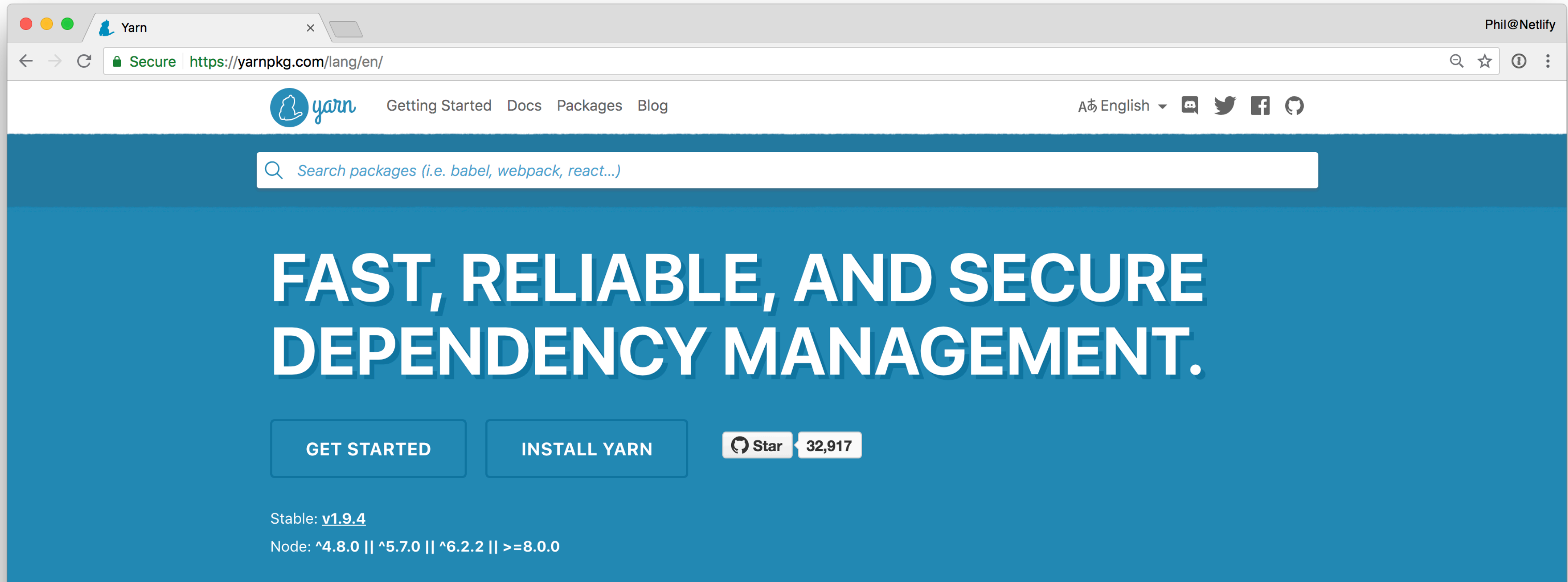
Since component logic is written in JavaScript

## Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node

# YARNPKG.COM



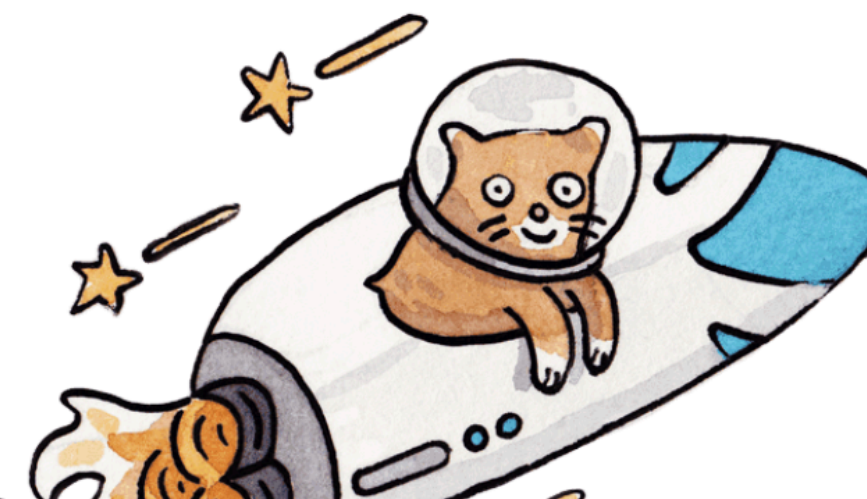
The screenshot shows a web browser window with the URL `https://yarnpkg.com/lang/en/`. The page features a blue header with the Yarn logo, navigation links for 'Getting Started', 'Docs', 'Packages', and 'Blog', and social media icons for GitHub, Twitter, Facebook, and YouTube. Below the header is a search bar with the placeholder text 'Search packages (i.e. babel, webpack, react...)'. The main content area has a large blue background with the text 'FAST, RELIABLE, AND SECURE DEPENDENCY MANAGEMENT.' in white. Below this text are three buttons: 'GET STARTED', 'INSTALL YARN', and a GitHub Star button showing '32,917' stars. At the bottom of the blue section, the text 'Stable: v1.9.4' and 'Node: ^4.8.0 || ^5.7.0 || ^6.2.2 || >=8.0.0' is displayed.

## FAST, RELIABLE, AND SECURE DEPENDENCY MANAGEMENT.

[GET STARTED](#)[INSTALL YARN](#)[Star](#) 32,917Stable: [v1.9.4](#)Node: [^4.8.0](#) || [^5.7.0](#) || [^6.2.2](#) || [>=8.0.0](#)

### Ultra Fast.

Yarn caches every package it downloads so it never needs to download it again. It also parallelizes operations to maximize resource utilization so install times are faster than ever.



# VUEJS.ORG

Vue.js

Secure | <https://vuejs.org>

Learn ▾ Ecosystem ▾ Team Support Vue ▾ Translations ▾

## The Progressive JavaScript Framework

[WHY VUE.JS?](#) [GET STARTED](#) [GITHUB](#)

Special Sponsor

Function as a Service Platform and Library

### Approachable

Already know HTML, CSS and JavaScript? Read the guide and start building things in no time!

### Versatile

An incrementally adoptable ecosystem that scales between a library and a full-featured framework.

### Performant

20KB min+gzip Runtime  
Blazing Fast Virtual DOM  
Minimal Optimization Efforts

Patreon Sponsors

# A LITTLE MORE "DYNAMIC"

# SMASHINGMAGAZINE.COM

Smashing Magazine membership page. The header includes navigation links for Articles, Books, Events, Jobs, and Membership, along with a search bar. The main heading is "Smashing Membership" with a sub-headline: "The best thing invented since yummy cappuccino. Smashing books, webinars and conferences – for just 1 coffee a month!". Below this is a "Choose your option." section with three membership tiers: "Supporter" (\$3), "Member" (\$5), and "I Feel Smashing" (\$9). Each tier includes a list of benefits and a "Become a Smashing" button.

Smashing Magazine articles page. The header features navigation links and a search bar. A "Don't Miss These Latest Articles" section highlights four featured articles, each with a "Leave a comment" button and a "Seriously, red?" badge. The articles are: "How To Start Using Sketch And Framer X" by Martina Pérez, "Increase Customer Centricity With Workshops" by Claire Mason, "When Do We Need A Design System? An Interview With Brad Frost" by Vitaly Friedman, and "Inspiring Wallpapers For A Colorful June (2019 Edition)" by Cosima Mielke. Each article includes a "Leave a comment" button and a "Seriously, red?" badge.

Smashing Magazine events page. The header includes navigation links and a search bar. The main heading is "SmashingConf 2019-2020" with a sub-headline: "Smashing Conferences are friendly, inclusive events for people who care about their work. No fluff, no fillers, no multi-track experience – just actionable insights applicable to your work right away. With live sessions on accessibility, performance, front-end, UX, CSS, JavaScript and interface design. Please respect our code of conduct." Below this is a "What's SmashingConf Like? ↓" button and a "Get 10% off all tickets with your Smashing Membership. Just sign in." button. The page features two event cards: "Toronto, Canada Jun 25-26, 2019" and "London, UK Jul 9-10, 2019". Each card includes a description of the event and an "Explore all speakers and topics" button.





blog speaking about search

# HAWKSWORX

/hɔːkswɔːks/

The blog, playground, and generally a home on the web for Phil Hawksworth, a web nerd, amateur comedian, and cat botherer.

The source code of this site is available on [GitHub](#) and is hosted and updated by [Netlify](#) automatically after each code commit

Other than where specified, the content on this site is published under a [Creative Commons Attribution 3.0 licence](#).

Subscribe to a feed of blog posts on this site

You can usually find me on [Twitter](#). And you can sometimes find me at web development [conferences](#), where I might be talking about development techniques. I blog here less frequently than I'd like, but manage it a little more often on the [Netlify blog](#).

Wherever you find me, online or in the real world, please do come and say a friendly hello.

## Recently on the blog

August 19th 2019

### [Power up Gatsby sites with A/B testing on Netlify](#)

[#jamstack](#) , [#netlify](#) , [#gatsby](#)

If you host your Gatsby site with [Netlify](#), you can create A/B or multi-variant tests without the need for client-side UI manipulation or other JavaScript intervention. This post and demo video explain how.

Edit settings

## Build hooks

---

Build hooks give you a unique URL you can use to trigger a build.

Twitter activity trigger:

[https://api.netlify.com/build\\_hooks/58d50555d6865d3a3d5a1f19](https://api.netlify.com/build_hooks/58d50555d6865d3a3d5a1f19)



Comment approved:

[https://api.netlify.com/build\\_hooks/5a9178c6df99537d12e5d057](https://api.netlify.com/build_hooks/5a9178c6df99537d12e5d057)



Alexa voice command:

[https://api.netlify.com/build\\_hooks/5b75f84e82d3f10467d27f41](https://api.netlify.com/build_hooks/5b75f84e82d3f10467d27f41)



[Learn more about build hooks in the docs](#) ↗

Add build hook

## Build image selection

---

BETA

**PHILIP GOT COCKY**

**SOME PEOPLE ON  
THE INTERNET NOTICED**



Are you being servered?

@philhawksworth





**Zach Leatherman**

@zachleat

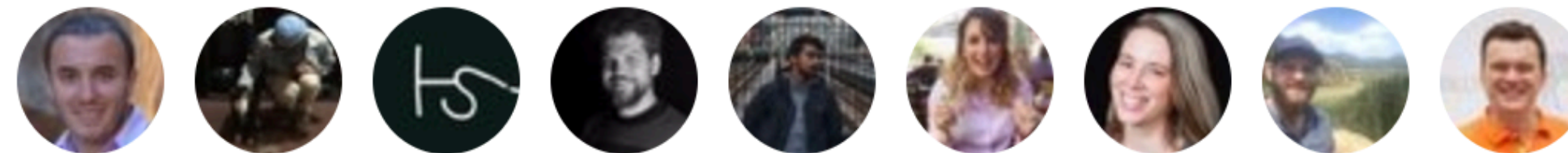
Following



free side project idea: HTML-only static site generated clock that deploys a new version to [@netlify](#) every minute

8:54 PM - 19 Jul 2018

30 Likes





**Phil Hawksworth**

@philhawksworth



Replying to [@zachleat](#) [@chriscoyier](#) and 2 others

Hold my beer.

10:25 PM - 19 Jul 2018

12 Likes





**Phil Hawksworth**

@philhawksworth



Replying to @philhawksworth @zachleat and 3 others

Only an idiot would make what you are suggesting.

[setyourwatchby.netlify.com](https://setyourwatchby.netlify.com)

2:02 PM - 20 Jul 2018

12 Retweets 41 Likes





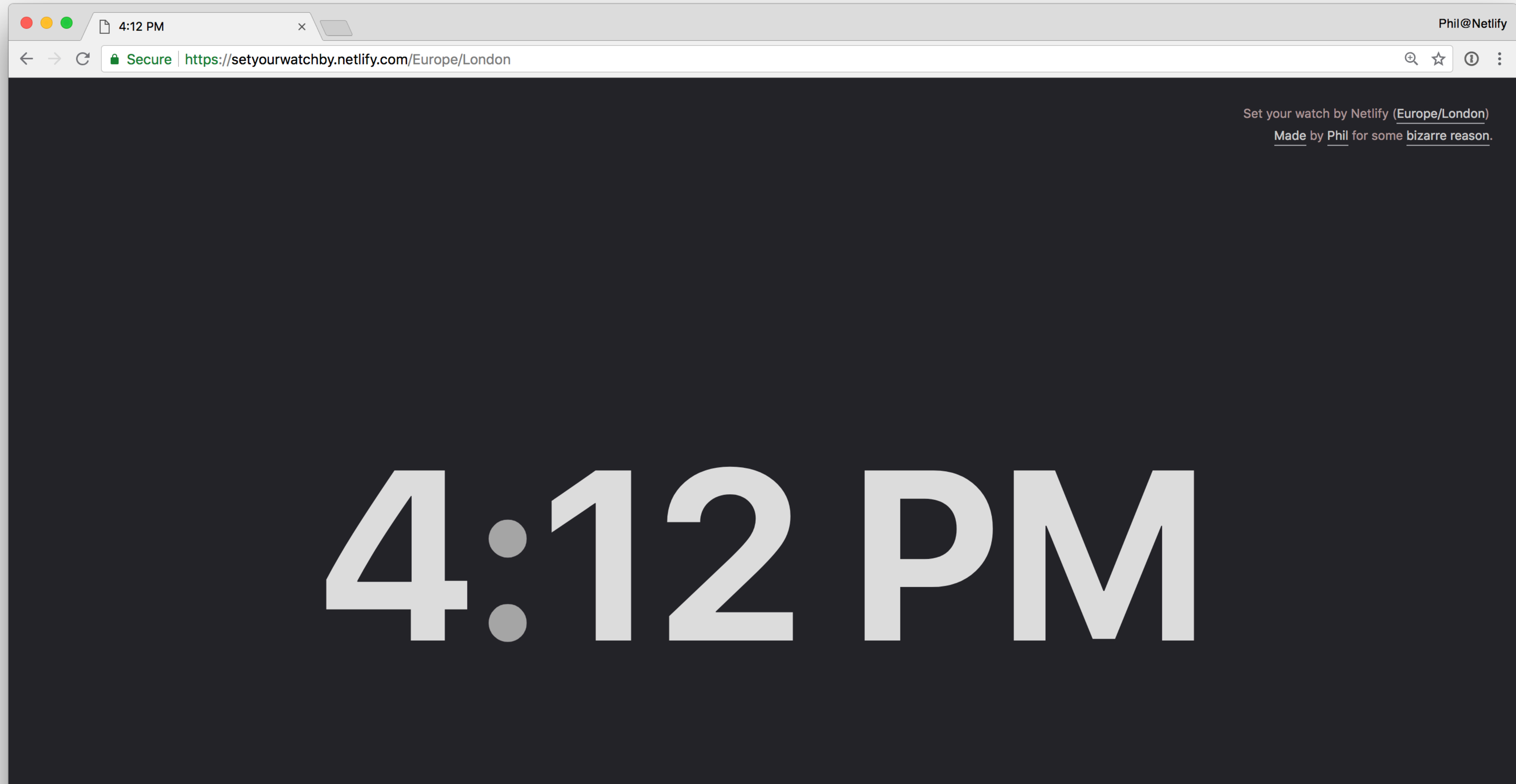


Are you being servered?

