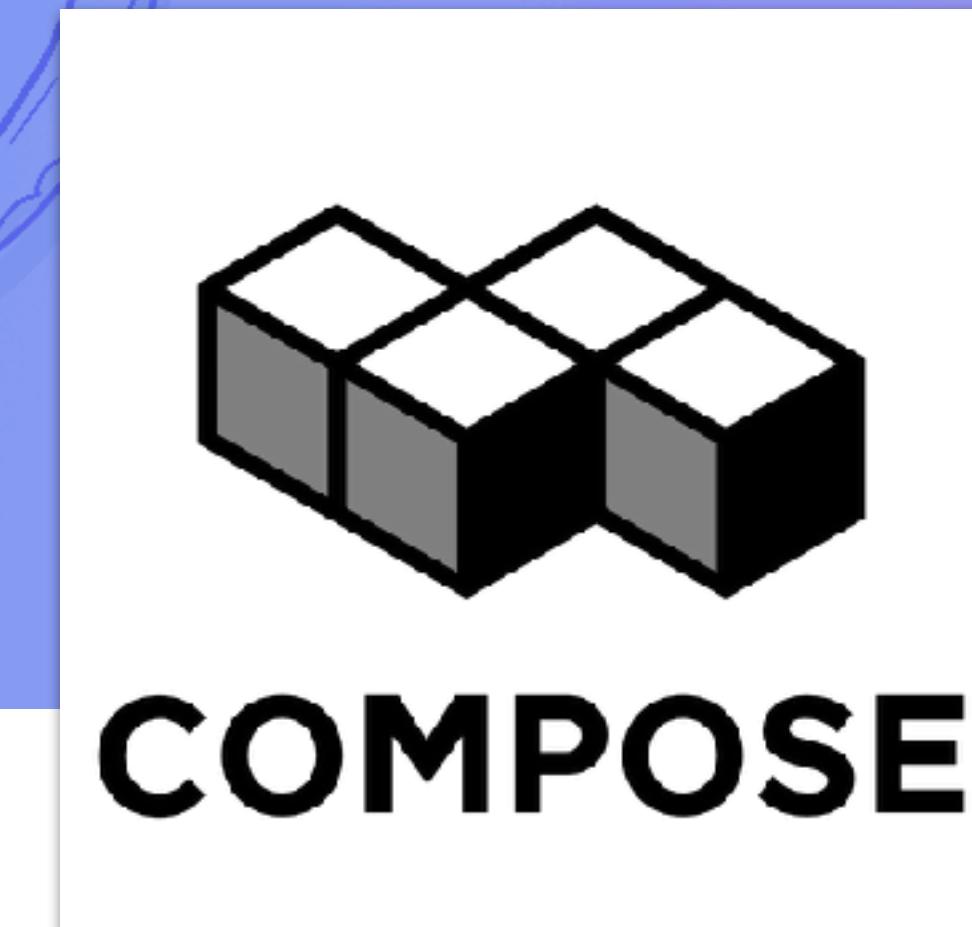


Down the Stack

BUILDING A JAVASCRIPT RUNTIME



Who the heck are you?



I'm Kurt! I helped build Ars Technica and learned painful performance lessons. Then I built a company called Compose. Now I'm working on fly.io to ... make apps fast.

