

# Getting to the glass

approaches to rendering views on the web

Phil Hawksworth, Netlify

**Getting to the glass**



**JavaScript dates + Daylight saving**

**January 32th**



Follow ...

Say hello to GPT-4o, our new flagship model which can reason across audio, vision, and text in real time: [openai.com/index/hello-gp...](https://openai.com/index/hello-gp...)

Text and image input rolling out today in API and ChatGPT with voice and video in the coming weeks.



**Getting to the glass**



# Getting to the glass





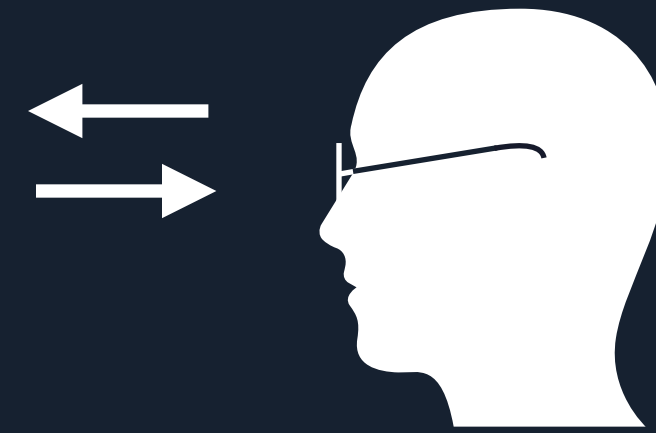
**Getting to the glass**

**“Time to glass”**





**All the magic  
that does the things  
that gets the stuff  
to the place**



# oh, hello

Phil Hawksworth

According to my family, I **work in computers**

A man is dressed as a computer case, with his head and hands protruding from the top and sides. He has a surprised expression, looking upwards and to the right. The computer case is dark grey with various ports, buttons, and a speaker grille. The background is a solid dark grey.

oh, hello

Phil Hawksworth

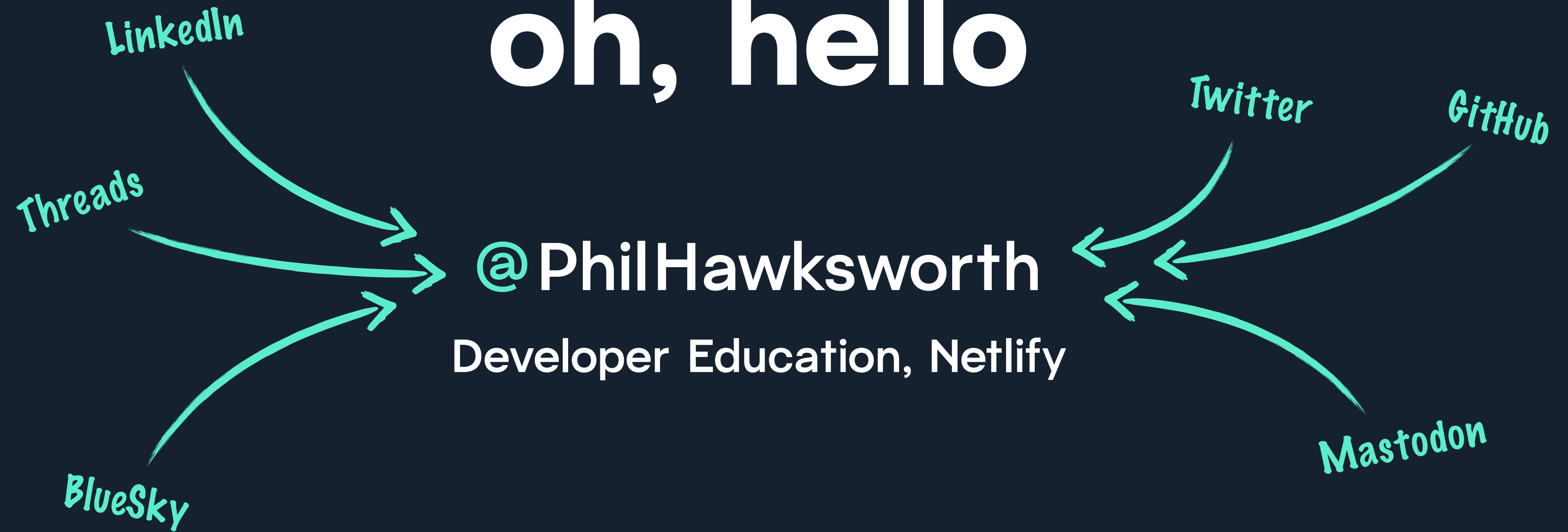
According to my family, I **work in computers**