@philhawksworth

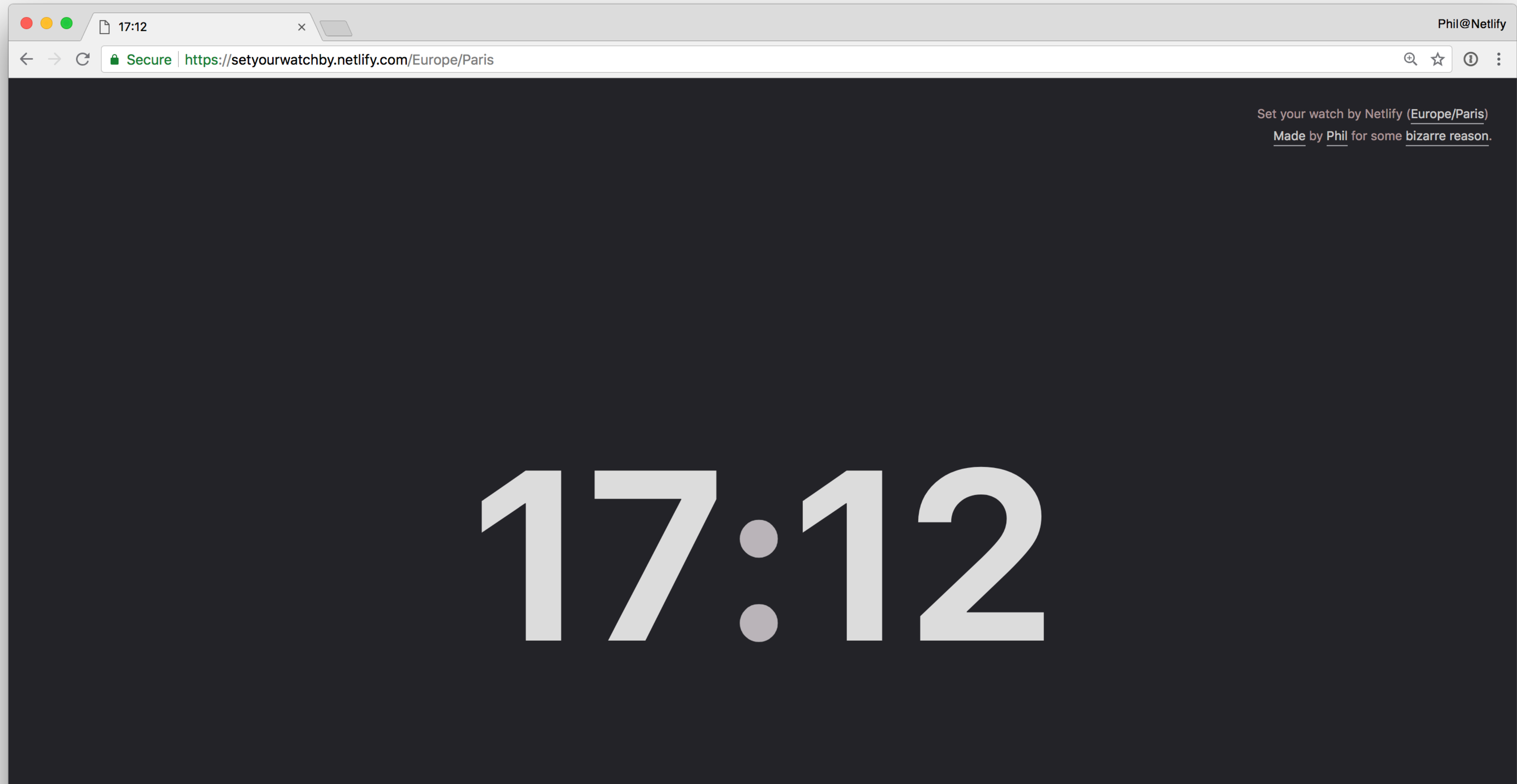
# FINDTHAT.AT/TIME

<https://setyourwatchby.netlify.com>



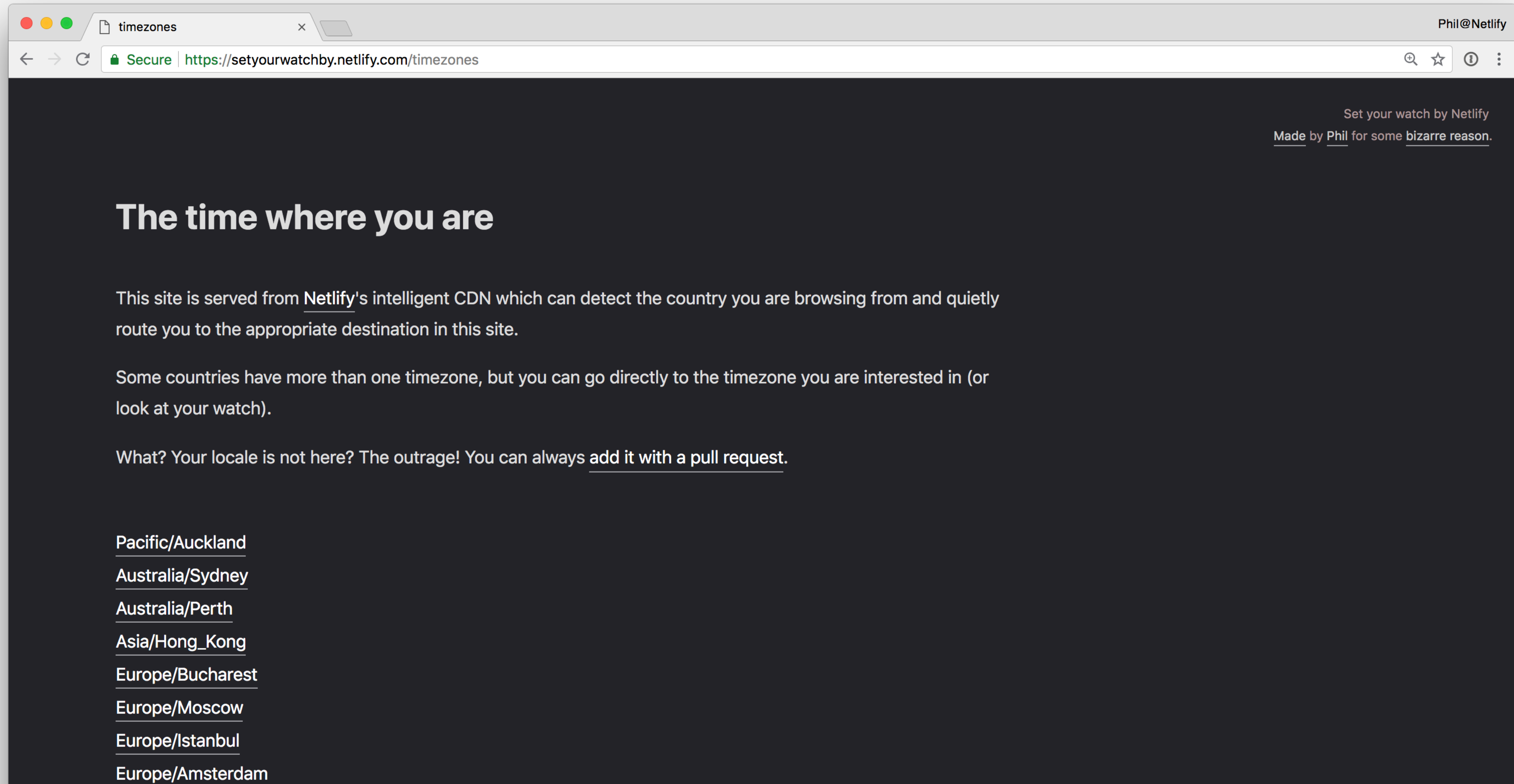
# FINDTHAT.AT/TIME

<https://setyourwatchby.netlify.com>



# FINDTHAT.AT/TIME

<https://setyourwatchby.netlify.com>



Phil@Netlify

timezones x

Secure | <https://setyourwatchby.netlify.com/timezones>

Set your watch by Netlify  
Made by [Phil](#) for some [bizarre reason](#).

## The time where you are

This site is served from [Netlify](#)'s intelligent CDN which can detect the country you are browsing from and quietly route you to the appropriate destination in this site.

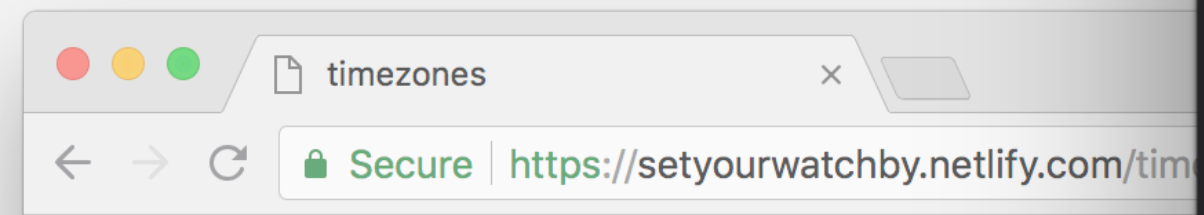
Some countries have more than one timezone, but you can go directly to the timezone you are interested in (or look at your watch).

What? Your locale is not here? The outrage! You can always [add it with a pull request](#).

- [Pacific/Auckland](#)
- [Australia/Sydney](#)
- [Australia/Perth](#)
- [Asia/Hong\\_Kong](#)
- [Europe/Bucharest](#)
- [Europe/Moscow](#)
- [Europe/Istanbul](#)
- [Europe/Amsterdam](#)

# FINDTHAT.AT/TIME

<https://setyourwatchby.netlify.com/timezones>



## The time where you

This site is served from [Netlify's](#) intelligent routing system which will route you to the appropriate destination based on your location.

Some countries have more than one time zone (e.g. Australia, look at your watch).

What? Your locale is not here? The output of the [timezones](#) endpoint will list the available time zones for your location.

[Pacific/Auckland](#)

[Australia/Sydney](#)

[Australia/Perth](#)

[Asia/Hong\\_Kong](#)

[Europe/Bucharest](#)

[Europe/Moscow](#)

[Europe/Istanbul](#)

[Europe/Amsterdam](#)

[Pacific/Auckland](#)

[Australia/Sydney](#)

[Australia/Perth](#)

[Asia/Hong\\_Kong](#)

[Europe/Bucharest](#)

[Europe/Moscow](#)

[Europe/Istanbul](#)

[Europe/Amsterdam](#)

[Europe/Paris](#)

[Europe/London](#)

[America/New\\_York](#)

[America/Chicago](#)

[America/Denver](#)

[America/Los\\_Angeles](#)

# WEBTASK.IO

The screenshot shows a web browser window with the URL `https://webtask.io/make`. The page features a dark theme and a sidebar on the left. The main content area is split into two panels: a 'Scheduler' panel on the left and a code editor on the right.

**Scheduler Panel:**

- Buttons: Refresh, Remove schedule
- Checkbox:  Write an advanced schedule
- Fields: Minute (\* / 1), Hour (\*), Day (\*), Month (\*), Weekday (\*)
- Button: Update schedule
- Stats: Run count: 57540, Error count: 10

**Code Editor Panel:**

- File name: Netlify-HTML-time-scheduler
- Code content:

```
1 var request = require('request');
2
3 module.exports = function(cb) {
4   request({
5     method: 'POST',
6     url: 'https://api.netlify.com/build_hooks/5b51210902ed836d8a7073c4
7   });
8   cb(null, { msg: 'done '});
9 };
10
```

At the top of the browser window, there is a blue banner with the text: "Would you like the webtask.io experience in your own SaaS? Try Extend".

# WEBTASK.IO

The screenshot shows the Webtask Editor interface. A modal window is open for editing a task. The modal has a dark background and contains the following elements:

- Schedule Selector:** Five buttons labeled Minute, Hour, Day, Month, and Weekday. The Minute button is selected and contains the cron expression `*/1`. The other buttons contain a single asterisk `*`.
- Update Button:** A large cyan button labeled "Update schedule".
- Stats Section:** A section titled "Stats" showing "Run count: 455517" and "Error count: 34".
- Execution Status:** A green checkmark icon followed by the text "Completed a minute ago".

The background shows the editor interface with a code editor containing JavaScript code and a navigation bar with "Documentation" and "Extend" links.

**( DON'T DO THIS )**

*a quick look at*  
**TWO**  
**PRINCIPLES**





Are you being servered?

@philhawksworth

# PRE-RENDERING

# PRE-RENDERING

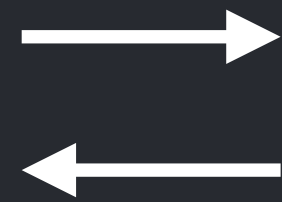
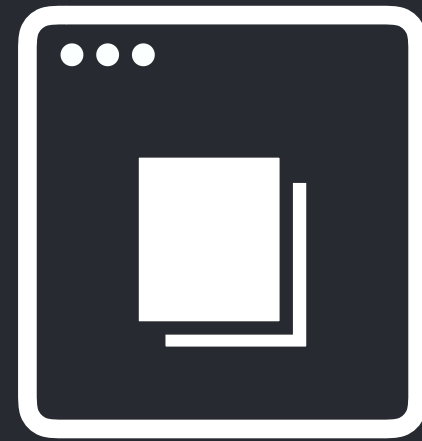
**SERVERLESS  
ON-DEMAND**

*motives for*  
**PRE-RENDERING**

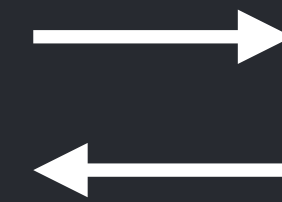
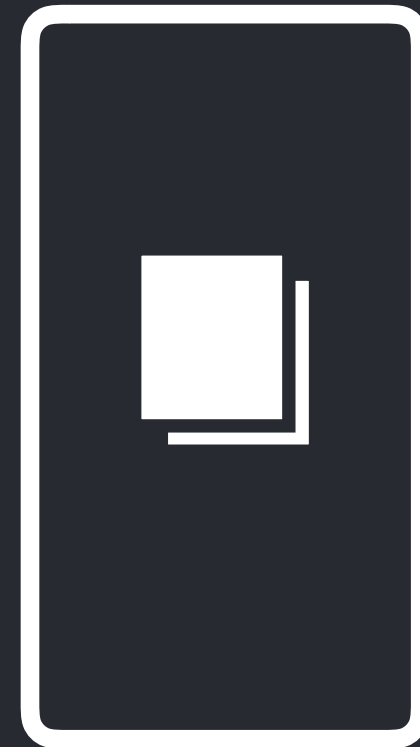
**DOING THE WORK NOW,  
SO YOUR SERVERS  
DON'T HAVE TO LATER**

**PUT DISTANCE BETWEEN  
THE COMPLEXITY  
AND THE USER**

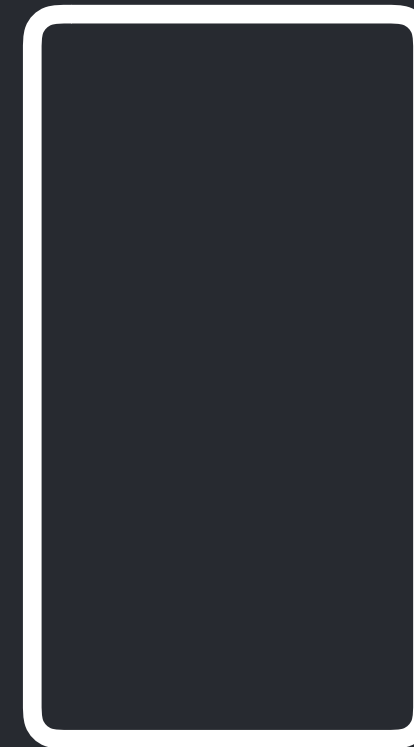
*traditional*



**CDN**



**LOAD BALANCER**



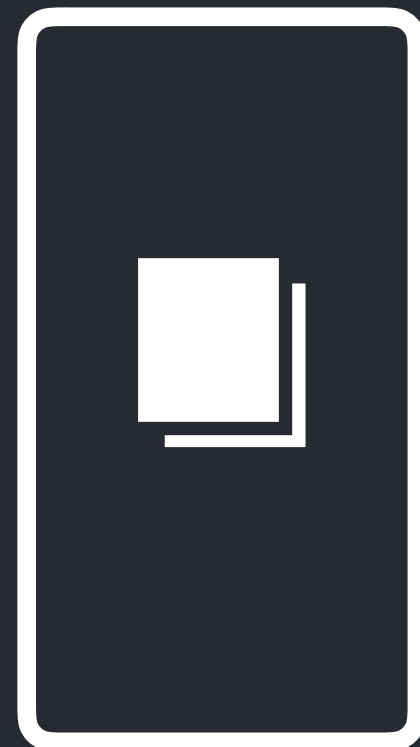
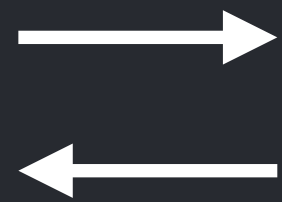
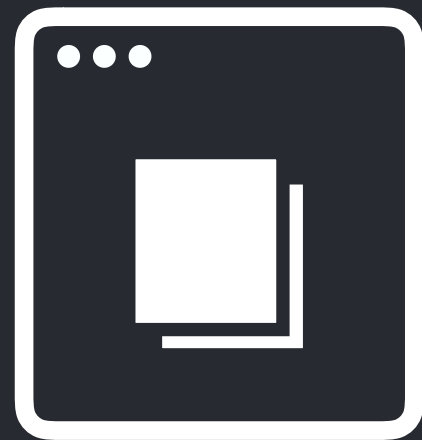
**WEB SERVERS**