Glossary

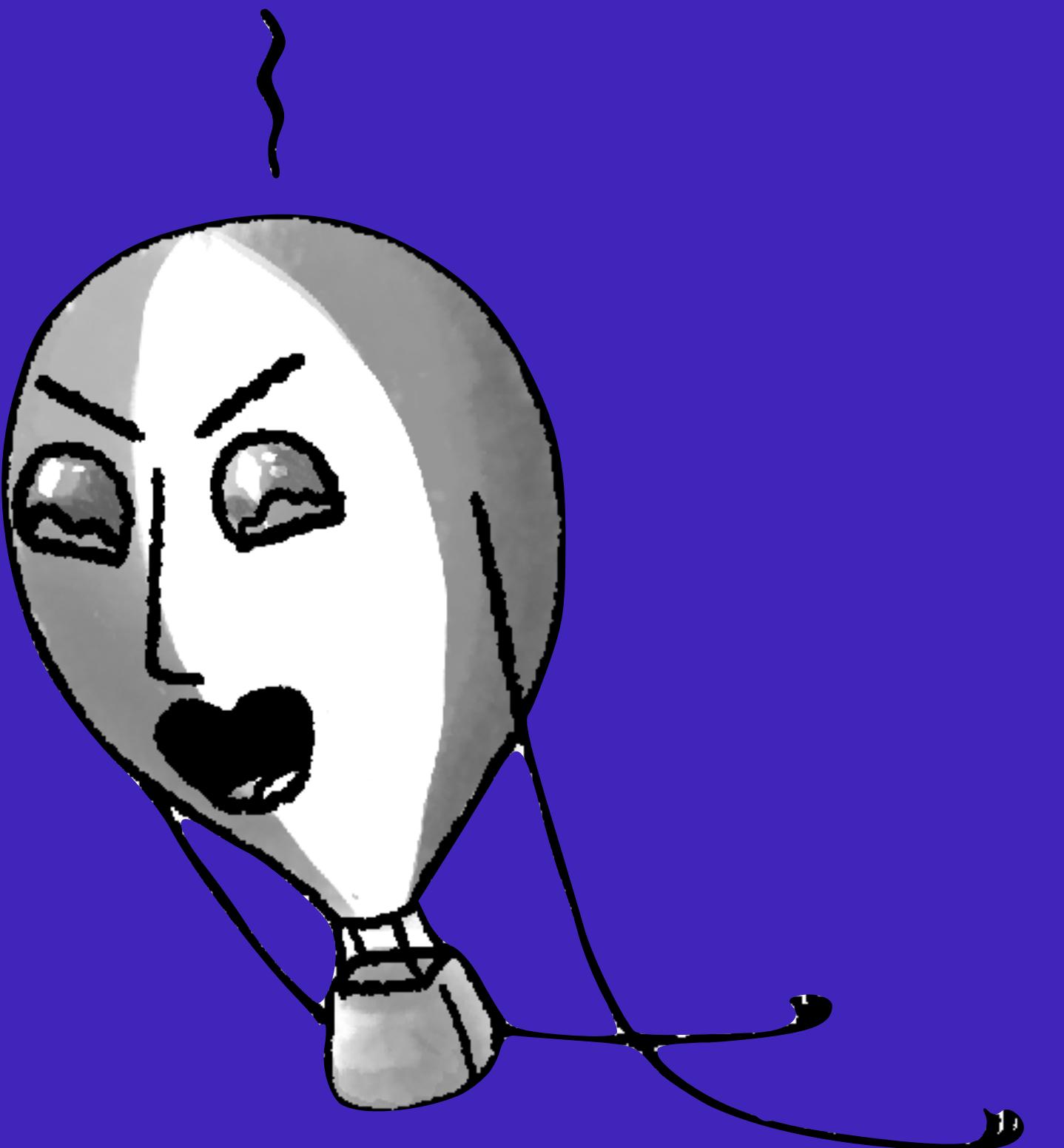
JavaScript Engine: executes JavaScript, manages heap, no “side effects”

JavaScript Runtime: engine + APIs + execution control

We built a Global
JavaScript runtime.



Why the fudge would
you do something like
that?



First, Global Apps ...

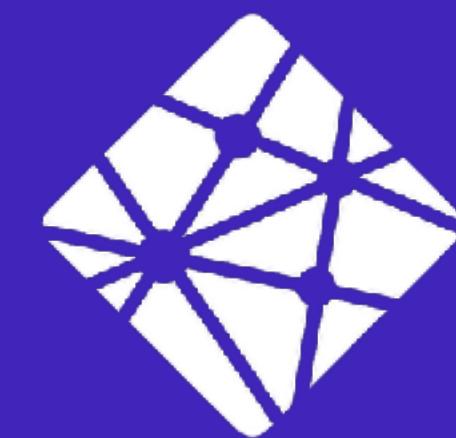
Modern apps serve users all over the world. Logic and data should be very close to the users who need it.

This is not new, you use a lot of Global Apps.

They're just insanely hard to build.