# oh, hello

Phil Hawksworth

Developer Education, Netlify

# oh, hello





How do we do the best job of  
**getting to the glass?**

**It depends\***

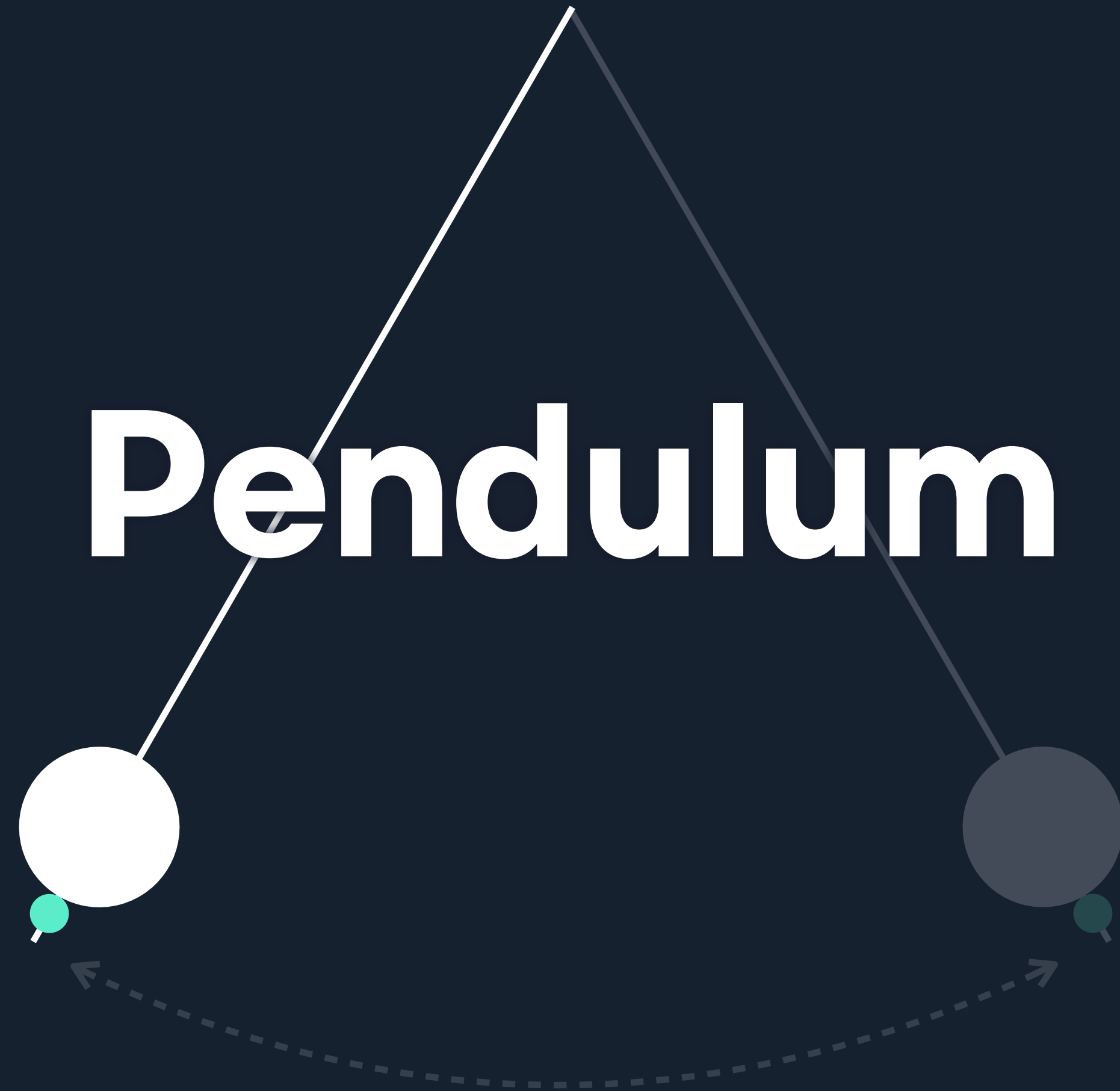
\* more on that later

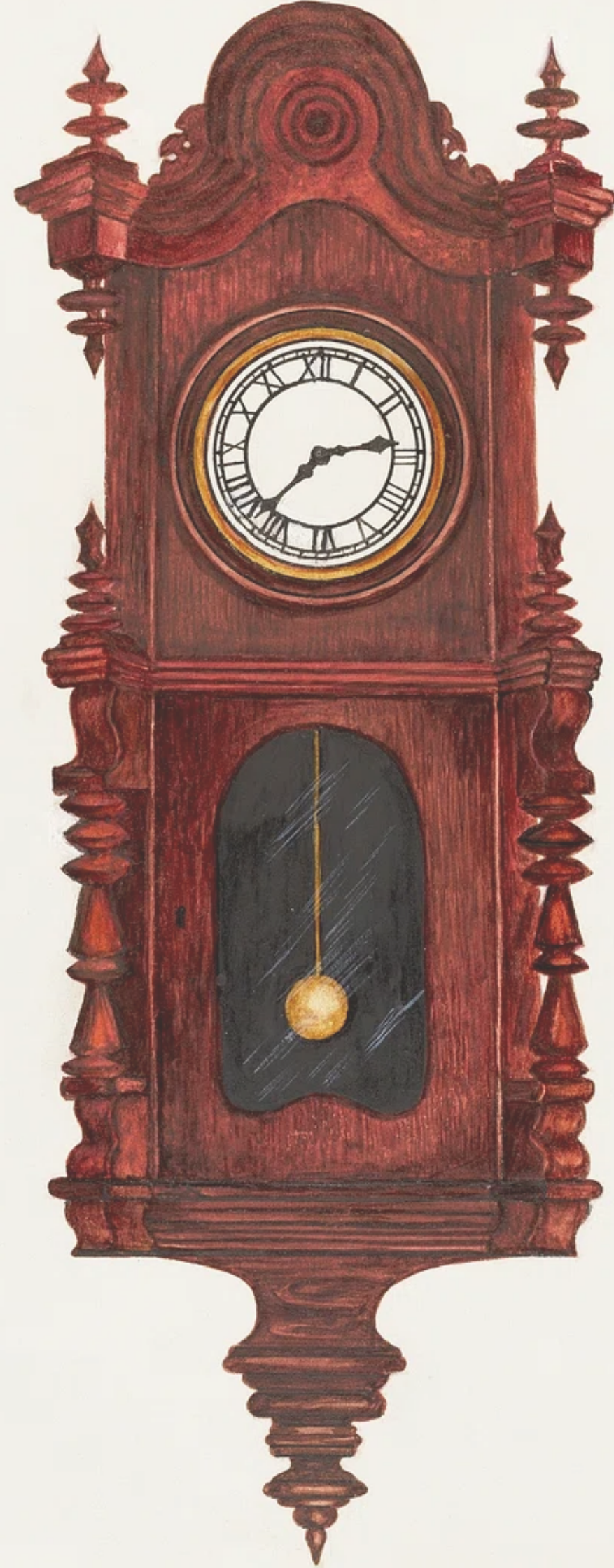




**Trends in  
web development  
are cyclical**

# Pendulum



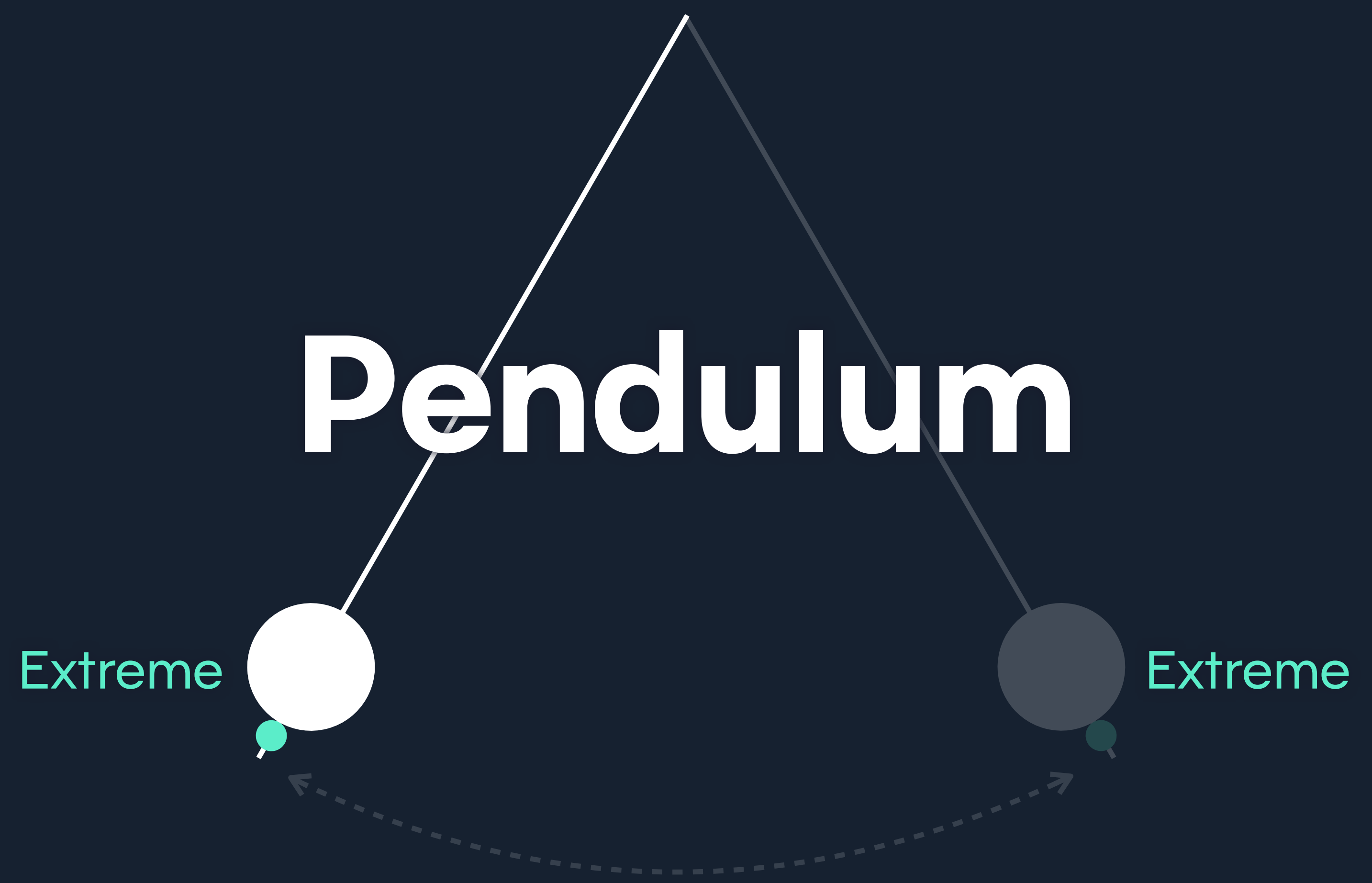


# Pendulum

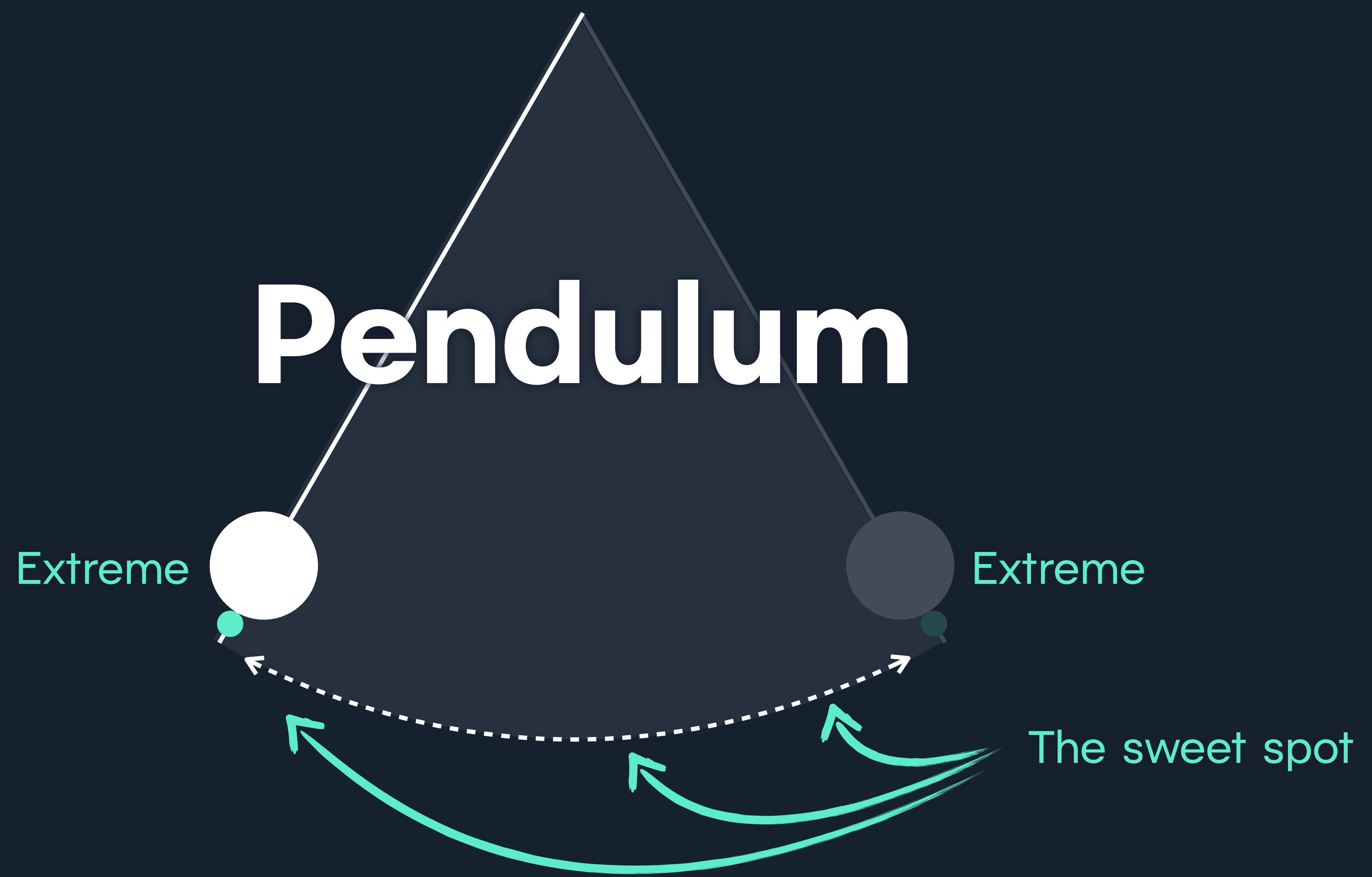
# Pendulum



# Pendulum



# Pendulum



**It depends**

ESR / ISR / ODB / SPA / MPA / ESR

MPA / MPA / SSR / CSR / SSG /

SPA / MPA / SSR / CSR / SSG

SSR / CSR / SSG / DPR / DSG

SSR / CSR / SSG / DPR / DSG /

ESR / ISR / ODB / SPA / MPA /



# It depends

Requirements / Costs / Time / Skills / Sustainability...

**But...**

# Simplicity

is always something to strive for

**The principle of**  
**least power**

Computer Science [...] spent a lot of effort making languages which were as powerful as possible.

Nowadays we have to appreciate the reasons for picking not the most powerful solution but the least powerful