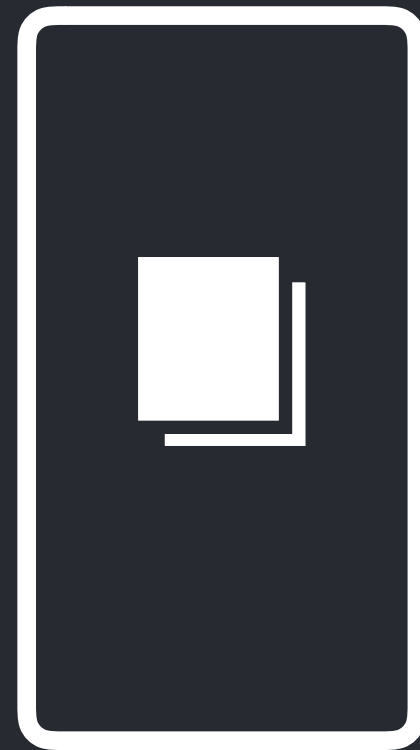
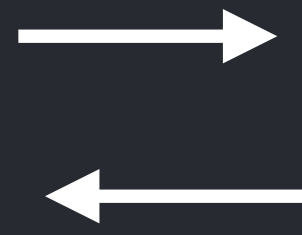
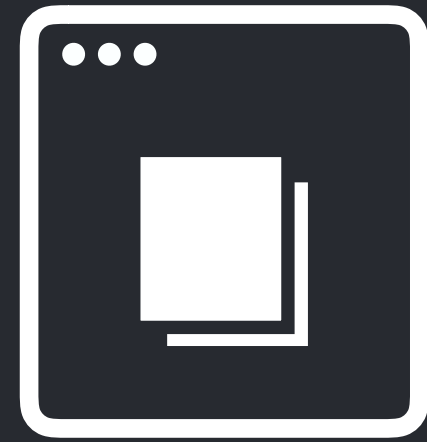
**DB SERVERS**



*jamstack*



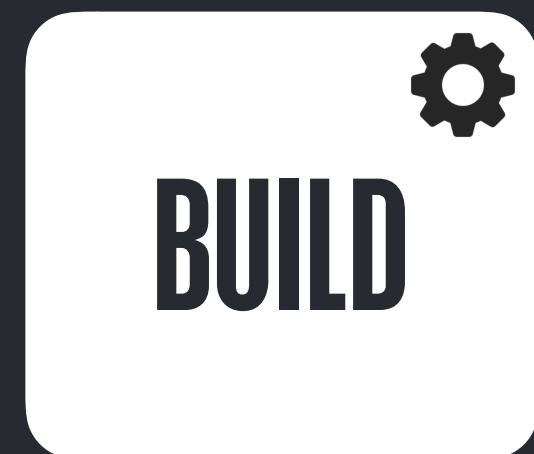
*jamstack*



REQUEST TIME

BUILD TIME

DEPLOYMENT



# JAMSTACK ADVANTAGES

---

**SECURITY**

---

**PERFORMANCE**

---

**SCALE**

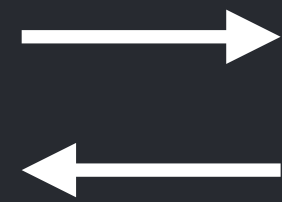
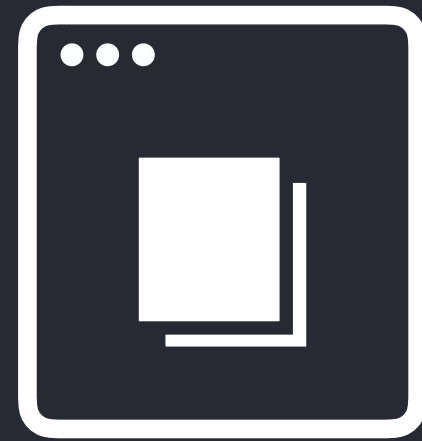




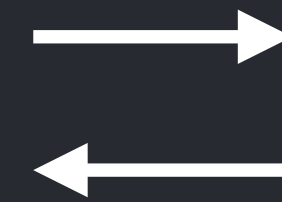
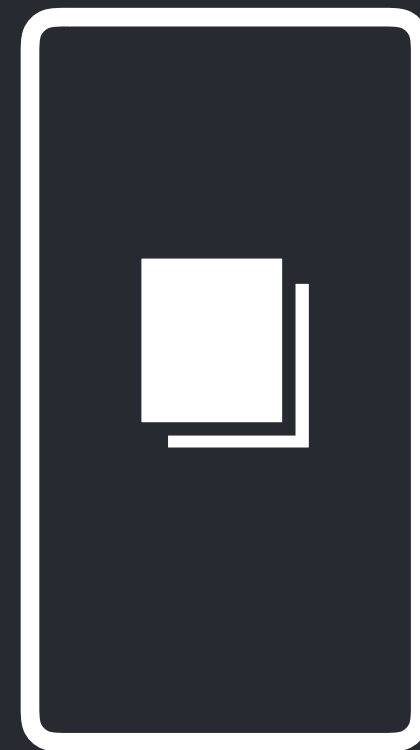
# SECURITY

# A GREATLY REDUCED SURFACE AREA

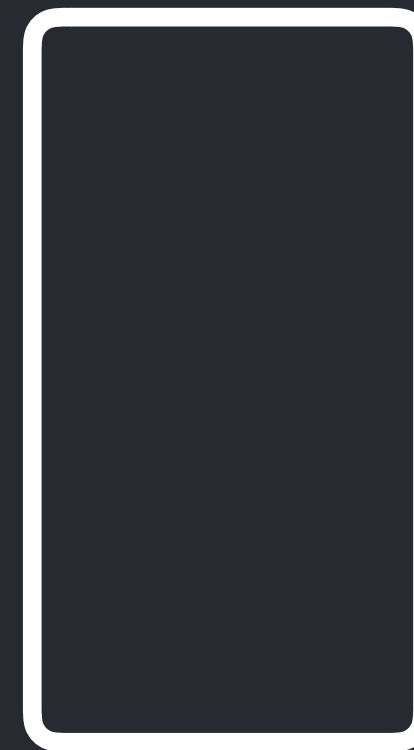
*traditional*



CDN



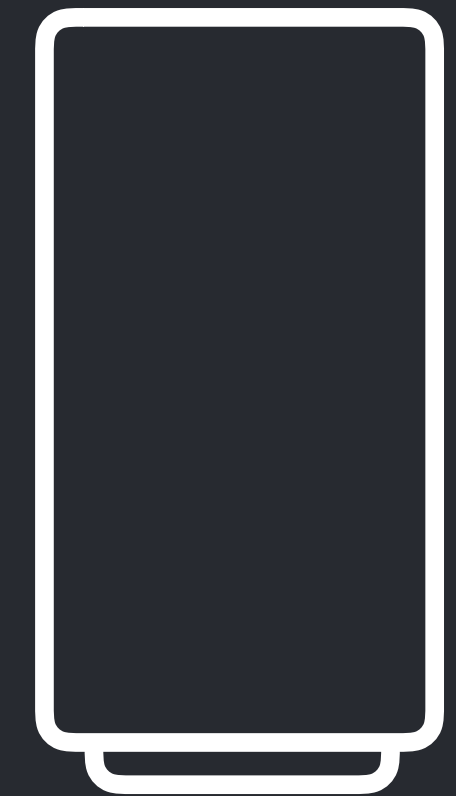
LOAD BALANCER



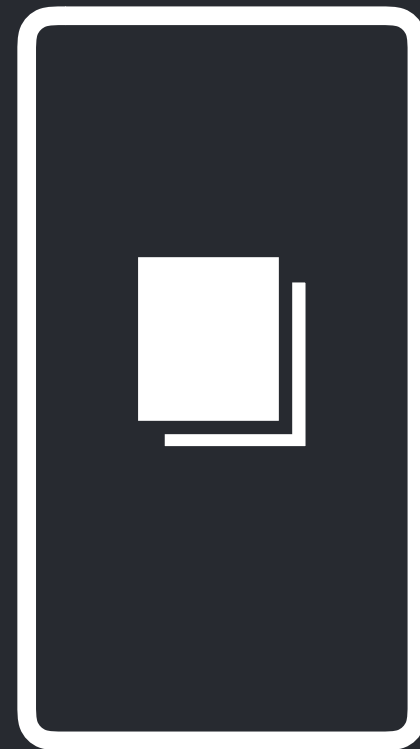
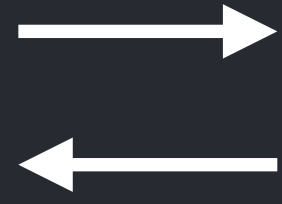
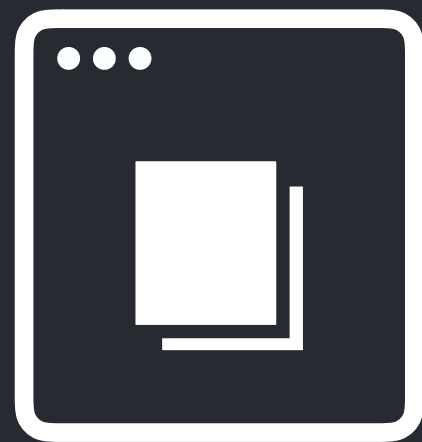
WEB SERVERS



DB SERVERS



*jamstack*



# FAR FEWER MOVING PARTS TO ATTACK



# PERFORMANCE

**TRADITIONAL STACKS**  
**ADD STATIC LAYERS**  
**IN ORDER TO IMPROVE**  
**PERFORMANCE**

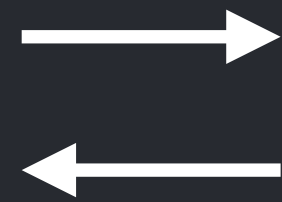
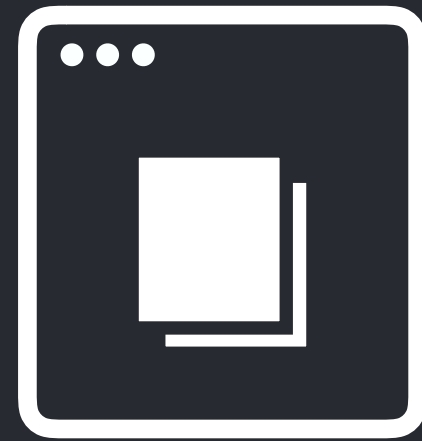


Are you being servered?

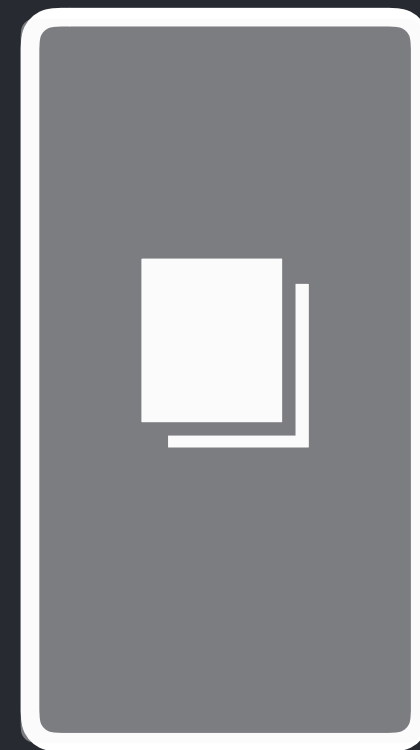
@philhawksworth

# CACHING GALORE

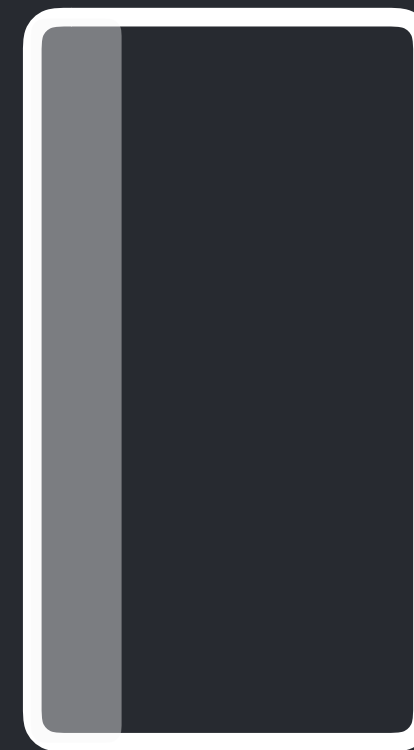
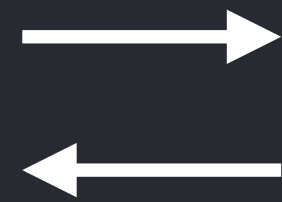
*traditional*



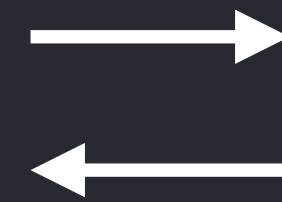
CDN



LOAD BALANCER



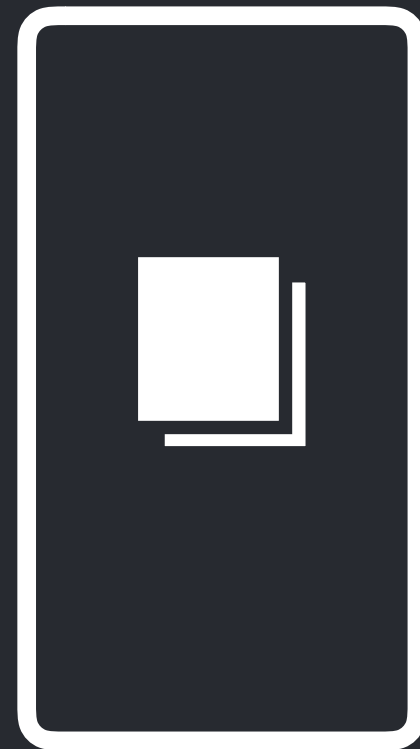
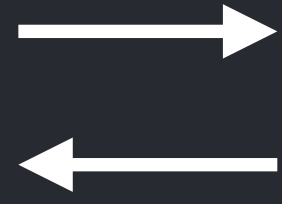
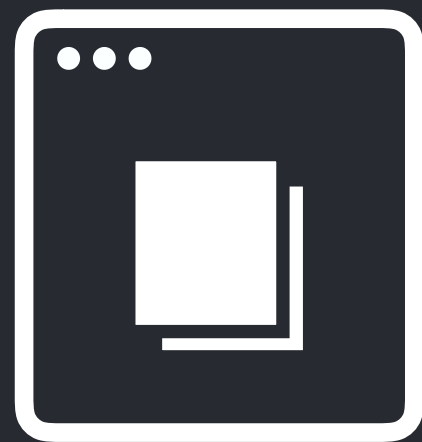
WEB SERVERS



DB SERVERS



*jamstack*



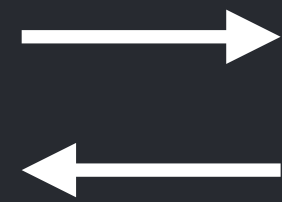
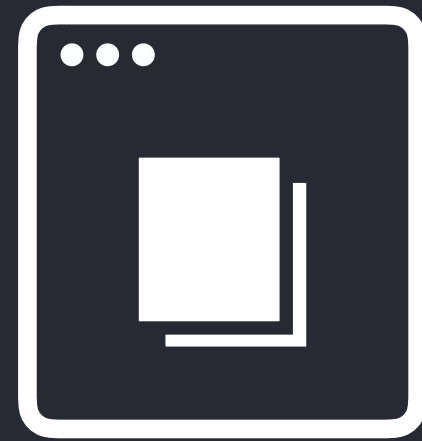