Cloudinary



netlify

tinderTM

You lose out when you
deploy to us-east-1



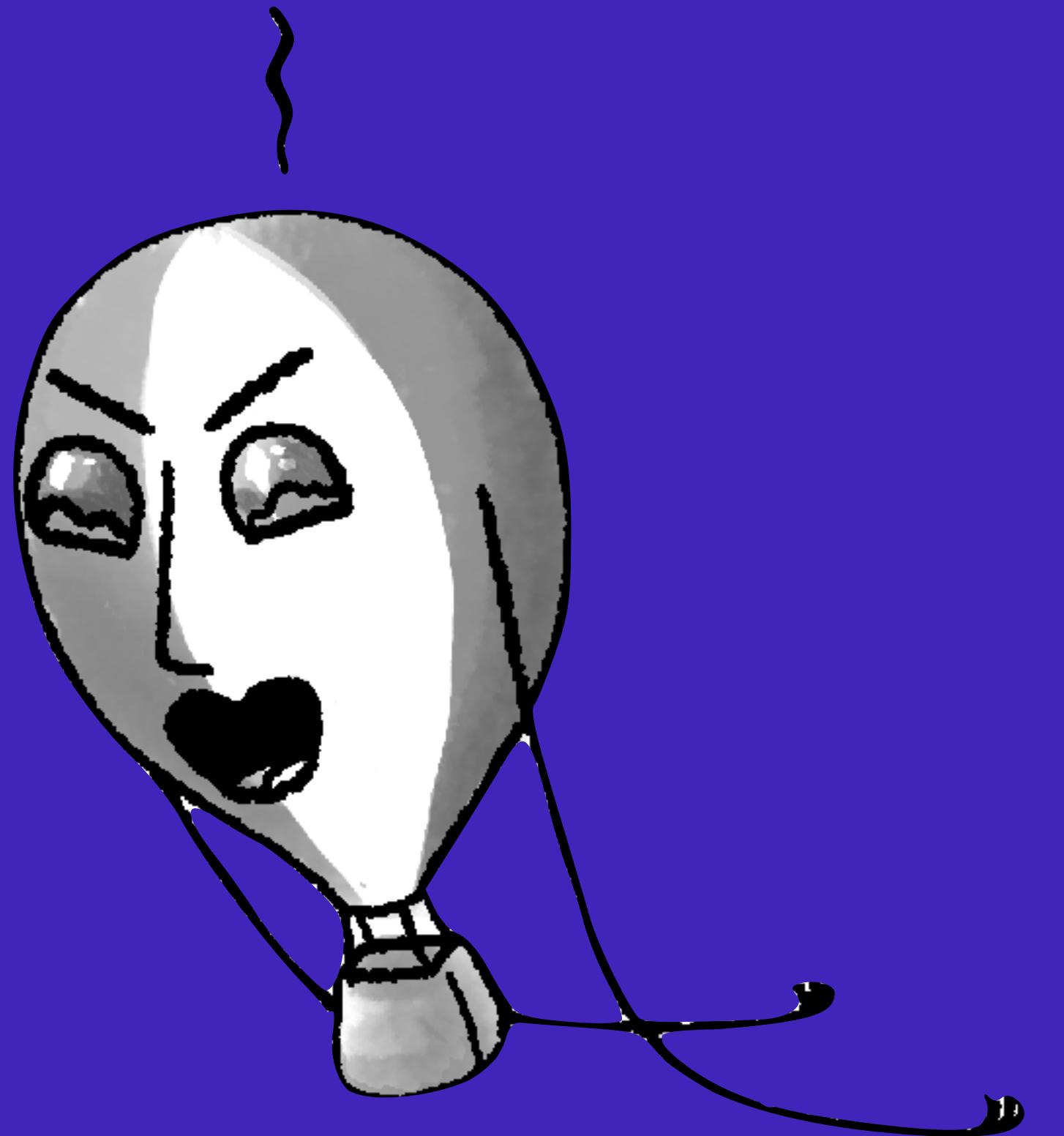
A modern React app

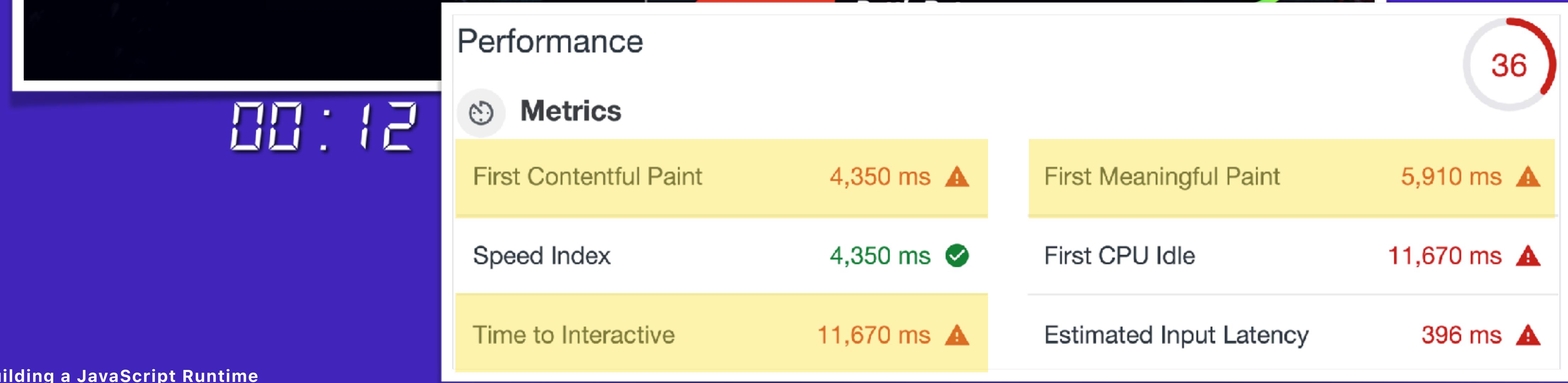


The onion of compromise.