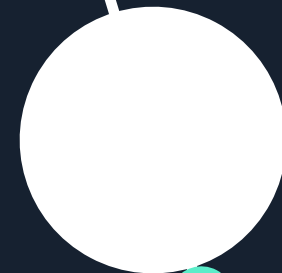
— Tim Berners-Lee

**Simplify as much as possible**

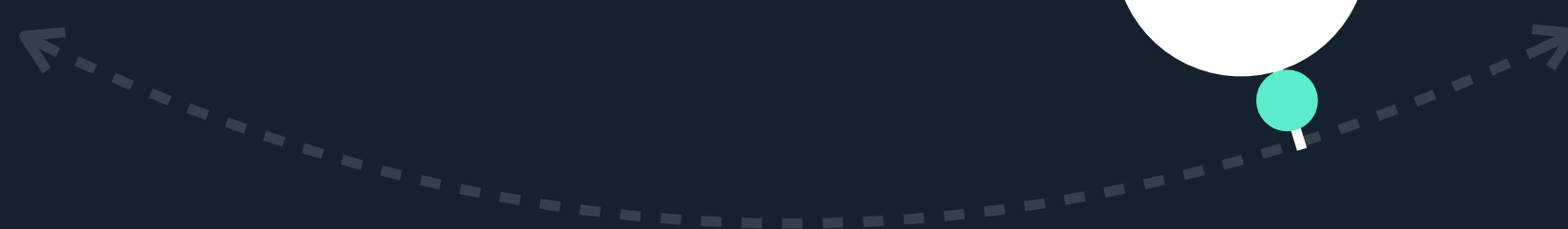
**But no more**

# Pendulum

Complexity



Simplicity



**Pragmatic**





How do we do the best job of  
**getting to the glass?**

ESR / ISR / ODB / SPA / MPA / ESR

MPA / MPA / SSR / CSR / SSG /

SPA / MPA / SSR / CSR / SSG

SSR / CSR / SSG / DPR / DSG

SSR / CSR / SSG / DPR / DSG

ESR / ISR / ODB / SPA / MPA /

# Pendulum

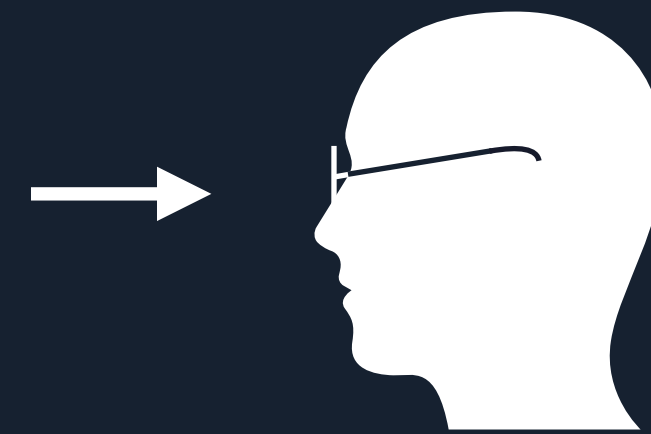
Client render



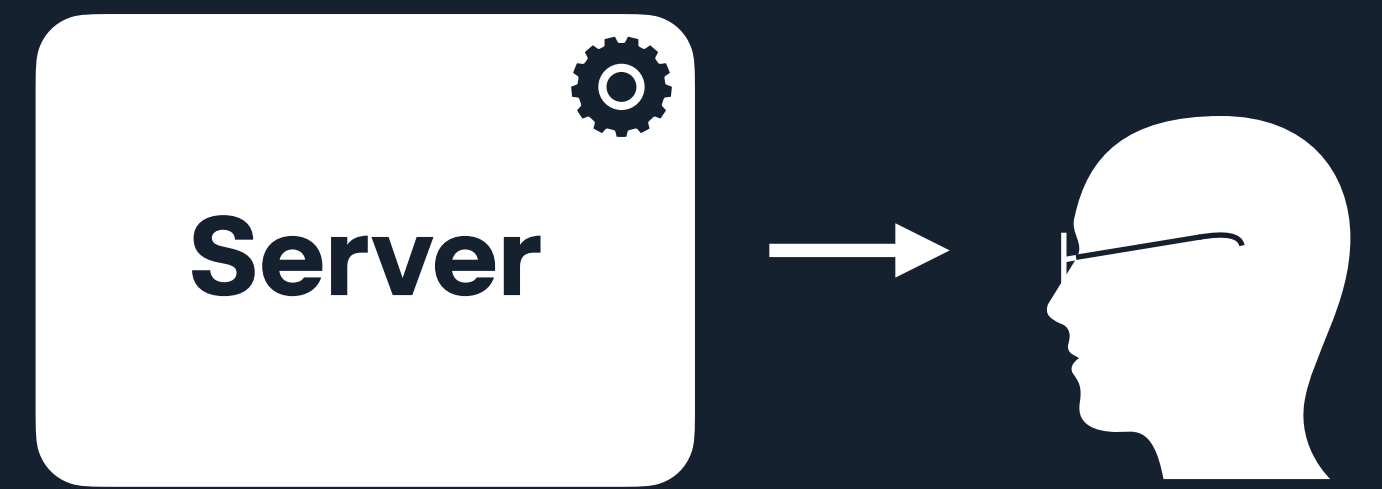
Server render



**All the magic  
that does the things  
that gets the stuff  
to the place**



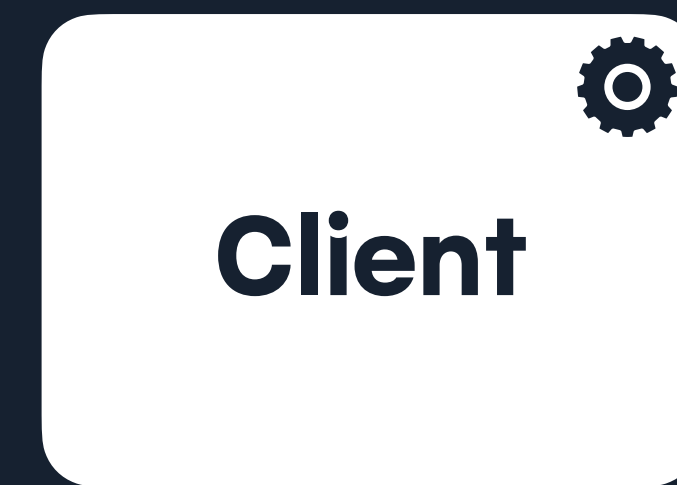
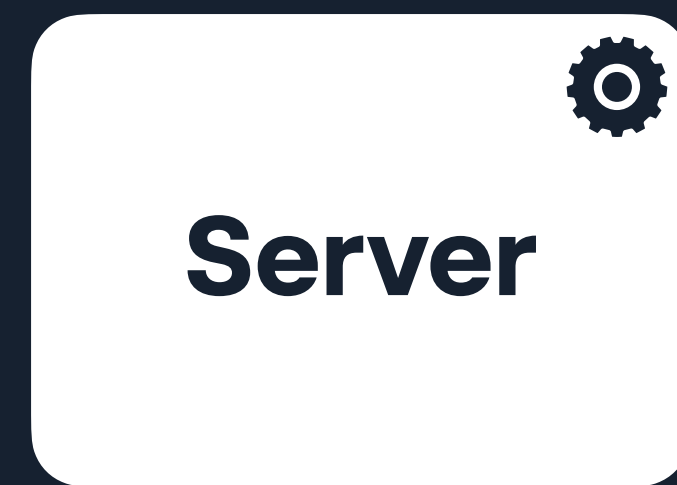
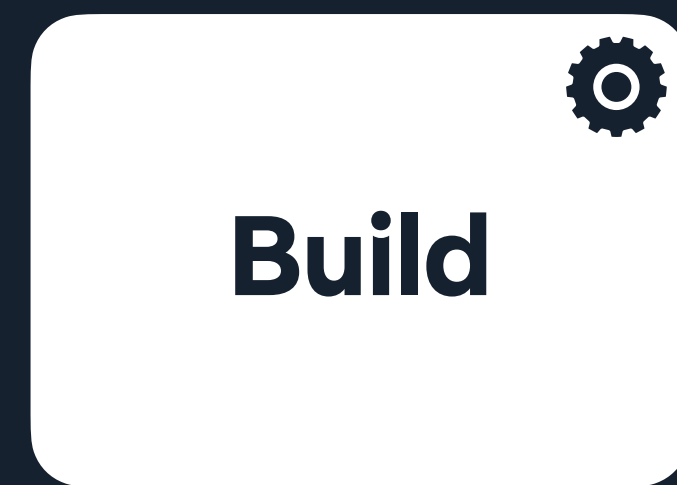






**Server-side**

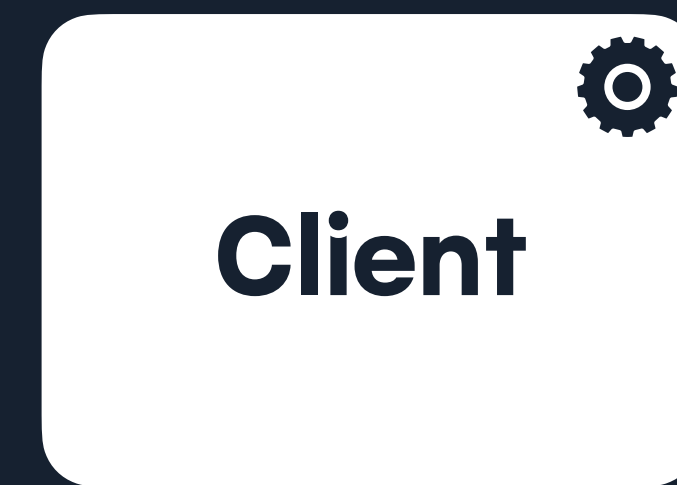
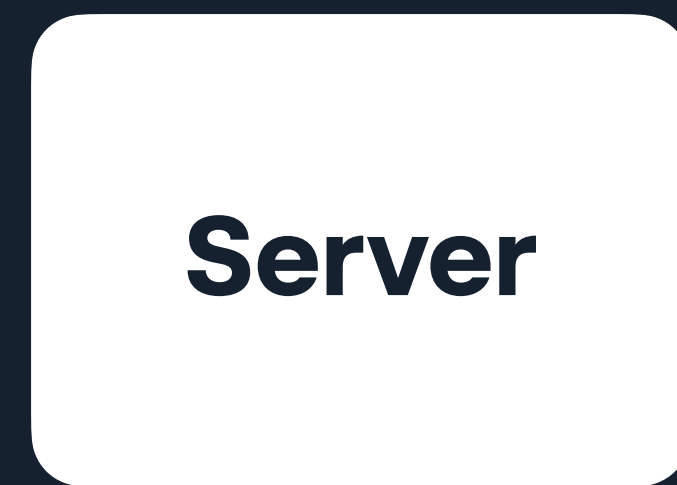
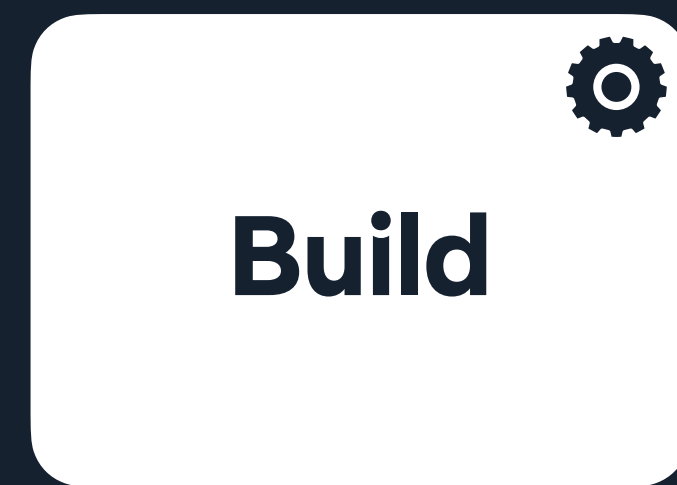
**Client-side**