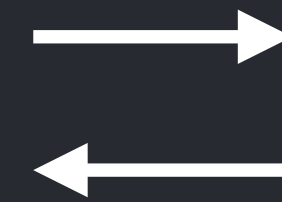
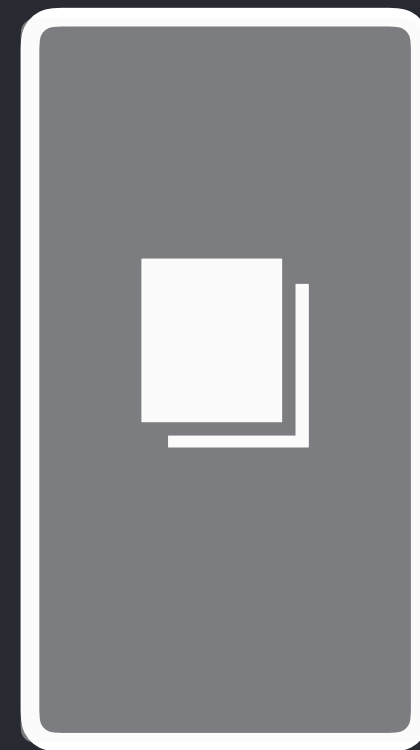
# SCALE

**TRADITIONAL STACKS  
ADD INFRASTRUCTURE  
IN ORDER TO SCALE**

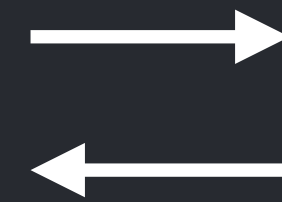
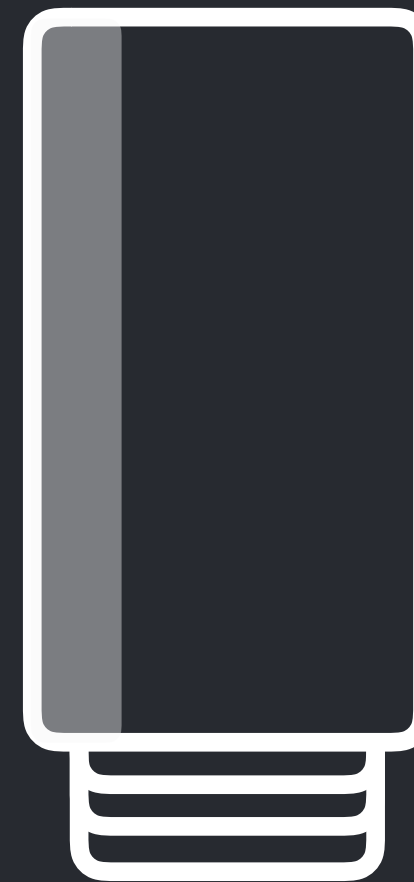
*traditional*



CDN



LOAD BALANCER



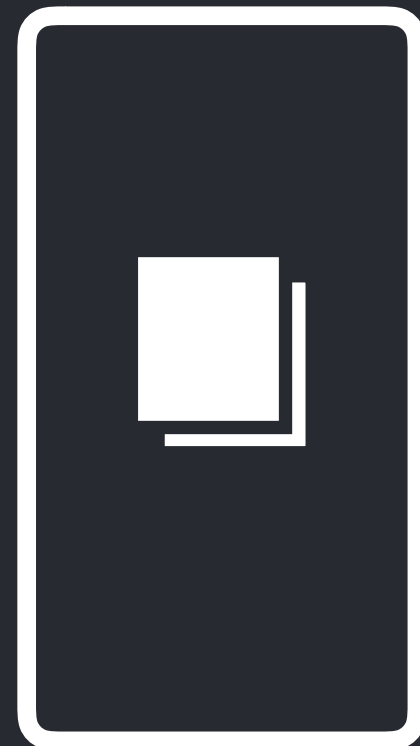
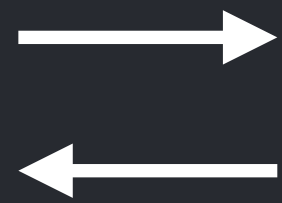
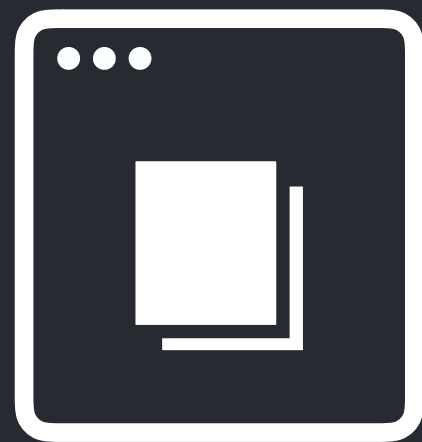
WEB SERVERS



DB SERVERS



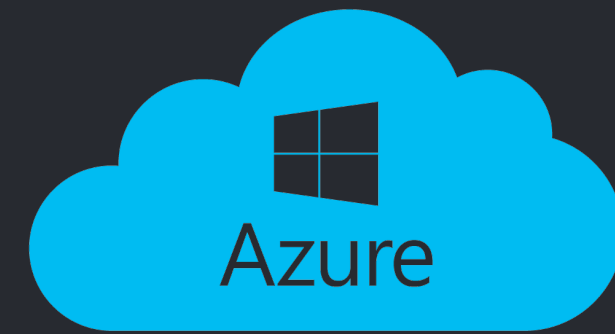
*jamstack*



# DECOUPLING

# GET TO THE ~~CHOPPER~~ CDN





# PRE-RENDERING

**SERVERLESS  
ON-DEMAND**

# SERVERLESS ON-DEMAND



# jamstack

*A modern architecture —*

*Create fast and secure sites and dynamic apps  
with **J**avaScript, **A**PIs, and pre-rendered  
**M**arkup, served without web servers*

---

**I ABSOLUTELY POSITIVELY  
NEED A LITTLE LOGIC  
ON A SERVER**

**MANY OF US, SOMETIMES**

A stylized illustration of a superhero in a black suit with a red cape, running through a storm. The background is a mix of brown, tan, and grey tones, with dark, jagged shapes representing rain or debris. The superhero is in a dynamic, forward-leaning pose, with one foot on the ground and the other lifted. The overall mood is one of urgency and action.

# SERVERLESS TO THE RESCUE

**LET'S JUST RECREATE  
EVERYTHING IN  
SERVERLESS FUNCTIONS**

**THAT VOICE OF TEMPATION**

# jamstack

*A modern architecture —*

*Create fast and secure sites and dynamic apps  
with **JavaScript**, **APIs**, and **pre-rendered  
Markup**, served without web servers*

*free*  
**FROM INFRASTRUCTURE**

*but not free*  
**FROM DYNAMIC LOGIC**

# AUGMENTATION

# ENHANCEMENT



# STATIC FIRST

MARKUS SCHORK, UNILEVER

# AN EXAMPLE

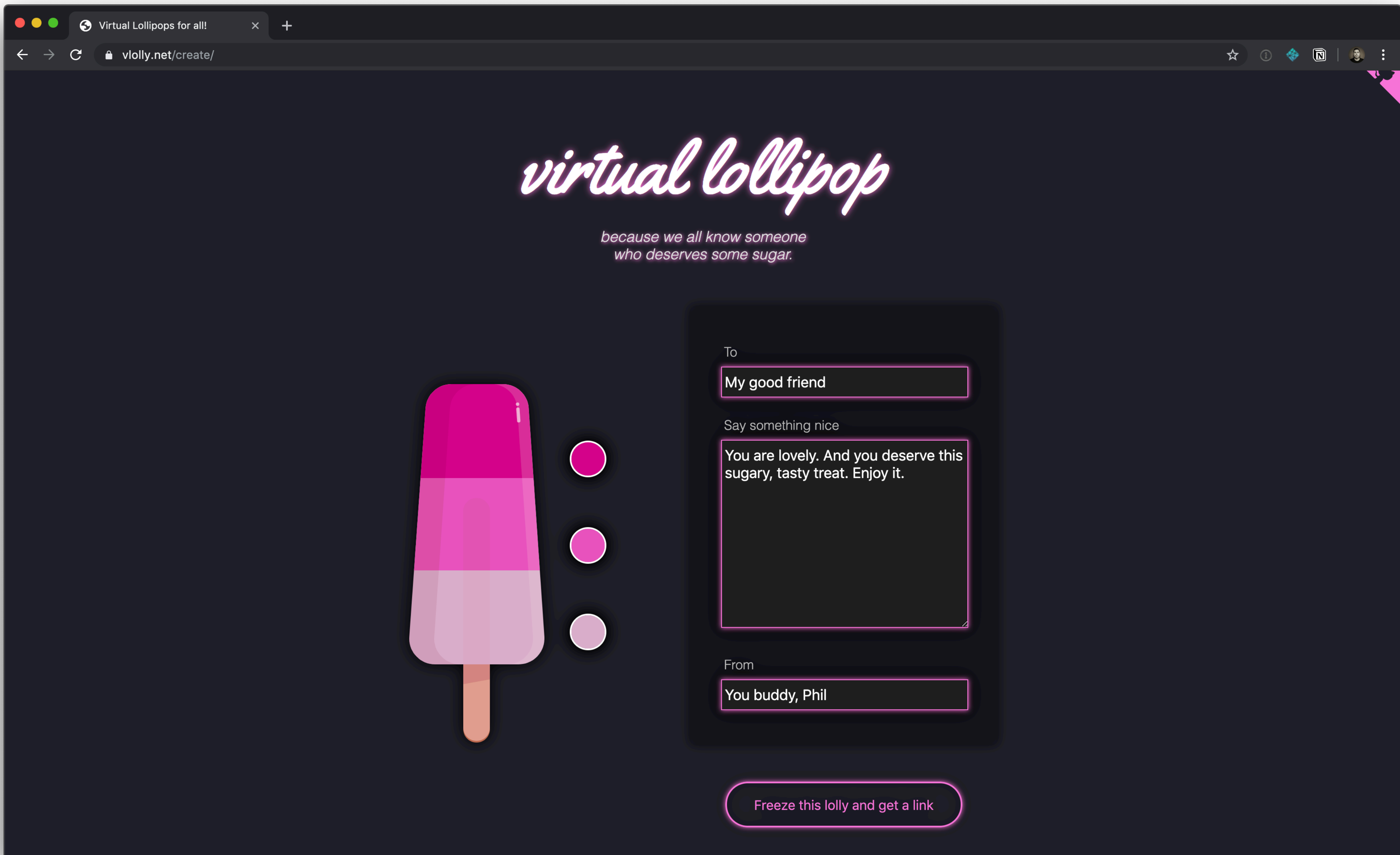


# virtual lollipop

*because we all know someone  
who deserves some sugar.*



Make a new lollipop to send to a friend

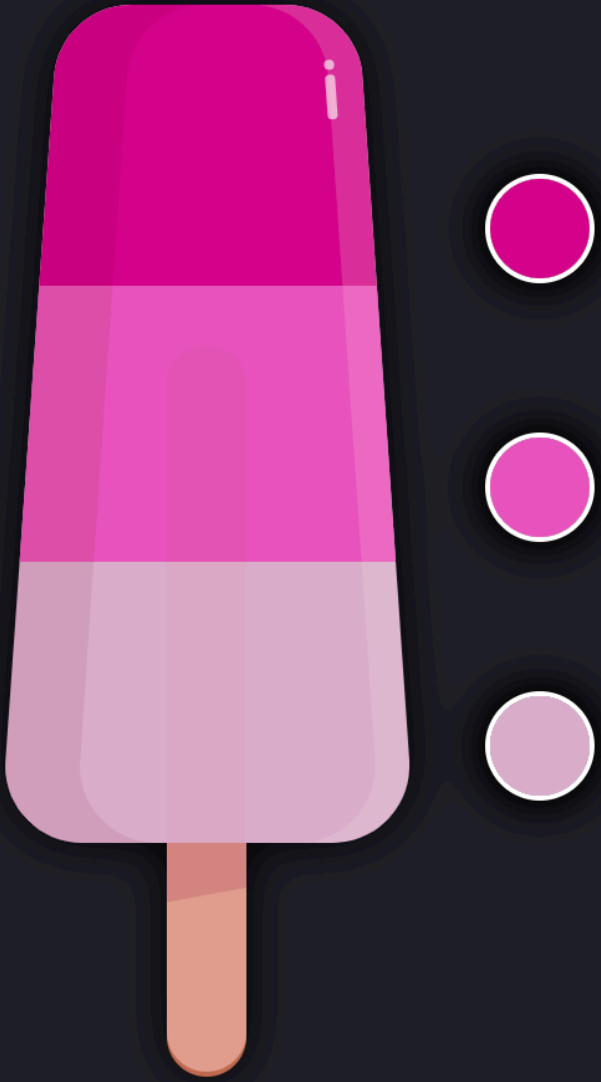


Virtual Lollipops for all! x +

vlolly.net/create/ ☆ ⓘ ⚙️ 📄 👤 ⋮

# virtual lollipop

*because we all know someone  
who deserves some sugar.*



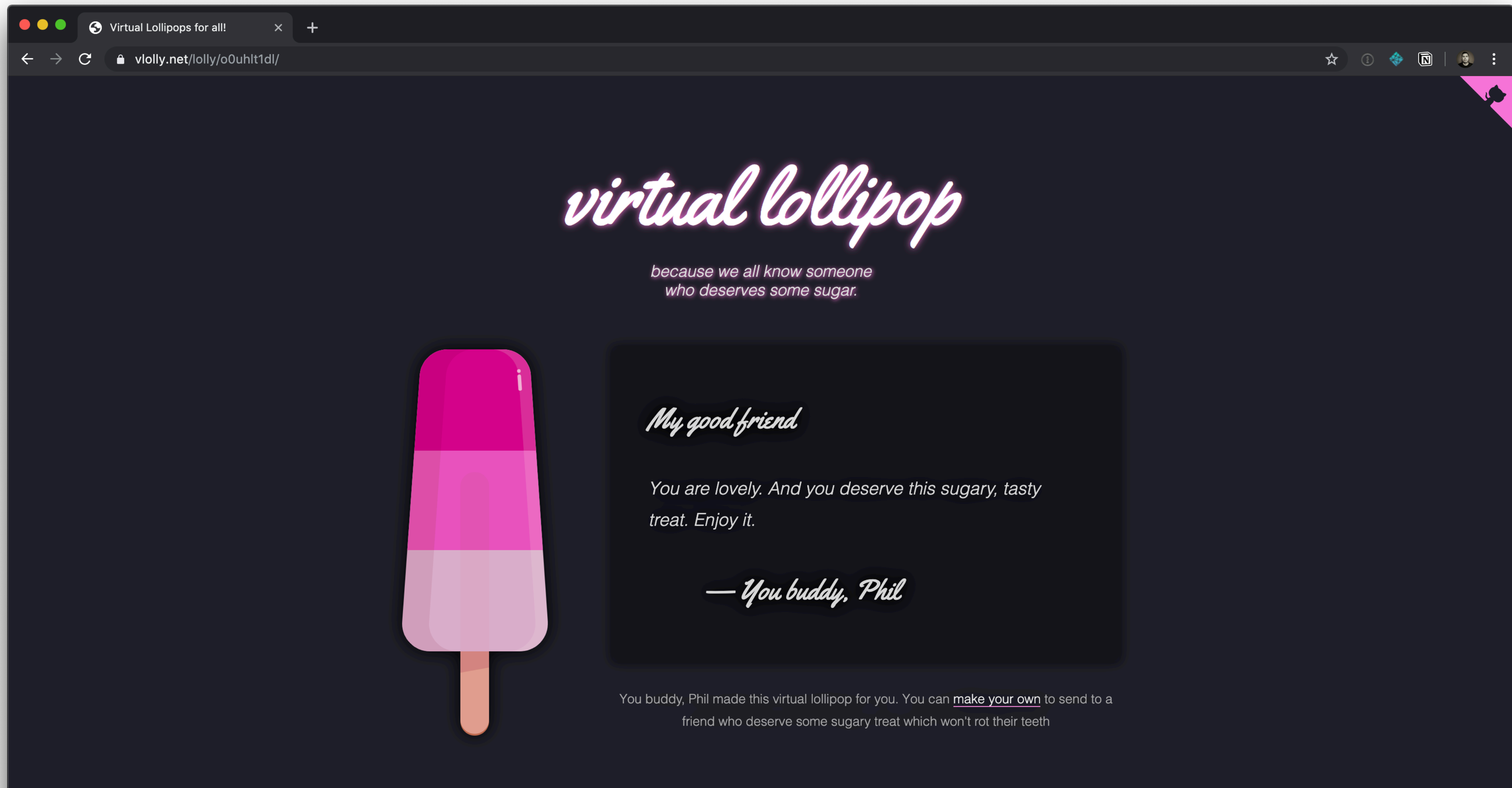
To

Say something nice

From

Freeze this lolly and get a link

# VLOLLY.NET/LOLLY/3hmtt2r3b

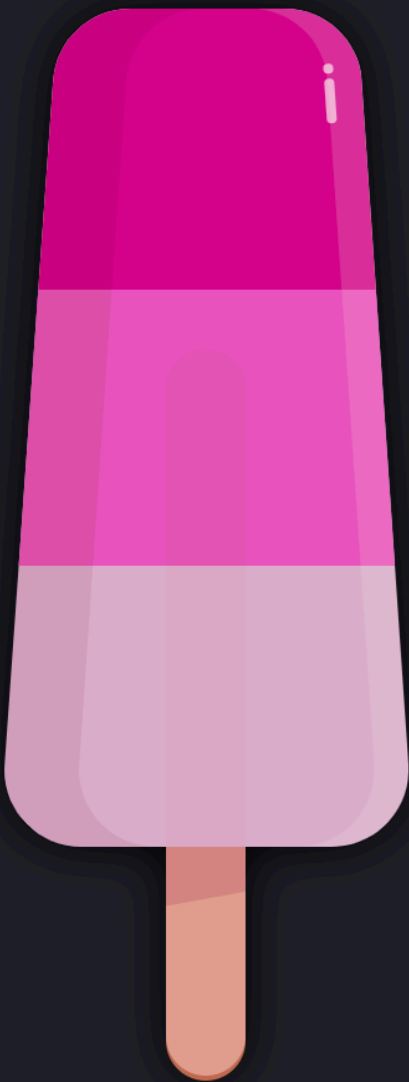


Virtual Lollipops for all! x +

volly.net/lolly/o0uht1dl/ ☆ ⓘ 🔍 📄 👤 ⋮

## *virtual lollipop*

*because we all know someone  
who deserves some sugar.*



*My good friend*

*You are lovely. And you deserve this sugary, tasty  
treat. Enjoy it.*

*— You buddy, Phil*

You buddy, Phil made this virtual lollipop for you. You can [make your own](#) to send to a friend who deserve some sugary treat which won't rot their teeth



Are you being servered?