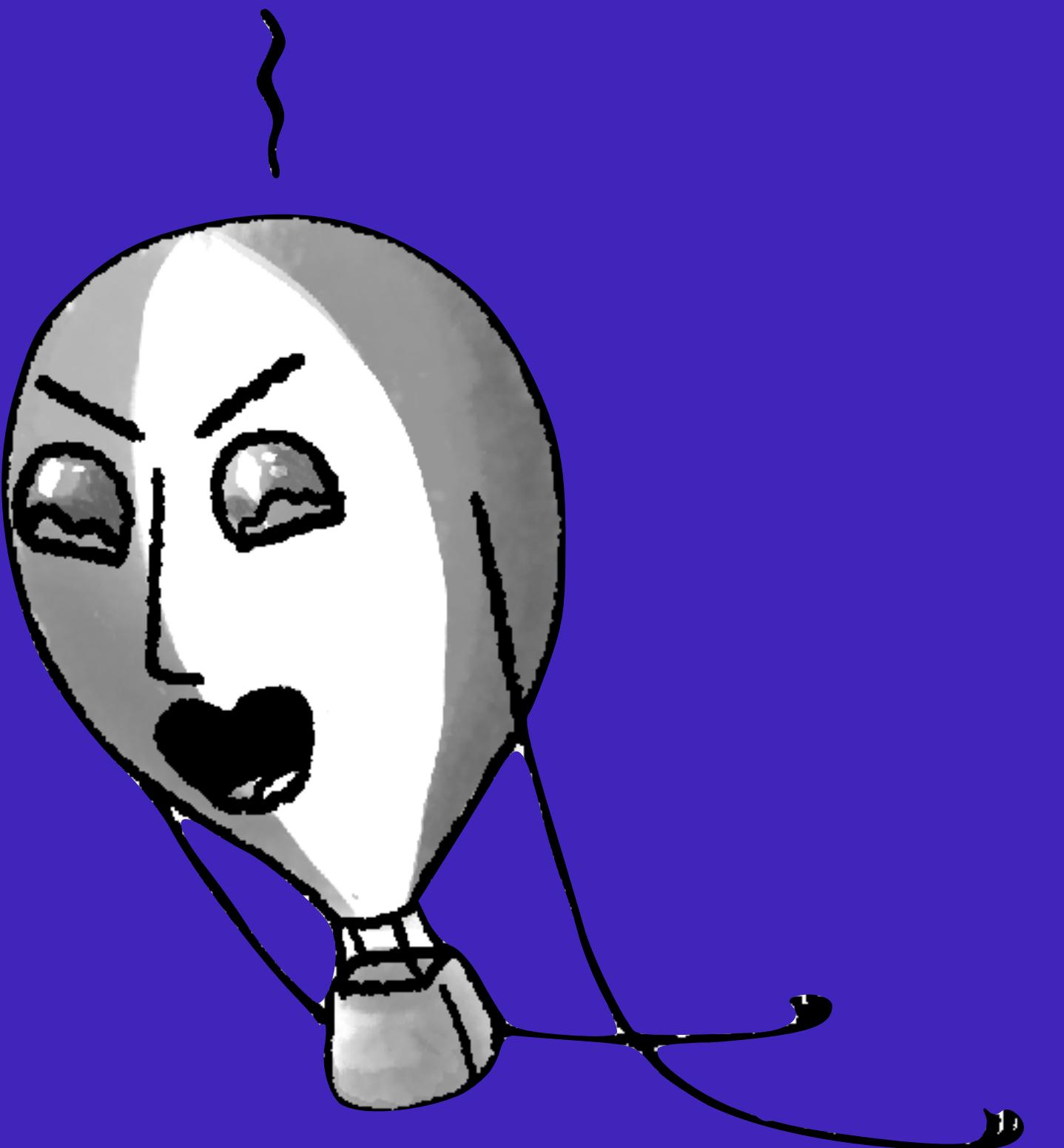
1. Your **application** logic
2. Someone else's **CDN**
3. An **intern's** third party JavaScript

Third  party 
JavaScript. 