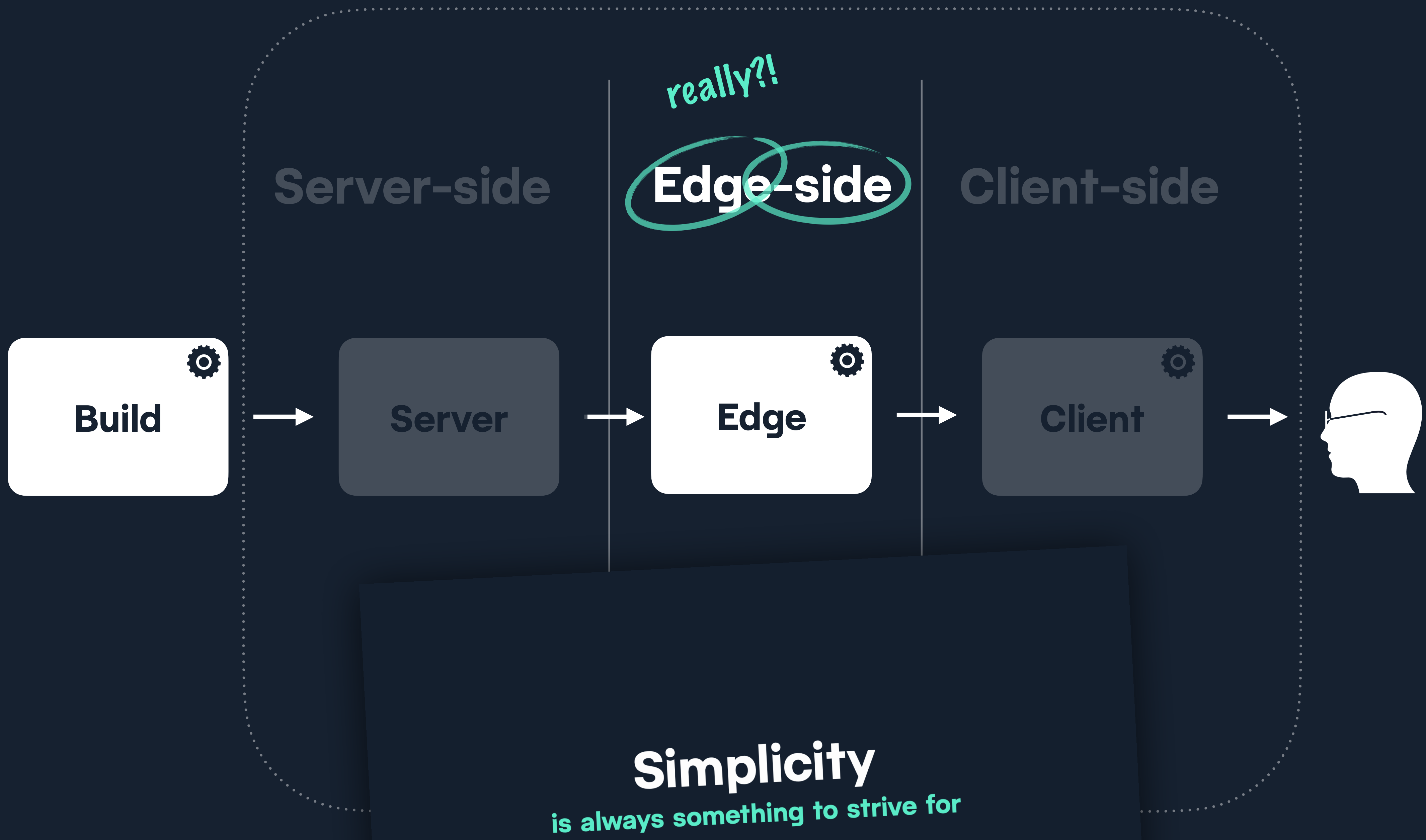




**Server-side**

**Client-side**





**Generate  
our pages**

**BUILD**



**Enhance  
our pages**

**EDGE**

POTENTIALLY



**Display  
our pages**

**CLIENT**

**BORING**

**I'm excited about  
this pattern**

— Shameless display of feelings,  
Phil Hawksworth,  
Just now



Follow ...

Say hello to GPT-4o, our new flagship model which can reason across audio, vision, and text in real time: [openai.com/index/hello-gp...](https://openai.com/index/hello-gp...)

Text and image input rolling out today in API and ChatGPT with voice and video in the coming weeks.







This website utilizes technologies such as cookies to enable essential site functionality, as well as for analytics, personalization, and targeted advertising purposes. You may change your settings at any time or accept the default settings.

You may close this banner to continue with only essential cookies, but we'd prefer you to accidentally press the big Accept button

[Cookie Policy](#)

Accept

[Review](#)





This website utilizes technologies such as cookies to enable essential site functionality, as well as for analytics, personalization, and targeted advertising purposes. You may change your settings at any time or accept the default settings.

You may close this banner to continue with only essential cookies, but we'd prefer you to accidentally press the big Accept button

[Cookie Policy](#)

Accept

[Review](#)



We use cookies because, well, we have to. Click "Accept" if you want to proceed - if not, tough luck. Manage your preferences if you dare by clicking "Cookie Settings".

We gather data through cookies to tailor content, track our traffic, and for some social media stuff. By accepting, you've committed. Read about it in our Privacy Policy if you care. Click the link to enlighten yourself.

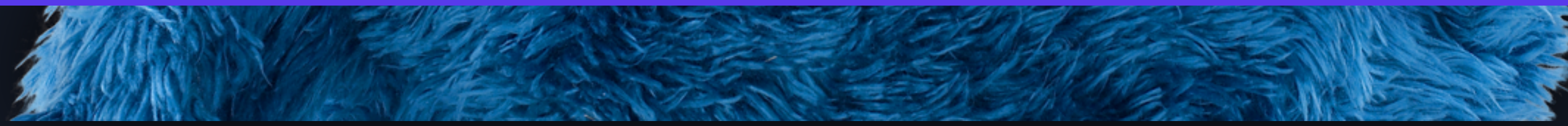
Your privacy matters, but so does our need for cookies. Hit "Accept All" if you have no choice. Choices are limited here. Adjust your settings in our Cookie Settings - or risk the consequences.