@philhawksworth

*requirements*

# PRE-GENERATED PAGES WITH REAL URLS

(EVEN FOR THE USER GENERATED CONTENT)

# DATA STORED IN A DATABASE

(BUT DON'T MAKE ME A DBA)

# INSTANT ACCESS TO NEW CONTENT

(WITHOUT WAITING FOR A REBUILD)



Are you being servered?

@philhawksworth

*tools*



# PRE-GENERATED PAGES WITH REAL URLS

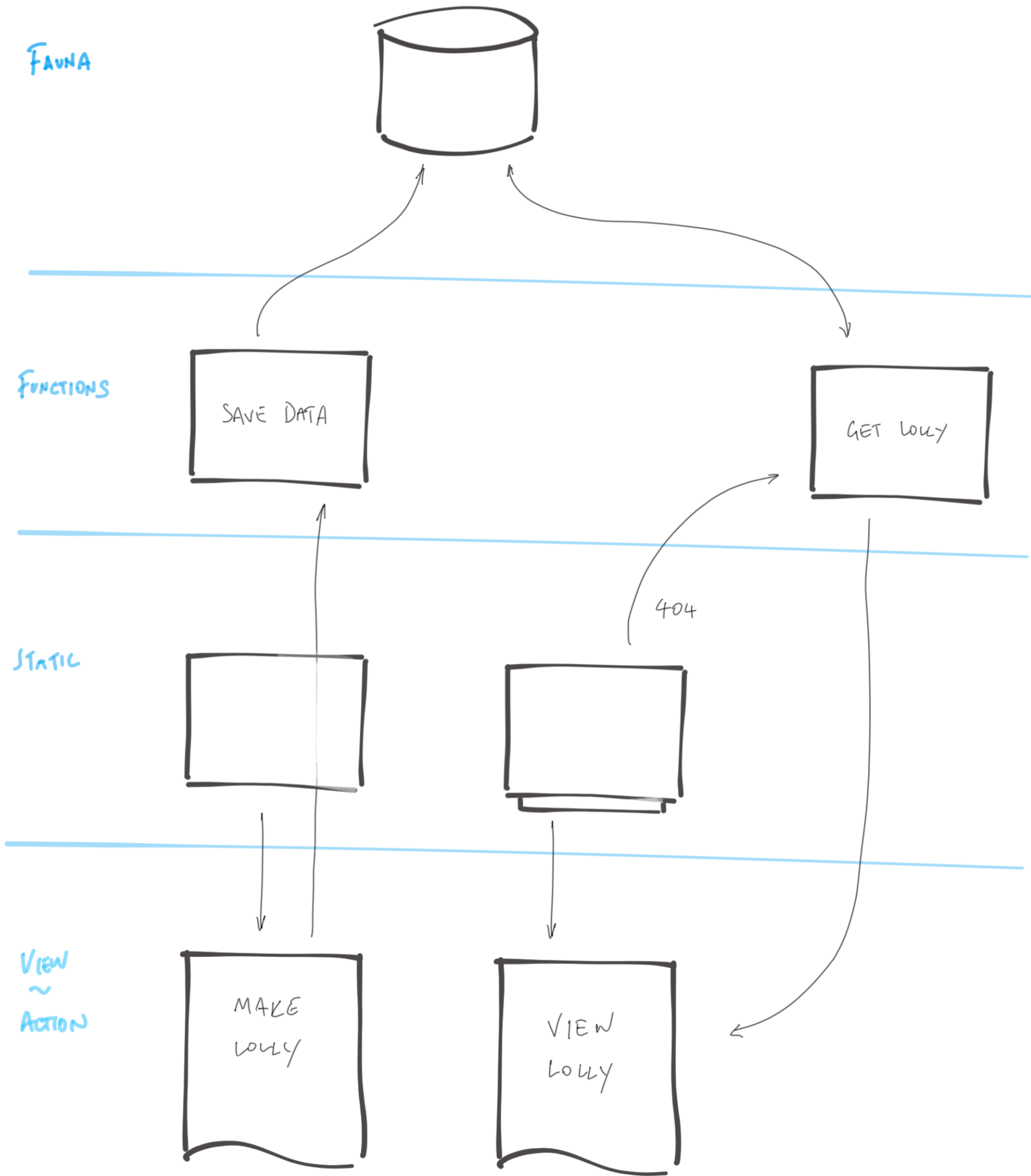
**ELEVENTY**

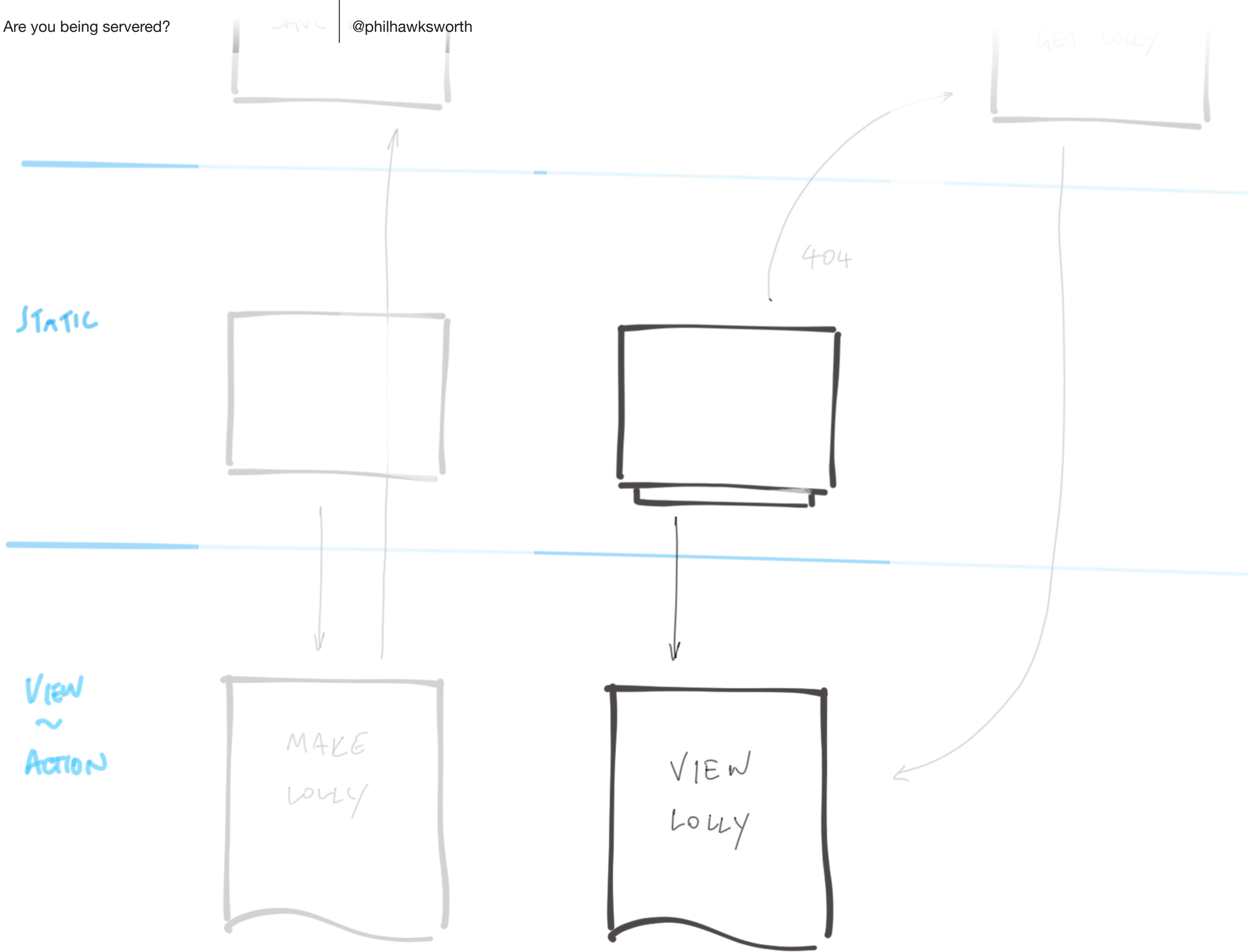
# DATA STORED IN A DATABASE

**FAUNADB**

# INSTANT ACCESS TO NEW CONTENT

**NETLIFY FUNCTIONS**



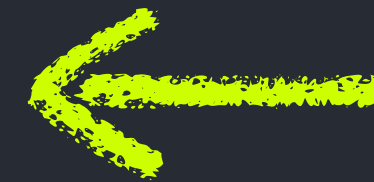


```
const faunadb = require('faunadb');
const q = faunadb.query;
const client = new faunadb.Client({
  secret: process.env.FAUNADB_SERVER_SECRET
});
```

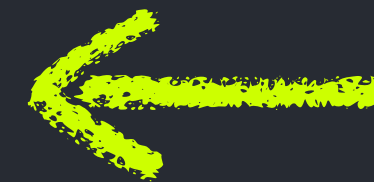


**AUTHENTICATE  
WITH THE DB**

```
module.exports = () => {
  return new Promise((resolve, reject) => {
    client.query(
      q.Paginate(q.Match(q.Ref("indexes/all_lollies")))
    ).then((response) => {
      const lollies = response.data;
      const getAllDataQuery = lollies.map((ref) => {
        return q.Get(ref);
      });
      return client.query(getAllDataQuery).then((ret) => {
        resolve(ret);
      });
    }).catch((error) => {
      reject(error);
    });
  })
}
```



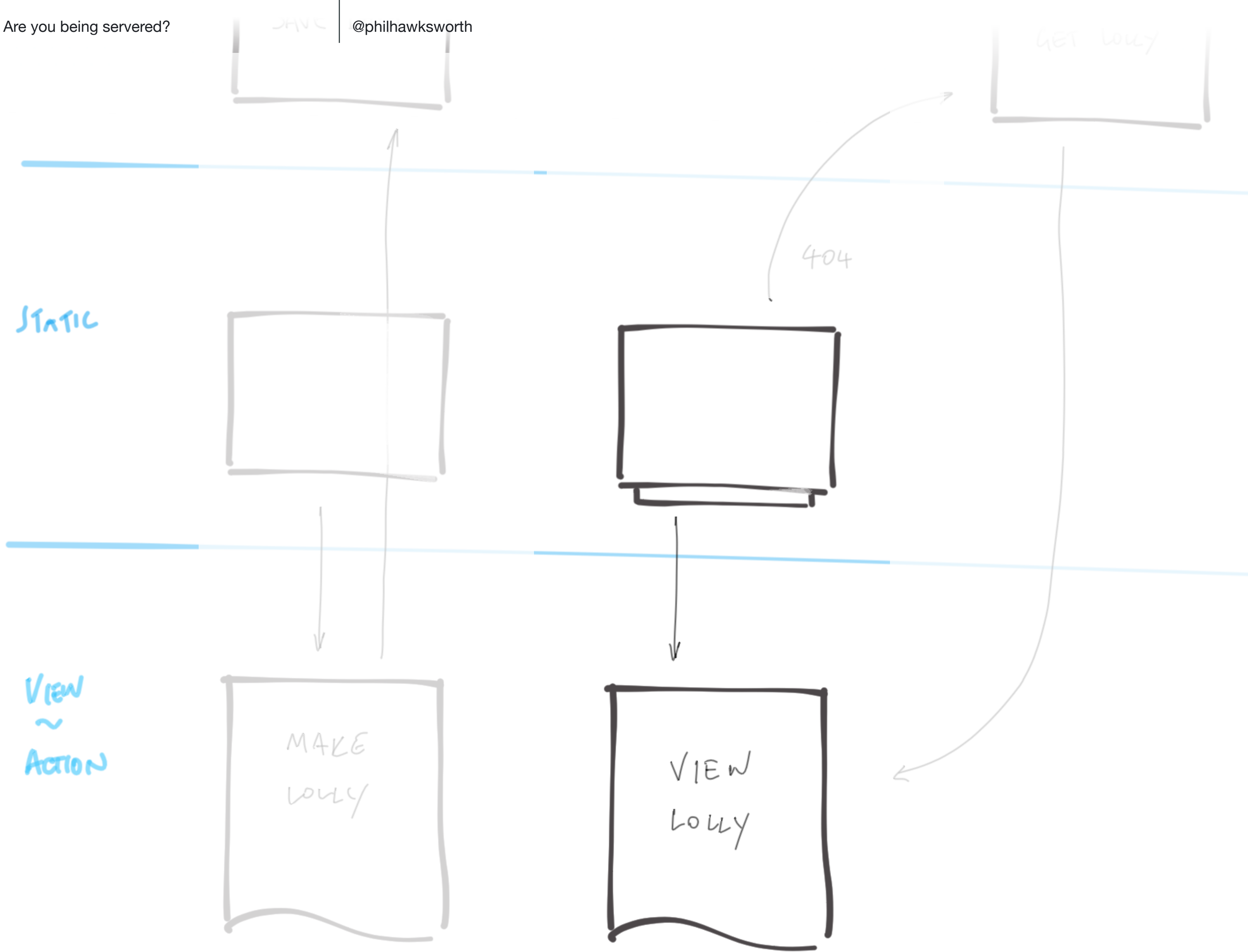
**QUERY FOR ALL  
THE LOLLIES... AND  
THEIR DATA**