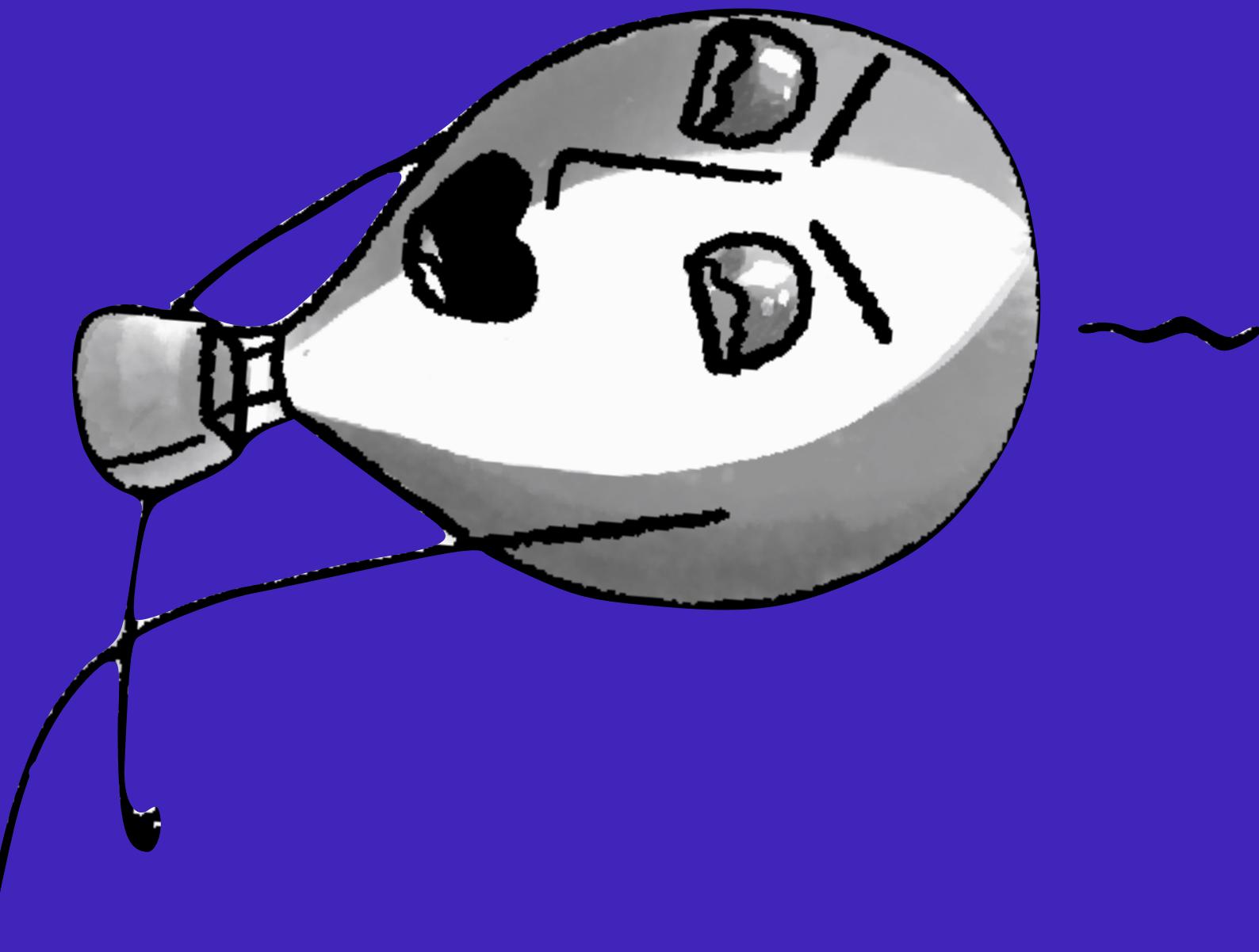




CDNs are all kinds of
limited



Does anyone even
understand the Vary
header?



The "Vary" header field in a response describes what parts of a request message, aside from the method, Host header field, and request target, might influence the origin server's process for selecting and representing this response. The value consists of either a single asterisk ("*") or a list of header field names (case-insensitive).

```
Vary = "*" / 1
```