Cookies make our site snazzy. Click "Accept All Cookies" or face the consequences. No turning back. Tinker with your cookie settings, if you dare, and uncover our cookie schemes in the Cookie Policy link.am. We are here to assist you and ensure that your browsing experience is both enjoyable and secure.



**Accept**

[Review](#)



**Display  
conditional  
Interface**

**SO**

**Third-party  
client-side  
JavaScript**

**BUT**

**Performance  
killers**

**Control  
cookies**

**Terrible  
interface**

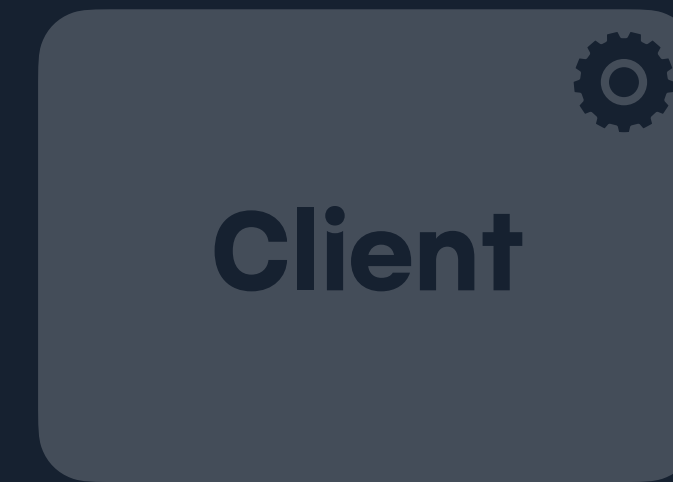
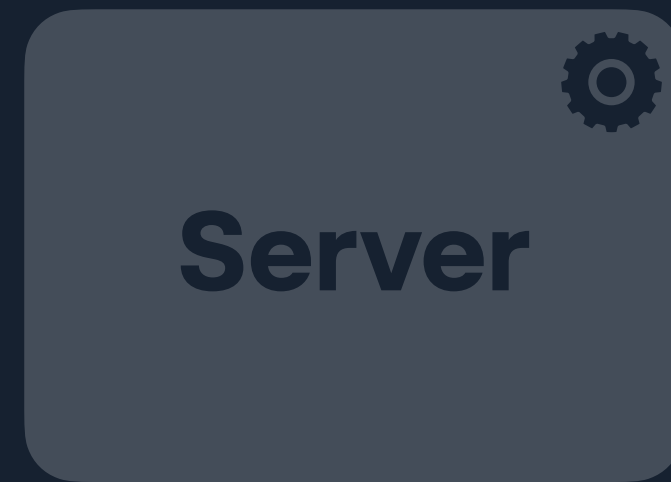




**Server-side**

**Edge-side**

**Client-side**



Search

New

Alex Anderson

Giovanni Benussi

Paul Bratslavsky

Pedro Cattori

Christopher Chedeau

Kent C. Dodds

Clifford Fajardo

Ryan Florence

Arisa Fukuzaki

Edmund Hung

Michael Jackson

Jon Jensen

Cat Johnson

 Remix Contacts

# Remix

This is a demo for Remix.  
Check out [the docs at remix.run](https://remix.run).



This site would like to use cookies (GB)

[Review](#) or [Accept](#)



This site would like to use cookies (GB)

[Review](#) or [Accept](#)



Search

New

Alex Anderson

Giovanni Benussi


Paul Bratslavsky

Pedro Cattori

Christopher Chedeau


Kent C. Dodds

Clifford Fajardo

 Remix Contacts



This is a demo for Remix.  
Check out [the docs at remix.run](https://remix.run).

 This site would like to use cookies (GB) [Review](#) or [Accept](#)

Elements Recorder Console Sources Network Performance Memory Application Security Lighthouse >> ⚙️ ⋮ ✕

```
<link rel="modulepreload" href="/build/root-QQU6HMQA.js">
<link rel="modulepreload" href="/build/routes/_index-QEW43Z5S.js">
▶ <script>...</script>
▶ <script type="module" async>...</script>
<!-- Cookie consent banner injected by Edge Functions due to the requester's location -->
▶ <style>...</style>
... ▶ <div class="nf-cookie-notification">...</div> flex == $0
<!-- end of injected cookie consent banner -->
</body>
</html>
```

html body **div.nf-cookie-notification**

Styles Computed Layout >>

Filter :hov .cls + 🗉 🗑️

```
element.style {
}
```

```
.nf-cookie-notification { (index):38
  font-family: sans-serif;
  font-size: 14px;
  position: fixed;
  bottom: 2em;
  right: 2em;
```

Kent C. Dodds

Clifford Fajardo

 Remix Contacts



This site



Elements

Recorder

Console

Sources

Network

Performance

Memory

Application

```
<link rel="modulepreload" href="/build/root-QQU6HMQA.js">
```

```
<link rel="modulepreload" href="/build/routes/_index-QEW43Z5S.js">
```

```
▶ <script> ... </script>
```

```
▶ <script type="module" async> ... </script>
```

```
<!-- Cookie consent banner injected by Edge Functions due to the requester's location -->
```

```
▶ <style> ... </style>
```

```
... ▶ <div class="nf-cookie-notification"> ... </div> flex == $0
```

```
<!-- end of injected cookie consent banner -->
```

```
</body>
```

```
</html>
```

html

body

div.nf-cookie-notification

**No JavaScript**

**No client-side JavaScript**

**HTML arrives complete**

**Location aware**

**Without needing permission**

**Sandboxed from other site code**  
**Yet under control of site owners**

# An edge function example

netlify/edge-functions/hello.ts

```
export default () => new Response("Hello world");

export const config = {
  path: "/hello"
};
```



# An edge function example

netlify/edge-functions/cookie-consent.ts

```
import { Context } from "@netlify/edge-functions";
import myCookieBannerTemplate from './utils/banner.js';
import cookiePolicyCountries from './utils/territories.js';

export default async (request: Request, context: Context) => {

  const countryCode = context.geo?.country?.code;
```

# An edge function example

```
netlify/edge-functions/cookie-consent.ts
```

```
export default async (request: Request, context: Context) => {  
  
  const countryCode = context.geo?.country?.code;  
  
  // Show the cookie banner to visitors from the correct countries  
  if(cookiePolicyCountries.includes(countryCode)) {  
  
    // Get the HTTP response content  
    const response = await context.next();  
    const page = await response.text();  
  }  
}
```

# An edge function example

```
netlify/edge-functions/cookie-consent.ts
```

```
// Show the cookie banner to visitors from the correct countries  
if(cookiePolicyCountries.includes(countryCode)) {
```

```
    // Get the HTTP response content
```

```
    const response = await context.next();
```

```
    const page = await response.text();
```

```
    // Insert the UI just before the end of the body tag
```

```
    const ui = myCookieBannerTemplate(countryCode);
```

```
    const regex = /<\s*\/body>/i;
```

```
    const updatedPage = page.replace(regex, ui);
```

# An edge function example

```
netlify/edge-functions/cookie-consent.ts

const page = await response.text();

// Insert the UI just before the end of the body tag
const ui = myCookieBannerTemplate(countryCode);
const regex = /<\s/body>/i;
const updatedPage = page.replace(regex, ui);
return new Response(updatedPage, response);
}
}
```

Search

New

Alex Anderson

Giovanni Benussi

Paul Bratslavsky

Pedro Cattori

Christopher Chedeau


Kent C. Dodds

Clifford Fajardo

 Remix Contacts



This is a demo for Remix.  
Check out [the docs at remix.run](https://remix.run).