**RESOLVE THE  
PROMISE OF LOLLIES**

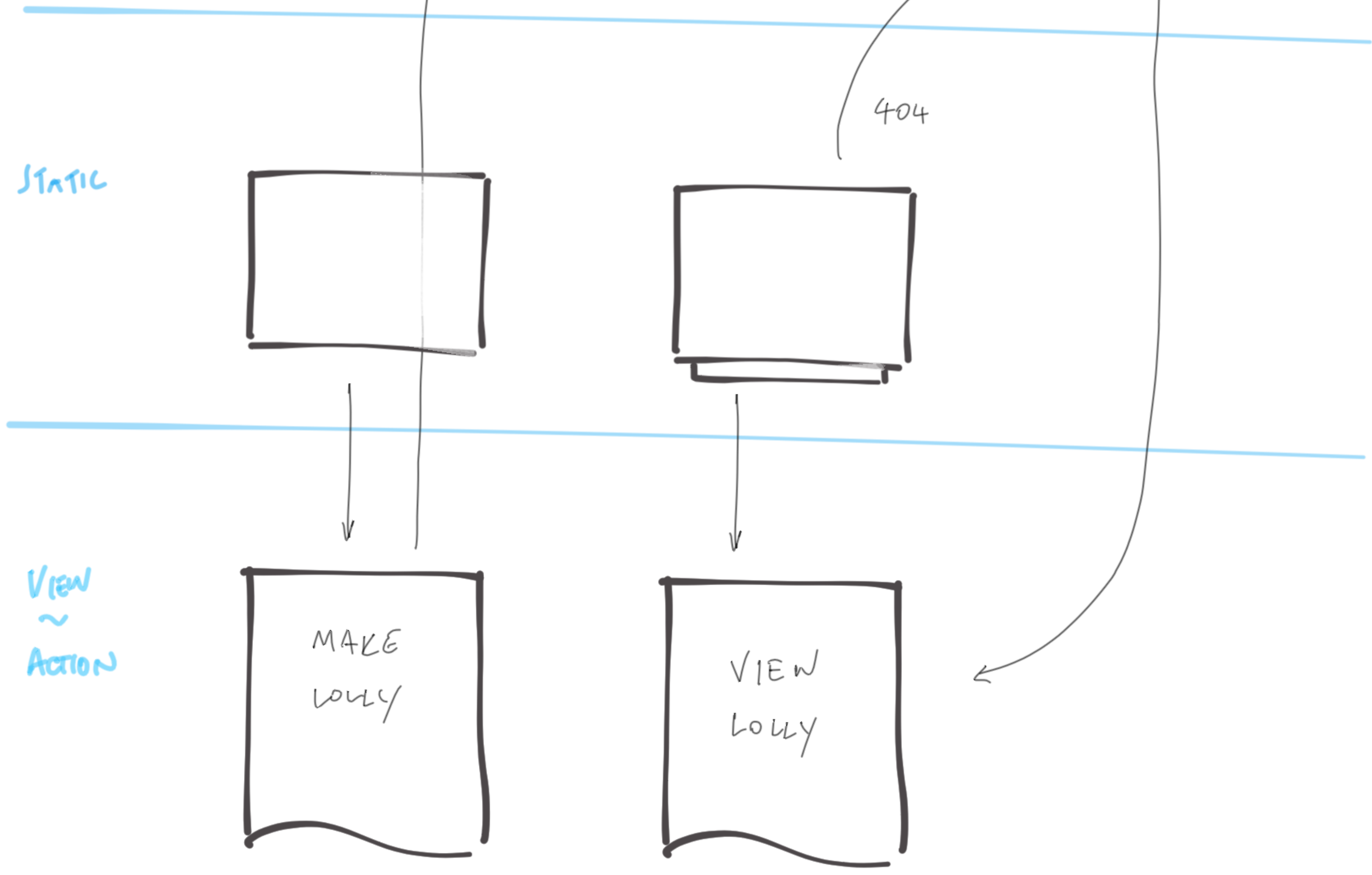


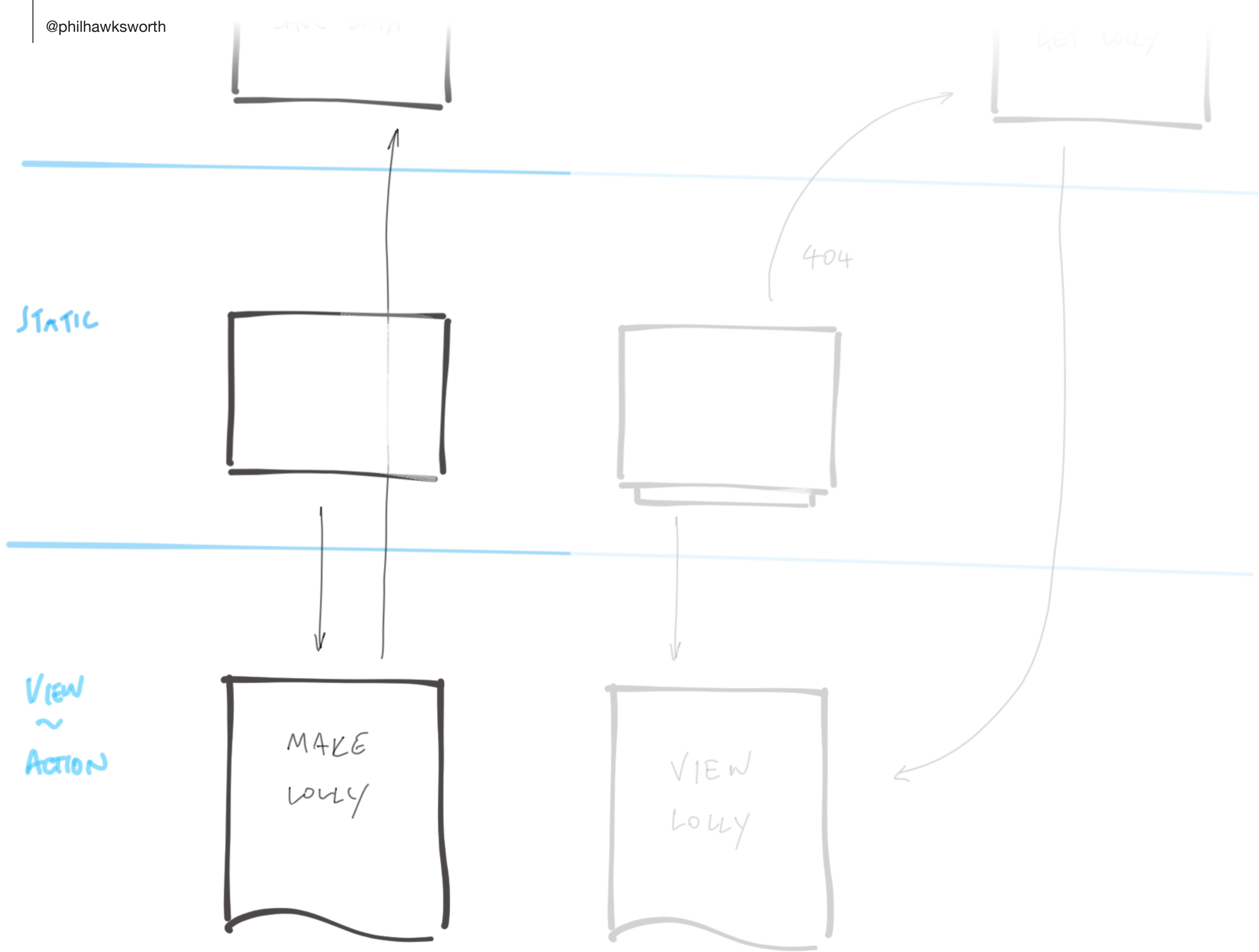
**CATCH ANY ERRORS**



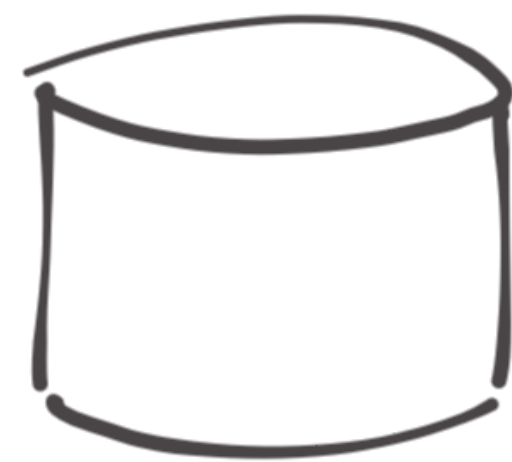
JAVC | @philhawksworth

GET LOLLY





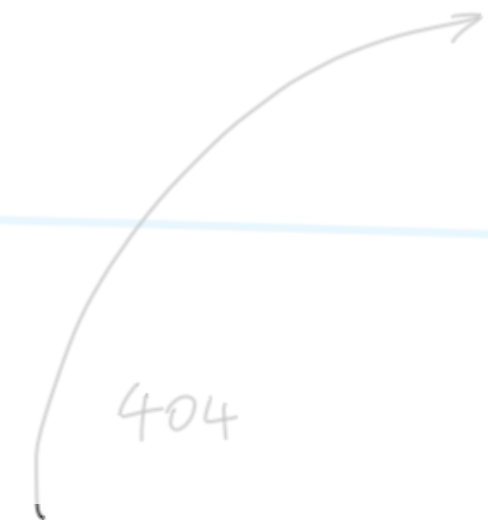
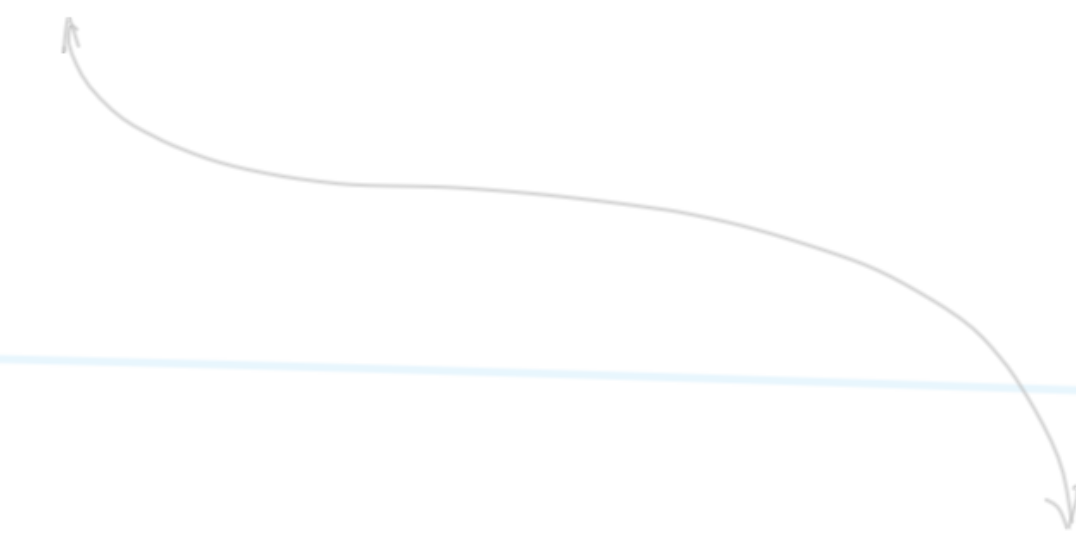
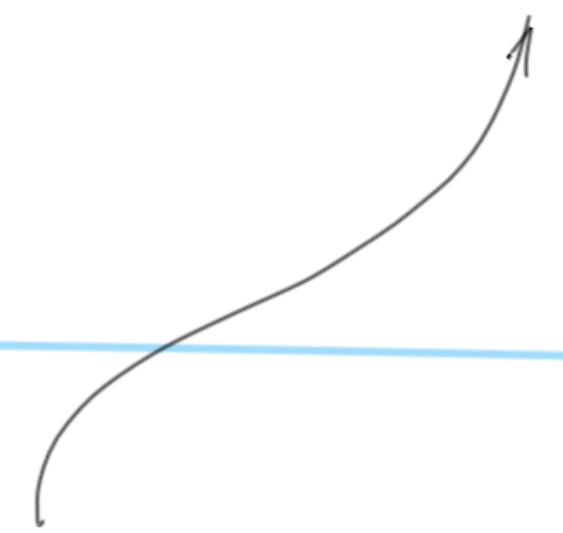
FAUNA



FUNCTIONS

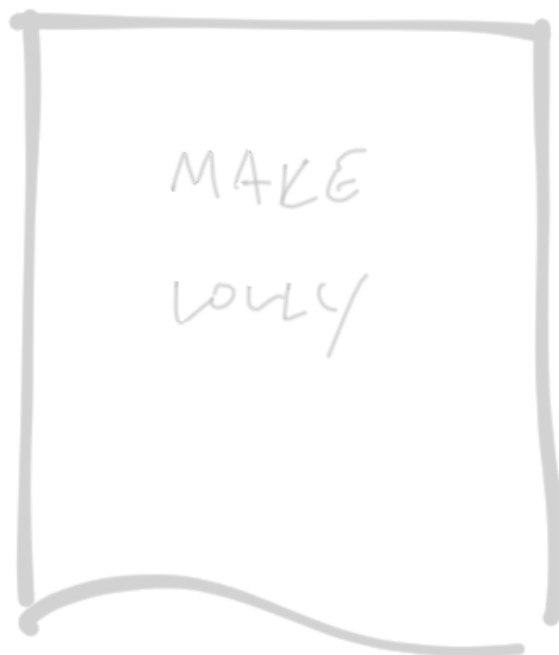


STATIC

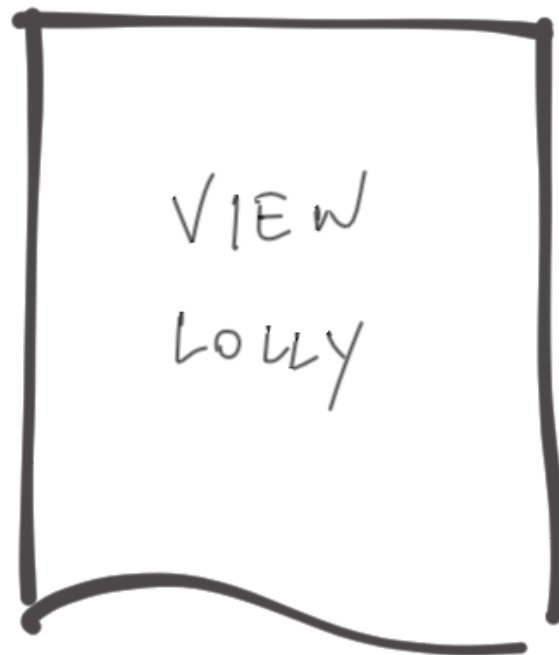
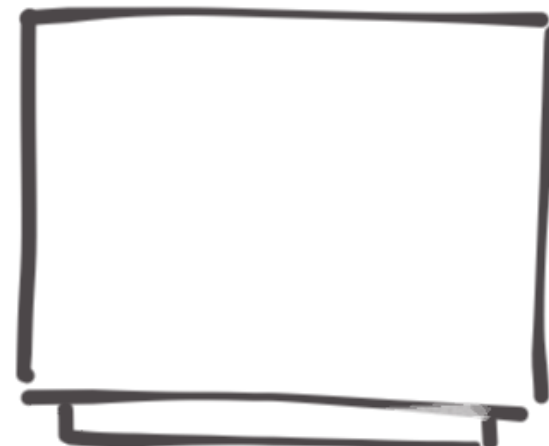




Static



View  
~  
Action



404





Are you being servered?

@philhawksworth

# 404

# TRIGGERS AND AUTOMATION

- General
- Build & deploy**
- Continuous Deployment
- Environment
- Post processing
- Deploy notifications
- Domain management
- Analytics
- Functions
- Identity
- Forms
- Large Media
- Access control

Branch deploys: **Deploy only the production branch**

[Learn more about deploy contexts in the docs](#)

Edit settings

### Build hooks

Build hooks give you a unique URL you can use to trigger a build.

New lolly in the freezer: **https://api.netlify.com/build\_hooks/5d46fa20da4a1b70047f2f04**

[Learn more about build hooks in the docs](#)

Add build hook

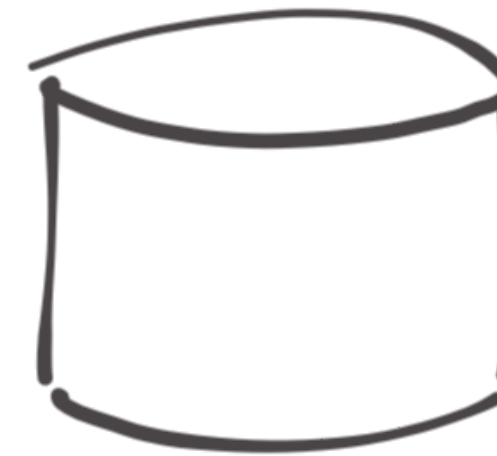
### Build image selection

BETA

Select a different build image to change the operating system and supported software versions in the environment where our buildbot builds your site.