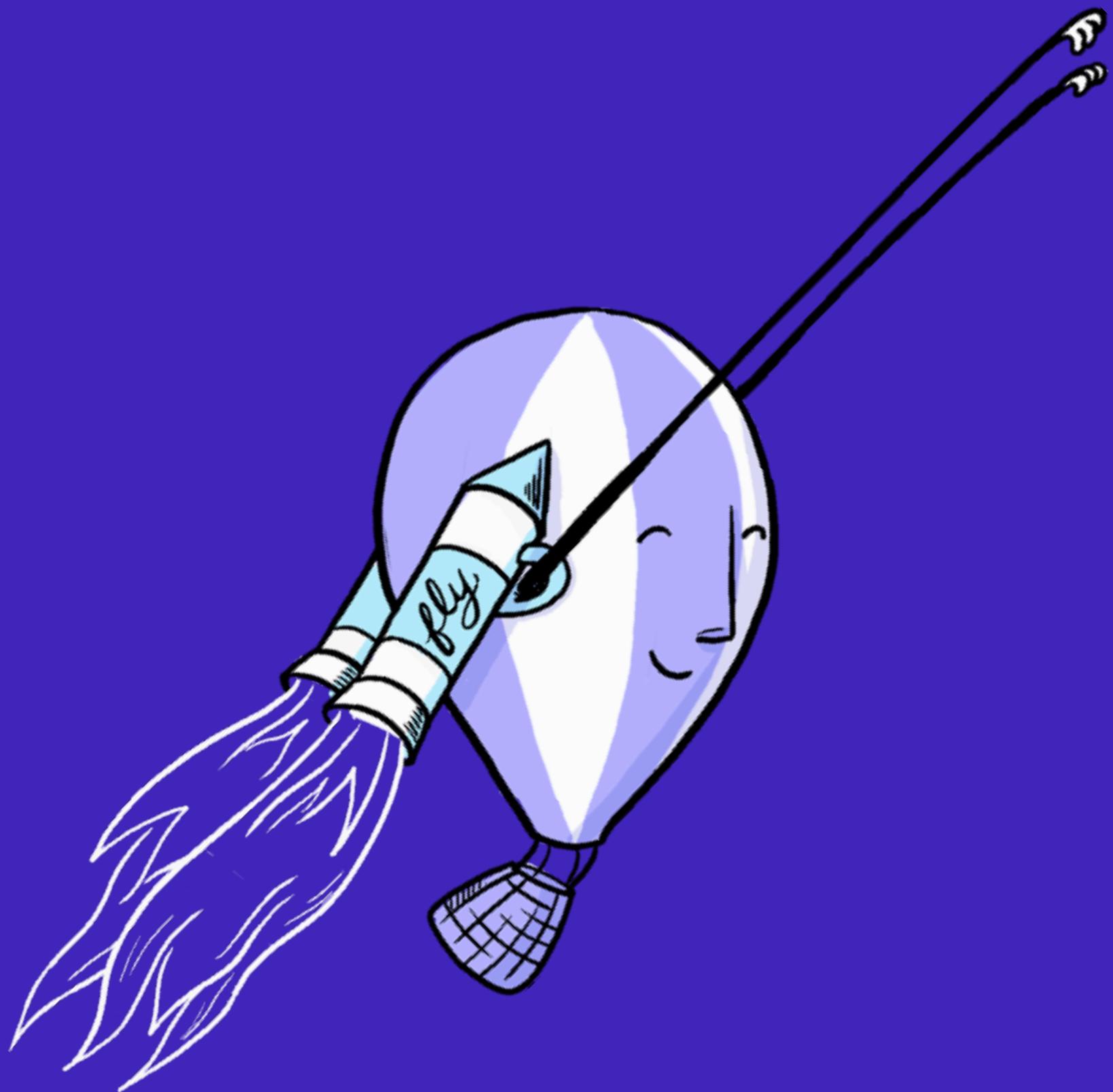
A Vary field value of "*" signals that anything about the request might play a role in selecting the response representation, possibly including elements outside the message syntax (e.g., the client's network address). A recipient will not be able to determine whether this response is appropriate for a later request without forwarding the request to the origin server. A proxy MUST NOT generate a Vary field with a "*" value.

A Vary field value consisting of a comma-separated list of names indicates that the named request header fields, known as the selecting header fields, might have a role in selecting the representation. The potential selecting header fields are not limited to those defined by this specification.

What we needed

1. Lightweight isolation, fast boot
2. APIs distributed data + logic
3. Something familiar

JavaScript! Web APIs!



JavaScript Engines

V8 (Google): “easy” to use from Node, C++, doable with Rust

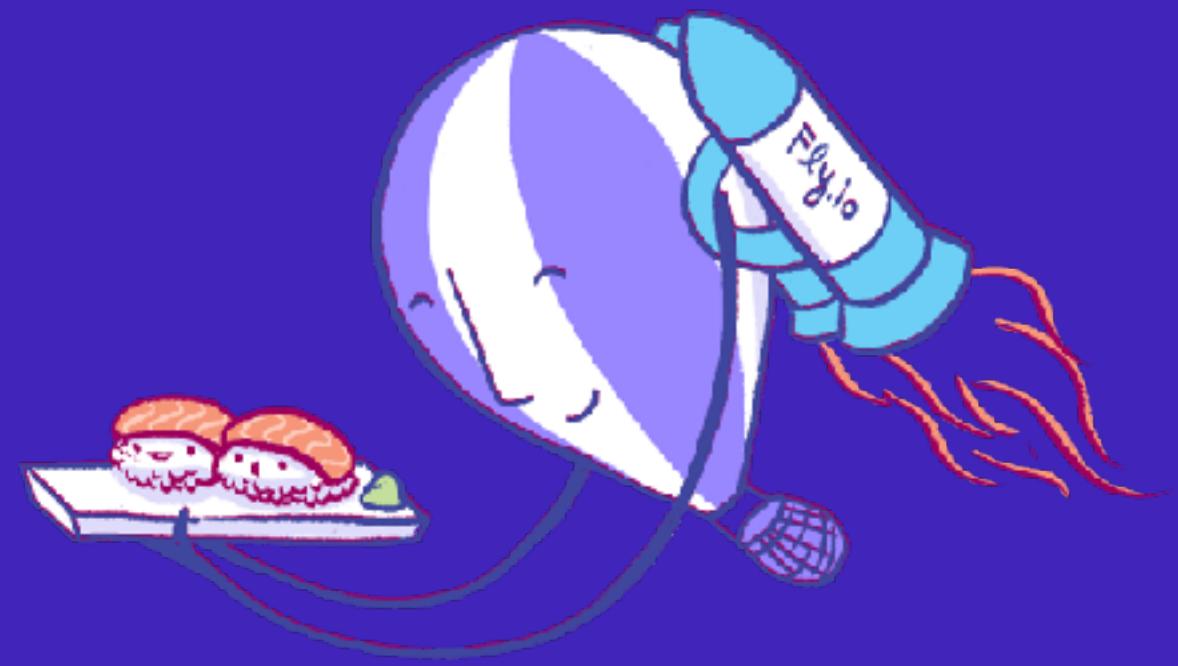
SpiderMonkey (Mozilla): start with Servo: github.com/servo/rust-mozjs