 This site would like to use cookies (GB) [Review](#) or [Accept](#)

```
Elements Recorder Console Sources Network Performance Memory Application Security Lighthouse >> ⚙️ ⋮ ✕  
<link rel="modulepreload" href="/build/root-QQU6HMQA.js">  
<link rel="modulepreload" href="/build/routes/_index-QEW43Z5S.js">  
▶ <script>...</script>  
▶ <script type="module" async>...</script>  
  <!-- Cookie consent banner injected by Edge Functions due to the requester's location -->  
▶ <style>...</style>  
... ▶ <div class="nf-cookie-notification">...</div> flex == $0  
  <!-- end of injected cookie consent banner -->  
</body>  
</html>  
html body div.nf-cookie-notification
```

```
Styles Computed Layout >>  
Filter :hov .cls + 🗉 🗑  
element.style {  
}  
.nf-cookie-notification { (index):38  
  font-family: sans-serif;  
  font-size: 14px;  
  position: fixed;  
  bottom: 2em;  
  right: 2em;
```

**Generate  
our pages**

**BUILD**



**Enhance  
our pages**

**EDGE**



**Display  
our pages**

**CLIENT**

POTENTIALLY

**Framework free**

**Framework agnostic**



**Platform primitives**

**Generate  
our pages**

**BUILD**



**Enhance  
our pages**

**EDGE**

POTENTIALLY



**Display  
our pages**

**CLIENT**

# Localised times



[Upcoming Events](#) ▾

[Recordings](#) ▾

[Talk Shows](#)

## Schedule

Moar Serverless will feature a full day of sessions. We are currently finalizing our schedule with more session details to be added.

### Thursday May, 23

---

16:00 16:05	<u><a href="#">Opening and Welcome Address</a></u> Sean C Davis
----------------	--

---

16:05 16:45	<u><a href="#">Getting to the Glass – Approaches</a></u> Phil Hawksworth
----------------	---

---

16:55 17:35	<u><a href="#">AI – Have it your way.</a></u> Matt Vella
----------------	---

---

17:45	<u><a href="#">Easter web development: Tackling</a></u>
-------	---

# Localized times



[Upcoming Events](#) ▾

[Recordings](#) ▾

[Talk Shows](#)

## Schedule

Moar Serverless will feature a full day of sessions. We are currently finalizing our schedule with more session details.

### Thursday May, 23

---

16:00 16:05	<u><a href="#">Opening and Welcome Address</a></u> Sean C Davis
----------------	--

---

16:05 16:45	<u><a href="#">Getting to the Glass – Approaches</a></u> Phil Hawksworth
----------------	---

---

16:55 17:35	<u><a href="#">AI - Have it your way.</a></u> Matt Vella
----------------	---

---

17:45	<u><a href="#">Easter web development: Tackling</a></u>
-------	---

FAQ



RAYCAST MERCH

VISIT RAYCAST

CART



Mug Black

£15

ADD TO CART

NOTES:

This mug is an understatement with it's black on black print.

While it won't make your hot beverages more delicious, it will make your colleagues jealous. It's dishwasher safe.



might like



# Pricing data and stock levels

**DONE**

**Generate  
our pages**

**BUILD**



**FAST**

**Enhance  
our pages**

**EDGE**



**DONE**

**Display  
our pages**

**CLIENT**

POTENTIALLY

**But what about  
data that's not  
close to the edge?**



## Netlify Developers

Compose Conference 2024 is coming! [Submit a talk](#)

April 30, 2024

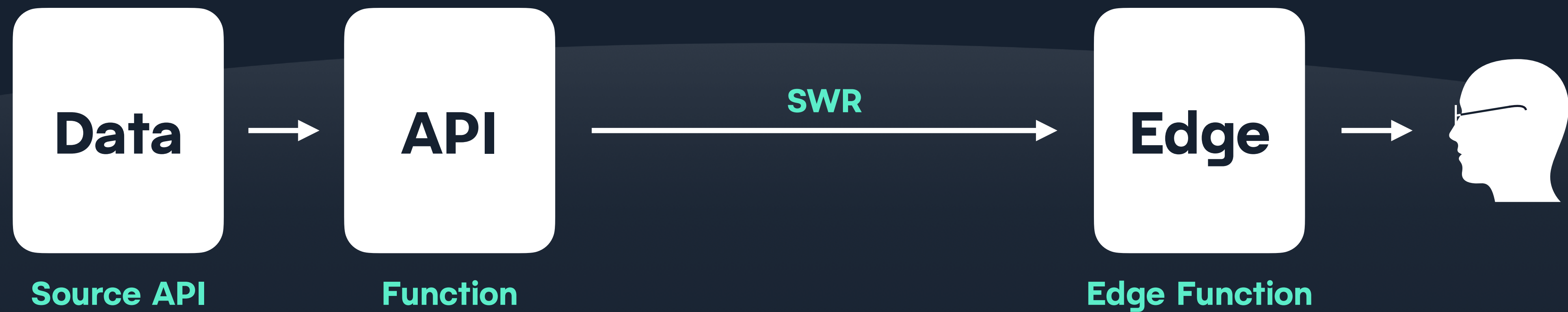
# How to make edge rendering fast

by Mathias Biilmann

After some initial hype around edge rendering, the mood in the front-end ecosystem has muted down somewhat after the surprising discovery that most datasets that people want to render on top of, sits in a database that runs in a single region and that's not going to magically get distributed all over the world. This post will show an approach to fast edge rendering on top of normal centralized APIs by using smart caching and cache invalidation.



# Making edge rendering fast



# Making the edge fast

We're going to do this with one simple addition to our stories function: adding a `netlify-cdn-cache-control` header with a `stale-while-revalidate` instruction.

Change the `netlify/functions/stories.ts` file to the following:

```
export default async () => {
  const resp = await fetch("https://hacker-news.firebaseio.com/v0/topstories.json");
  const ids = await resp.json();

  const stories = await Promise.all(
    ids.slice(0, 100).map(async (id) => {
      const story = await fetch(`https://hacker-news.firebaseio.com/v0/item/${id}.json`);
      return story.json();
    })
  );
};
```

`control` header with a `stale-while-revalidate` instruction.

Change the `netlify/functions/stories.ts` file to the following:

```
export default async () => {
  const resp = await fetch("https://hacker-news.firebaseio.com/v0/topstories.json");
  const ids = await resp.json();

  const stories = await Promise.all(
    ids.slice(0, 100).map(async (id) => {
      const story = await fetch(`https://hacker-news.firebaseio.com/v0/item/${id}.json`);
      return story.json();
    })
  );

  return new Response(JSON.stringify(stories), {
    headers: {
      "content-type": "application/json",
+     "netlify-cdn-cache-control": "public, max-age=0, stale-while-revalidate=86400",
    },
  });
};
```