Build image: **Ubuntu Xenial 16.04 (default)**

FAUNA



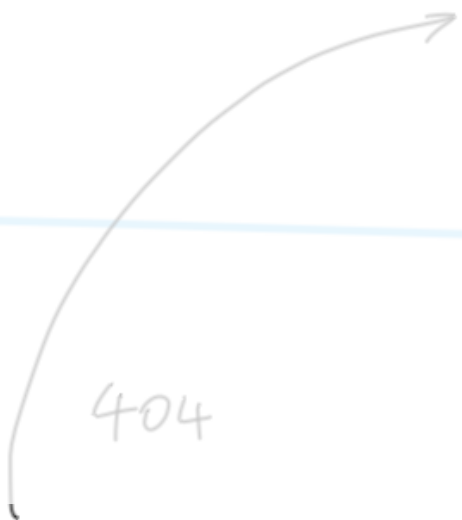
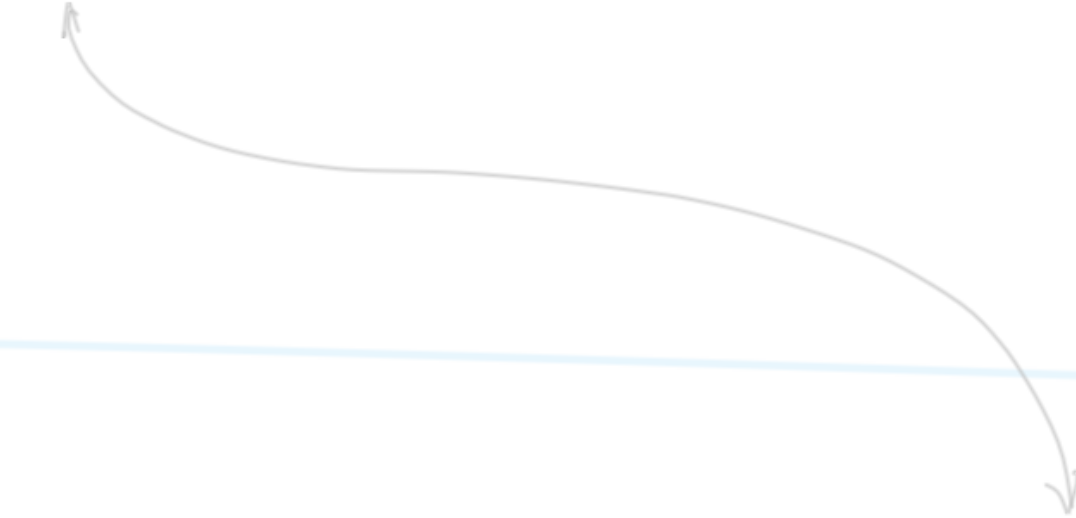
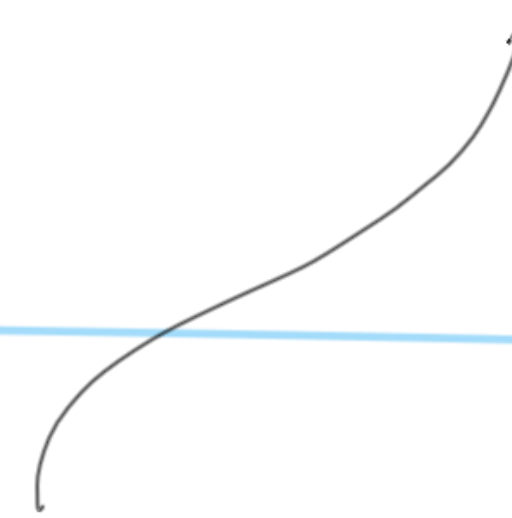
FUNCTIONS



STATIC



404



```
const faunadb = require('faunadb');
const shortid = require('shortid');
const querystring = require('querystring');
const axios = require('axios');
const q = faunadb.query
const client = new faunadb.Client({
  secret: process.env.FAUNADB_SERVER_SECRET
})
exports.handler = (event, context, callback) => {
  const data = querystring.parse(event.body);
  const uniquePath = shortid.generate();
  data.lollyPath = uniquePath;
  const lolly = {
    data: data
  };
  client.query(q.Create(q.Ref('classes/lollies'), lolly))
    .then((response) => {
      axios.post('https://api.netlify.com/build_hooks/5d46fa20da4a1b70047f2f04')
        .then(function (response) {
          console.log(response);
        })
        .catch(function (error) {
          console.log(error);
        });
      return callback(null, {
        body: JSON.stringify(response),
        statusCode: 302,
        headers: {
          Location: `/lolly/${uniquePath}`,
        }
      });
    }).catch((error) => {
      return callback(null, {
        statusCode: 400,
        body: JSON.stringify(error)
      });
    });
}
```

← AUTHENTICATE WITH THE DB

← POST THE DATA

← REBUILD

← SEND THE USER TO CHECK THE FREEZER FOR A LOLLY

**WHAT THE DICKENS DOES  
"CHECK THE FREEZER FOR  
A LOLLY" MEAN?**

**ALL OF YOU, POSSIBLY**

**CHECK FOR A  
PRE-GENERATED  
LOLLY PAGE**

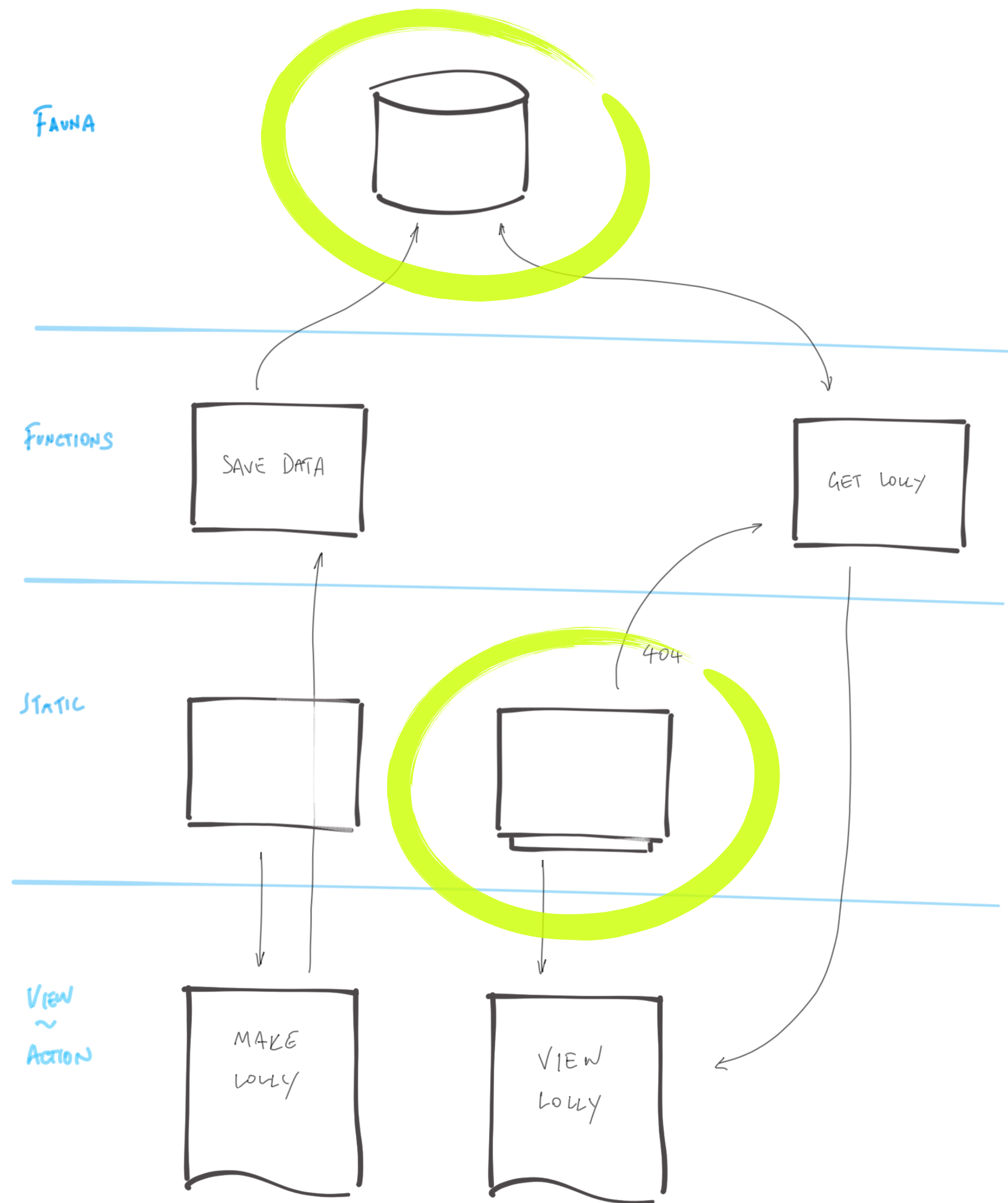


**SERVE THE  
STATIC  
PAGE**

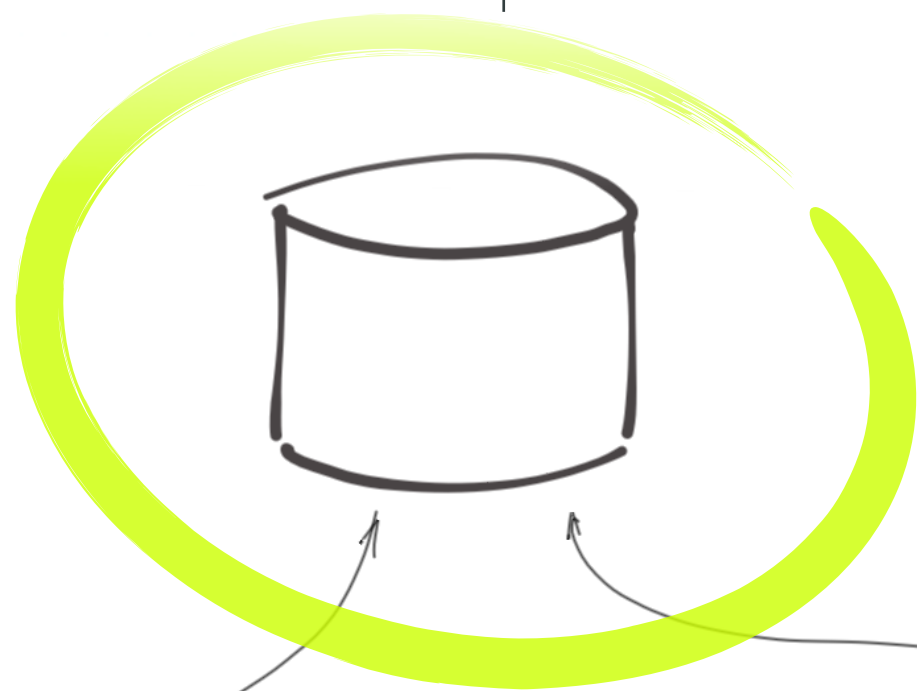


**RENDER A  
DYNAMIC  
VIEW**





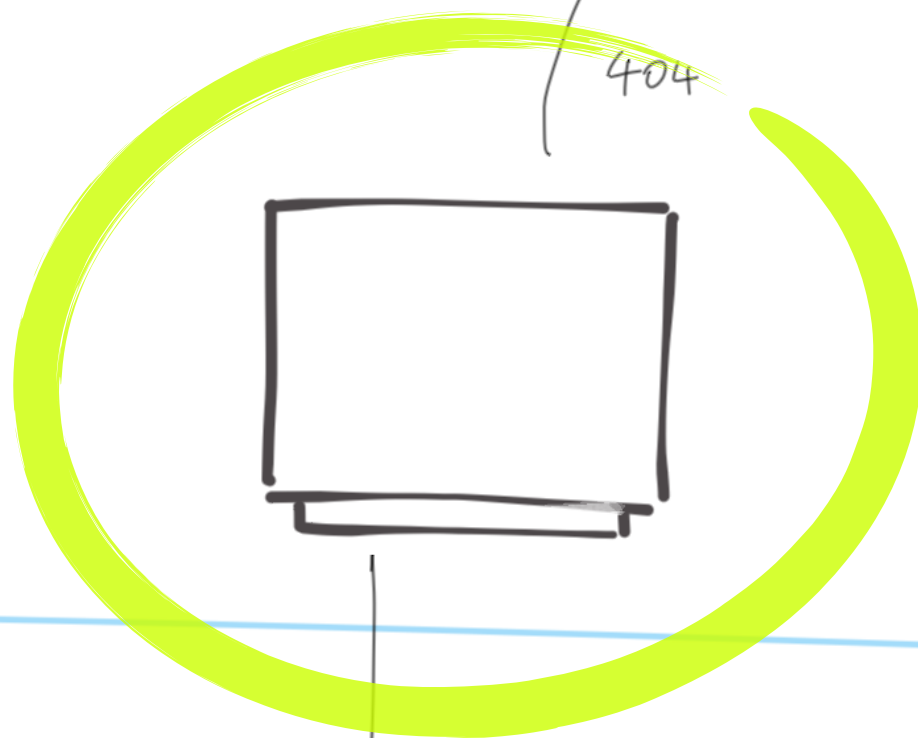
FAUNA



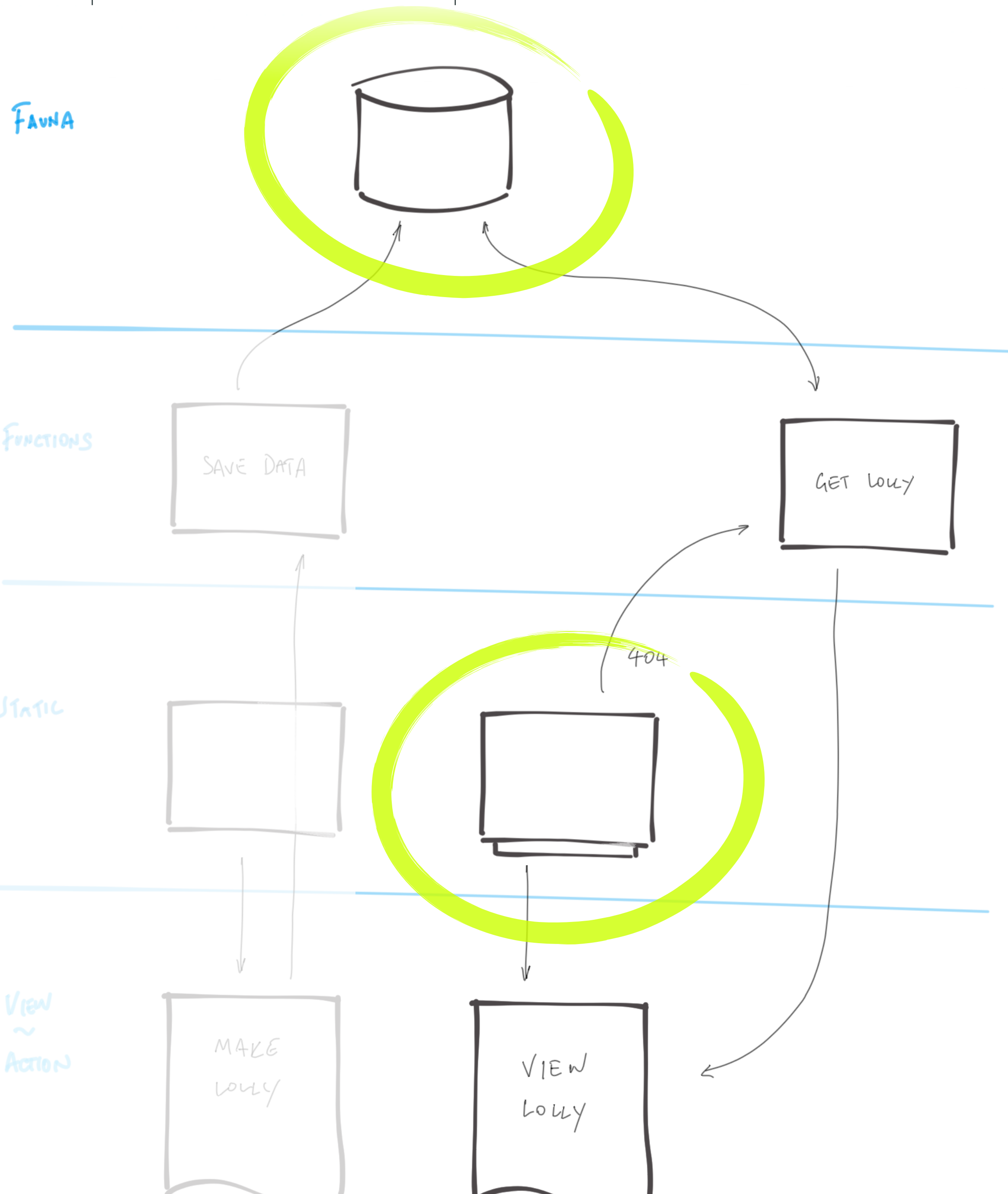
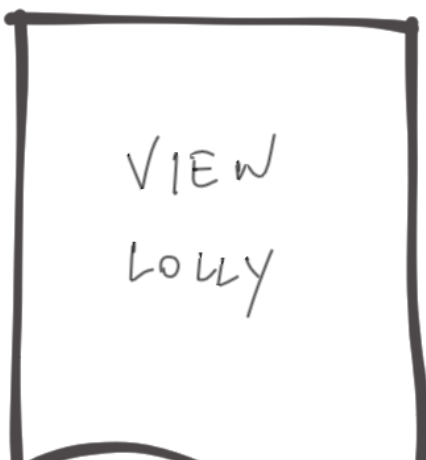
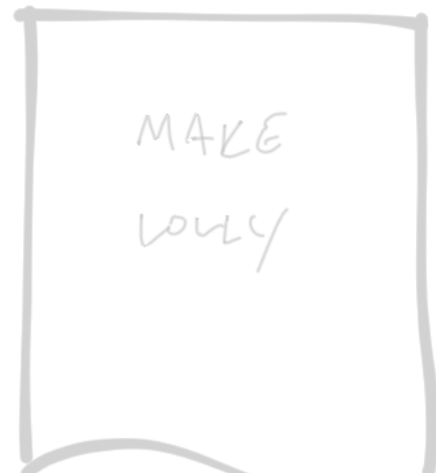
FUNCTIONS