Chakra (Microsoft): .NET, Rust, C++, great docs.

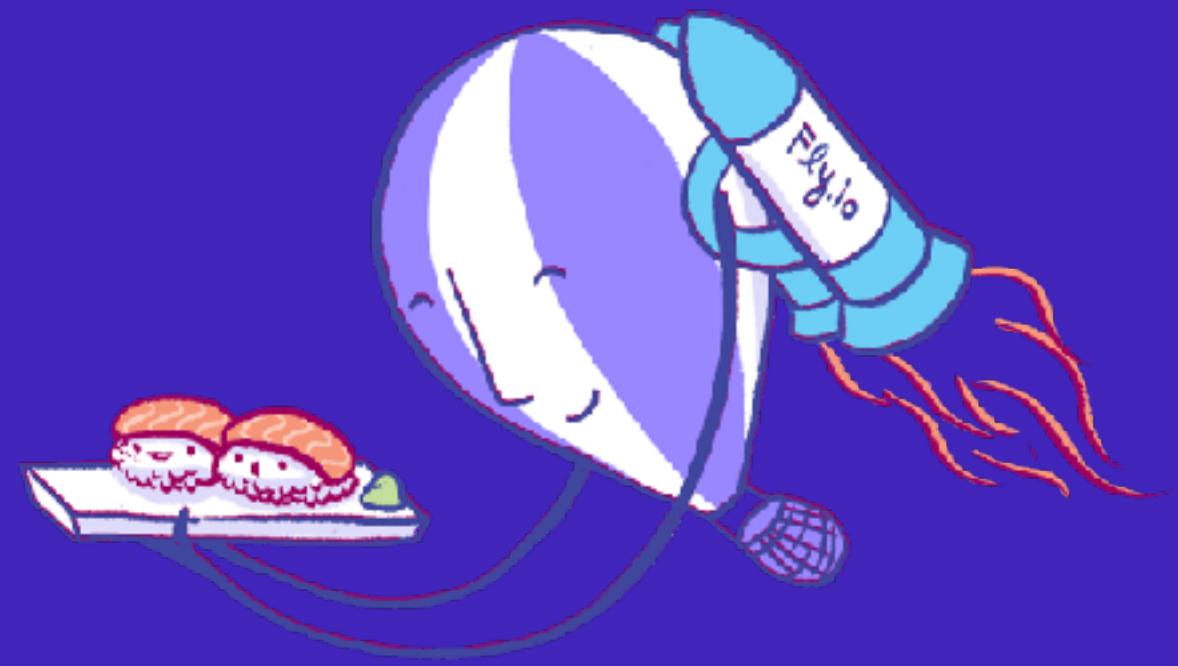
JSCore: Swift or Obj-C



Concept: Isolates (VMs)
manage resources, “isolate”
logic. And collect garbage.



Concept: Contexts are global scopes for code + heap. Isolates can have multiple contexts.