**A similar emerging pattern**

**PPR**

**Partial Prerendering**

**Generate  
our pages**

**BUILD**



**Enhance  
our pages**

**EDGE**



**Display  
our pages**

**CLIENT**

POTENTIALLY

**Generate  
our pages**

**BUILD / SSR**



**Request  
and stream  
enhancements**

**EDGE**



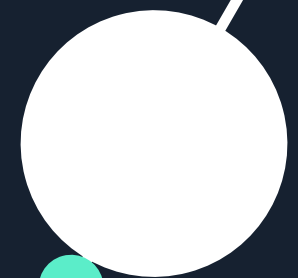
**Update  
our pages**

**CLIENT**

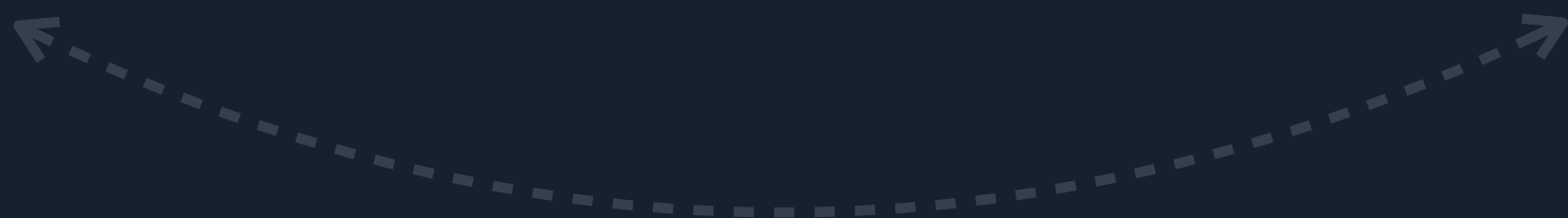
POTENTIALLY

# Pendulum

Monolithic



Decoupled







Developers

[Activity Feed](#)

[Guides](#)

[SDK](#)

[CLI](#)

[Documentation](#)

[Login](#)

[Signup](#)

## Netlify Developers

Compose Conference 2024 is coming! [Submit a talk](#)

May 23, 2024

# Partial Prerendering without a framework

by Matt Biilmann

Partial Prerendering might sound complex, but with Netlify's Platform Primitives it's simple to implement in plain static HTML without any framework or even build system, with the help of Netlify Edge Functions.

# Implementing Partial Prerendering in plain HTML

This example will implement support for Partial Prerendering in a very simple example, with a plain HTML document that has a slot for a dynamically generated time string, and a Netlify Function that returns the current server time. We'll use an Edge Function to tie it all together.

## The HTML

We'll start with a simple HTML page with a placeholder for some dynamic content in a `public/index.html`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PPR</title>
  </head>
  <body>
    <p>Today is: <span data-ppr="/partial/time">Loading...</span></p>
  </body>
</html>
```

Here we use a data tag to indicate that the content of an element should be loaded from a dynamic path

**Did we learn anything?**

**Rendering patterns  
are *yours* to control**

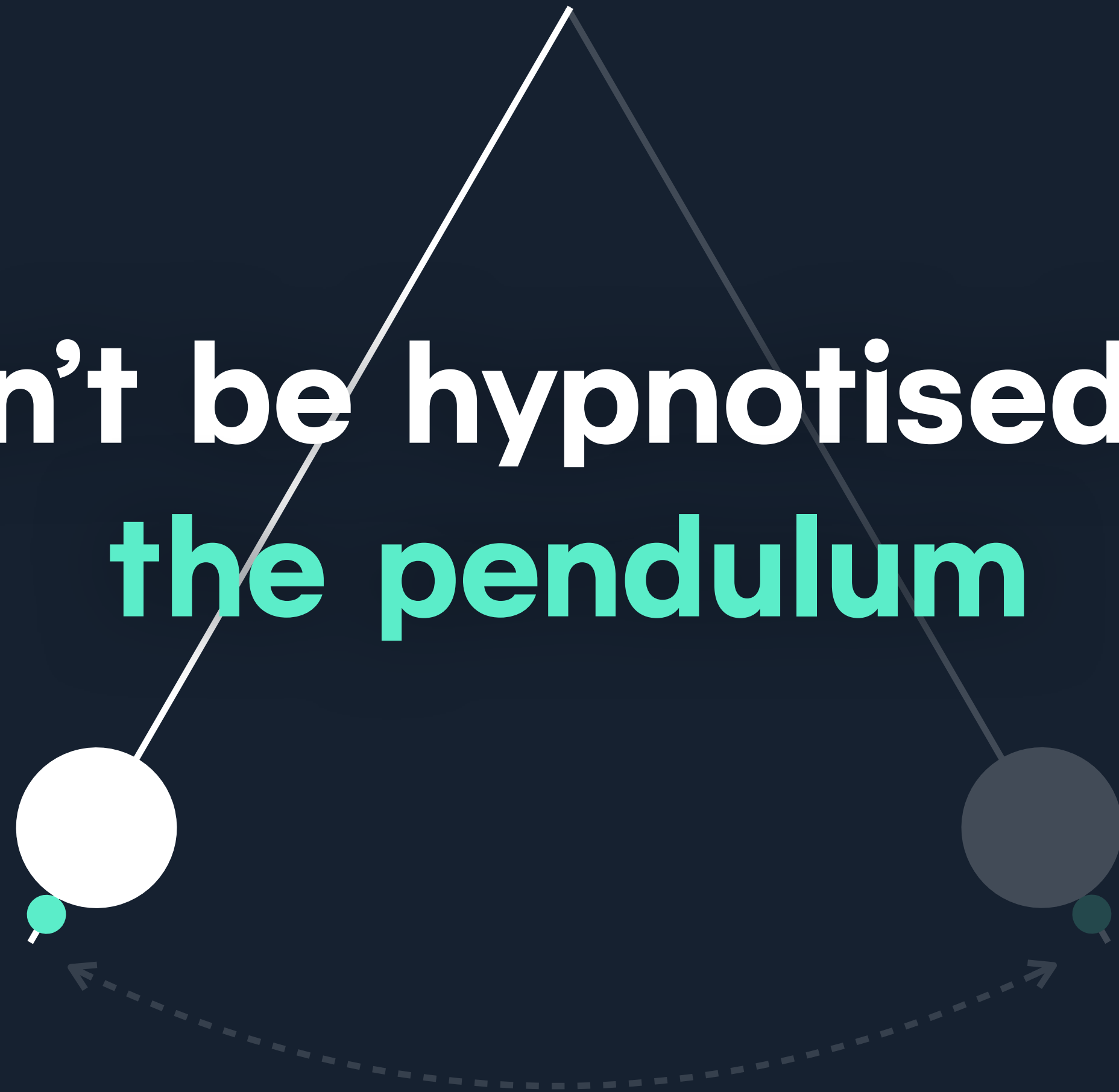
**Understand requirements.**

**Make choices.**

**Start simple.**

**Do as much as possible  
in advance**

**Don't be hypnotised by  
the pendulum**



**It depends**



A blue, furry Cookie Monster is shown from the chest up, sitting in the center. He has a wide, black, open-mouthed grin and two large, white eyes with black pupils and thick black eyebrows slanted downwards, giving him an angry or determined expression. The background is a dark, smoky environment with bright orange and yellow flames rising from the bottom and sides, framing the character. The overall tone is dramatic and intense.

**Should I piss off Cookie Monster?**



No

# Attributions and resources

## Links

<https://twitter.com/philhawksworth>

<https://hawksworx.com>

<https://edge-functions-examples.netlify.app/>

<https://developers.netlify.com/guides/how-to-make-edge-rendering-fast/>

<https://developers.netlify.com/guides/partial-prerendering-without-a-framework/>

## Images

<https://unsplash.com/photos/clear-drinking-glass-with-orange-liquid-4aWCVJA2uSo>

<https://unsplash.com/photos/a-tall-glass-sitting-on-top-of-a-table-fiN073Jt5L8>

[https://unsplash.com/photos/turned-on-flat-screen-monitor-6j\\_ImeycTrM](https://unsplash.com/photos/turned-on-flat-screen-monitor-6j_ImeycTrM)

[https://unsplash.com/photos/brown-and-black-floral-wreath-6BMjuiB\\_IF0](https://unsplash.com/photos/brown-and-black-floral-wreath-6BMjuiB_IF0)

<https://www.flickr.com/photos/tangledcontrolpads/639854414/in/photostream/>

<https://unsplash.com/photos/bonfire--98jVaVuGv0>

# Thanks

[@philhawksworth](#)