STATIC



VIEW  
~  
ACTION



```
const faunadb = require('faunadb');
const pageTemplate = require('./lollyTemplate.js');

const q = faunadb.query;
const client = new faunadb.Client({
  secret: process.env.FAUNADB_SERVER_SECRET
});

exports.handler = (event, context, callback) => {
  const path = event.queryStringParameters.id.replace("/", "");
  client.query(
    q.Get(q.Match(q.Index("lolly_by_path"), path))
  ).then((response) => {
    return callback(null, {
      statusCode: 200,
      body: pageTemplate(response.data)
    });
  }).catch((error) => {
    return callback(null, {
      body: JSON.stringify(error),
      statusCode: 301,
      headers: {
        Location: `/melted/index.html`,
      }
    });
  });
};
```

GET PAGE AS A  
JAVASCRIPT  
TEMPLATE LITERAL

AUTHENTICATE  
WITH THE DB

GET THE  
LOLLY DATA

SERVE LOLLY VIEW

NO LOLLY. BREAK  
THE BAD NEWS

*a quick demo?*



Are you being servered?

@philhawksworth

*enablers*

# CUSTOM 404 ROUTING



```
[[redirects]]  
  from = "/lolly/*"  
  to = "/.netlify/functions/showLolly?id=:splat"  
  status = 302  
  force = "false"
```

# EVENTS, TRIGGERS AND AUTOMATION



# FUNCTIONS AS A SERVICE

# DATABASE AS A SERVICE

# PRE-RENDERING

**SERVERLESS  
ON-DEMAND**



Are you being servered?

@philhawksworth

# A SHIFT

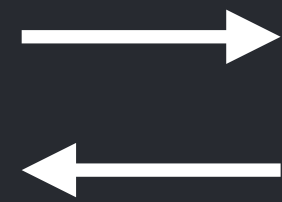
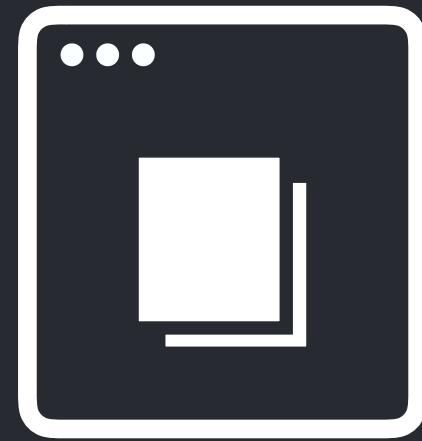
*from*

**SEEKING TO OPTIMIZE BY  
ADDING STATIC LAYERS**

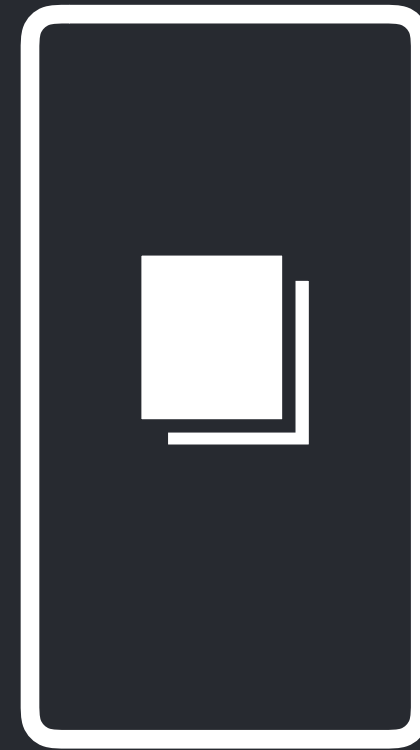
*to*

**STATIC FIRST**  
**ENHANCING IF REQUIRED**

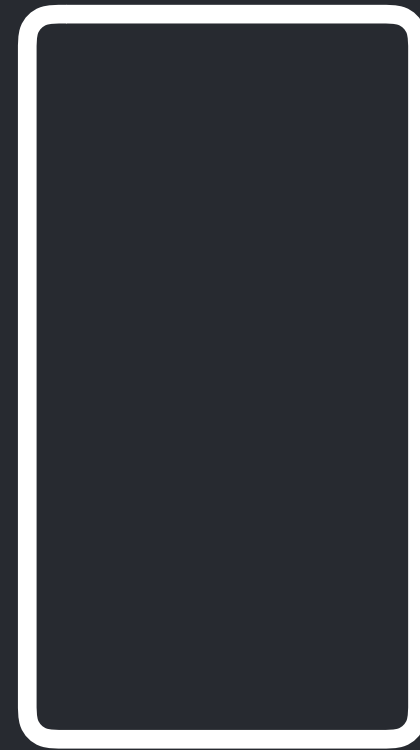
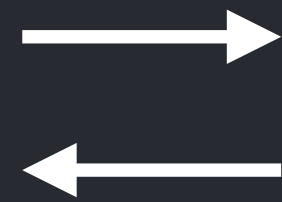
*traditional*



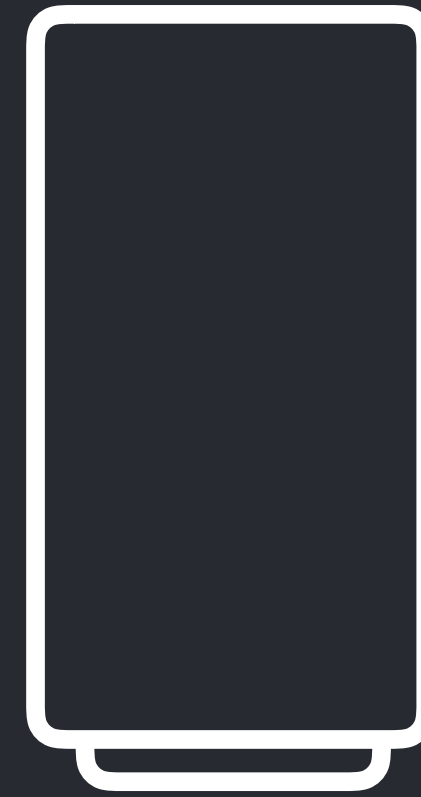
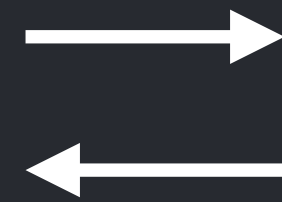
**CDN**



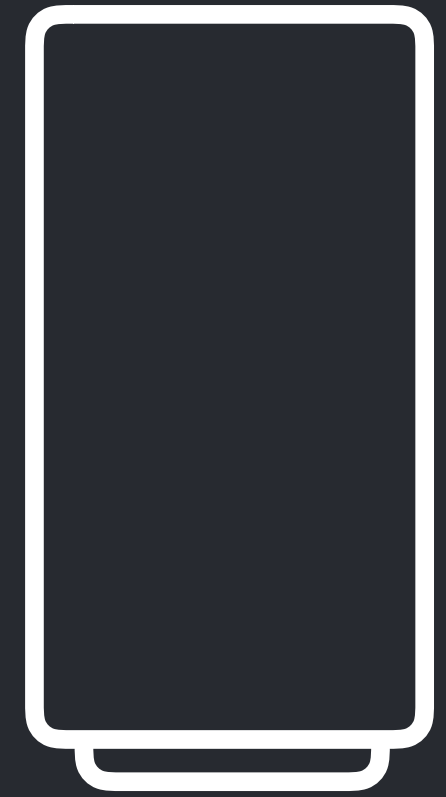
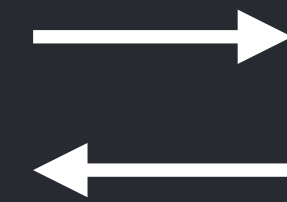
**LOAD BALANCER**



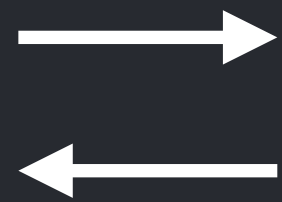
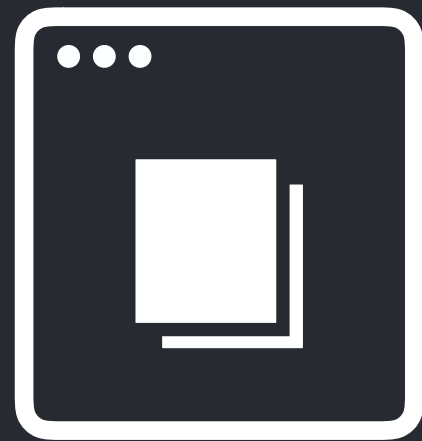
**WEB SERVERS**



**DB SERVERS**



*jamstack*

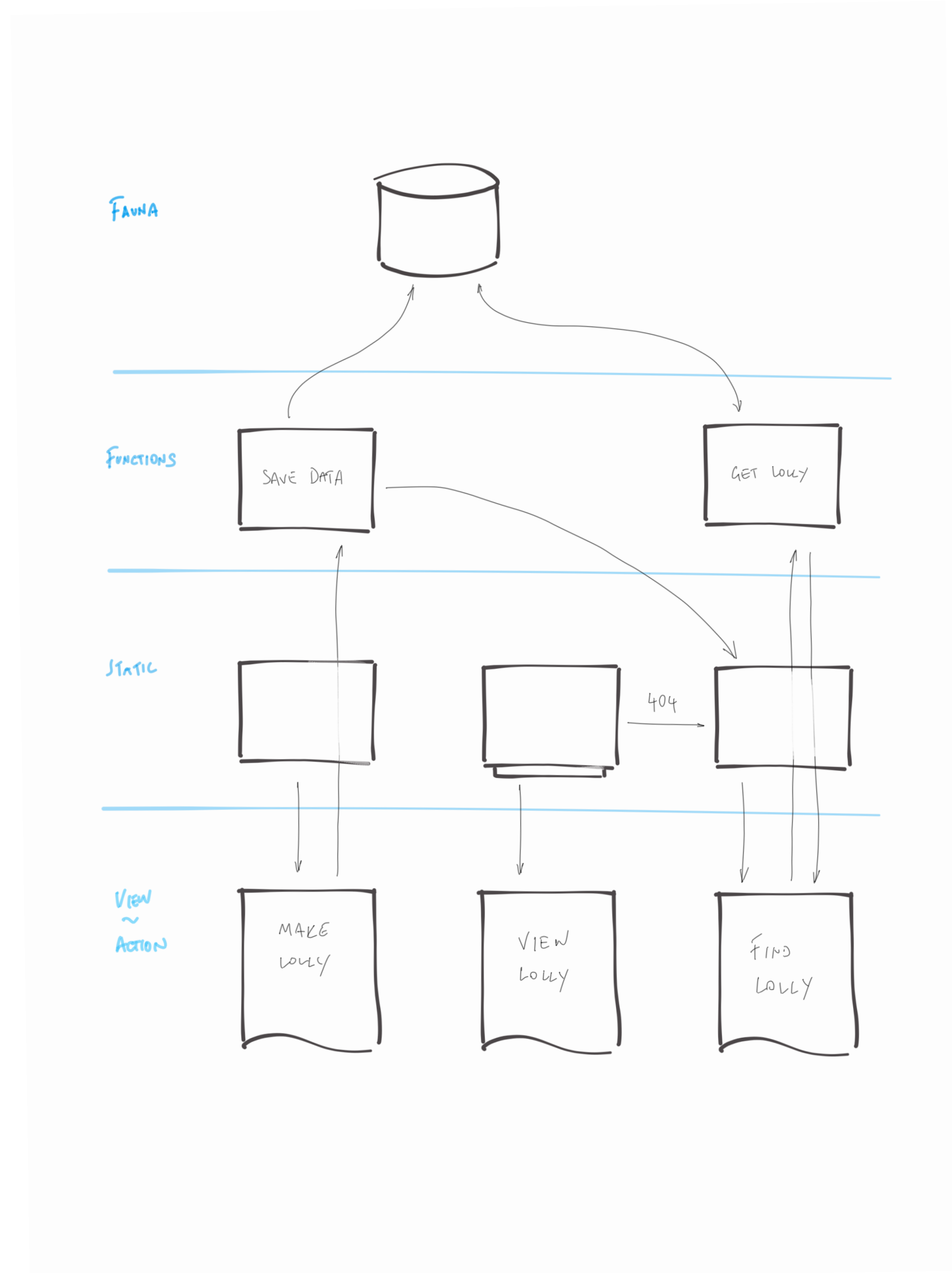


**SIMPLIFYING  
IS NOT DUMBING DOWN**



**SIMPLIFYING  
IS NOT DUMBING DOWN  
IT LETS US FOCUS  
ON WHAT REALLY IS  
IMPORTANT**

# EASIER TO REASON ABOUT



**WHAT SHALL  
WE MAKE?**



Are you being served?

@philhawksworth



# STATIC SITE GENERATORS

[staticgen.com](https://staticgen.com)

# NETLIFY

[www.netlify.com](https://www.netlify.com)

# VARIOUS RELATED RESOURCES FROM CHRIS COYIER

<https://serverless.css-tricks.com/>

# CITRIX PRESENTATION FROM JAMSTACKCONF NYC

[youtube.com/watch?v=kvS5h5domf0](https://youtube.com/watch?v=kvS5h5domf0)

# JAMSTACK CONFERENCE

[jamstackconf.com](https://jamstackconf.com)

# HEADLESS / DECOUPLED CMS

[headlesscms.org](https://headlesscms.org)

# ABOUT THE JAMSTACK / COMMUNITY

[jamstack.org](https://jamstack.org)

# BUILDING A URL SHORTENER WITH NETLIFY REDIRECTS

[findthat.at](https://findthat.at)

## SO I GUESS WE'RE FULL STACK NOW?

[full-stack.netlify.com](https://full-stack.netlify.com)

## MODERN WEB DEVELOPMENT ON THE JAMSTACK

[netlify.com/oreilly-jamstack/](https://netlify.com/oreilly-jamstack/)

## JAMSTACK SLACK

[jamstack.org/slack](https://jamstack.org/slack)

## JAMSTACK COMMENTS EXAMPLE

[jamstack-comments.netlify.com](https://jamstack-comments.netlify.com)

FINDTHAT.AT / SERVERED

THANKS

@PHILHAWKSWORTH