What's built in

1. **Compilation**
2. **Types:** primitives, arrays, objects, buffers, etc.
3. **Promises:** but only barely



Batteries Not Included

```
// browsers implement a `fetch` method
// for making HTTP requests
fetch("https://fly.io").then(function (r) {
  console.log("got response:", {
    status: r.status,
    size: r.body.length
  })
})
```

→ **embedding-v8** ./simple.ts examples/00-fetch.js
ReferenceError: fetch is not defined
at <isolated-vm>:3:1

No console for you

```
console.log("hello world")
```

- embedding-v8 ./simple.ts examples/01-hello.js
- embedding-v8 |

Libraries, schmibraries

```
let lib = require('anything')
```

→ **embedding-v8** ./simple.ts examples/06-require-lol.js
ReferenceError: require is not defined
at <isolated-vm>:1:11

Import isn't even a keyword

```
let lib = require('anything')
```

```
→ embedding-v8 ./simple.ts examples/05-import.js
```

```
/Users/kurt/code/fly.io/talks/embedding-v8/simple.ts:16
isolate.compileScriptSync(src).runSync(context)
^
```

```
SyntaxError: Unexpected token import [<isolated-vm>:1:1]
at Object.<anonymous> (/Users/kurt/code/fly.io/talks/embedding-v8/simple.ts:16:9)
```

Batteries Not Included

- 1. You control code execution**
- 2. Node.js: v8 + event loop + i/o**
- 3. What even is concurrency?**



JavaScript engines rely
on the “host” code for
almost everything.

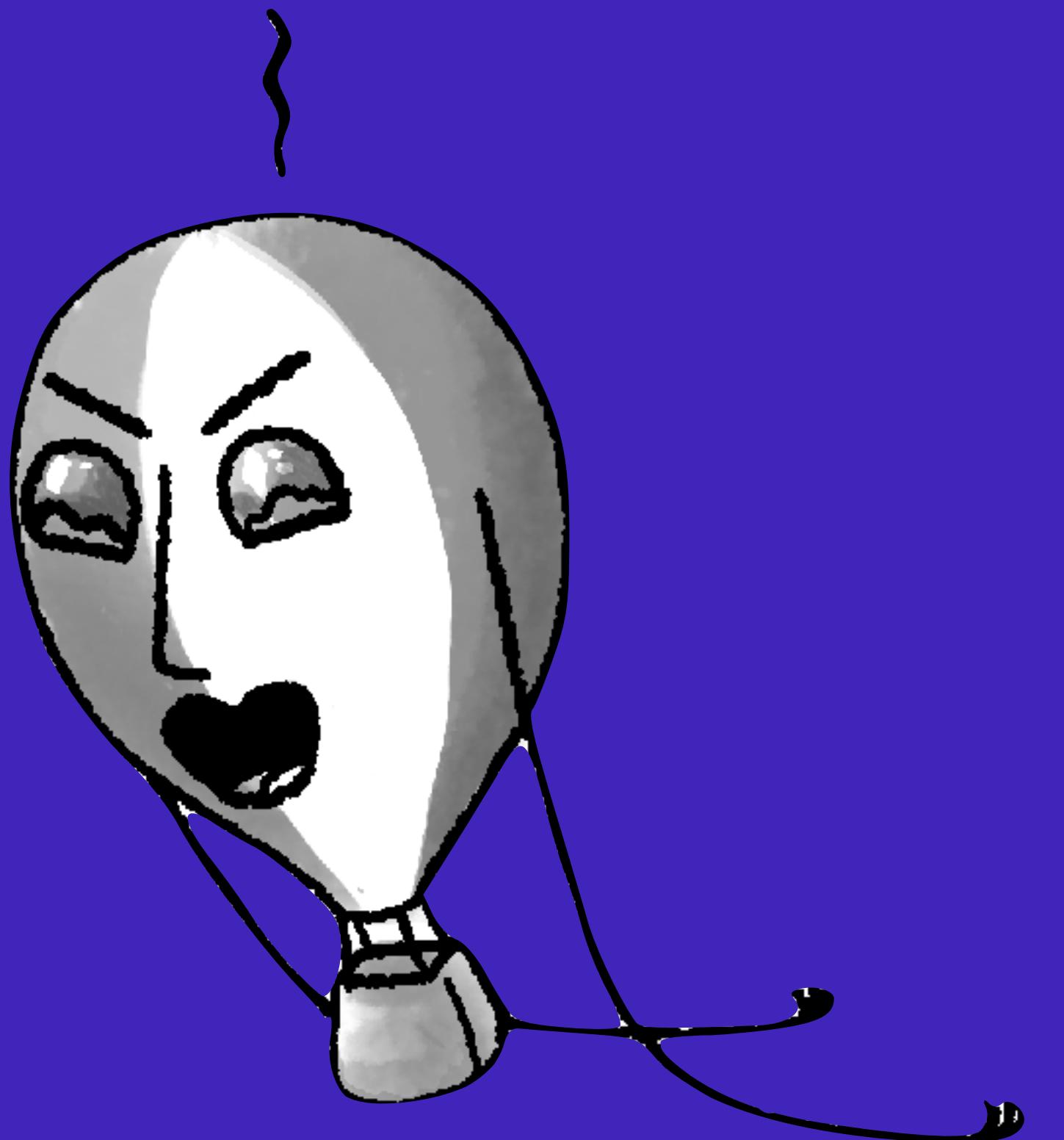


Getting started

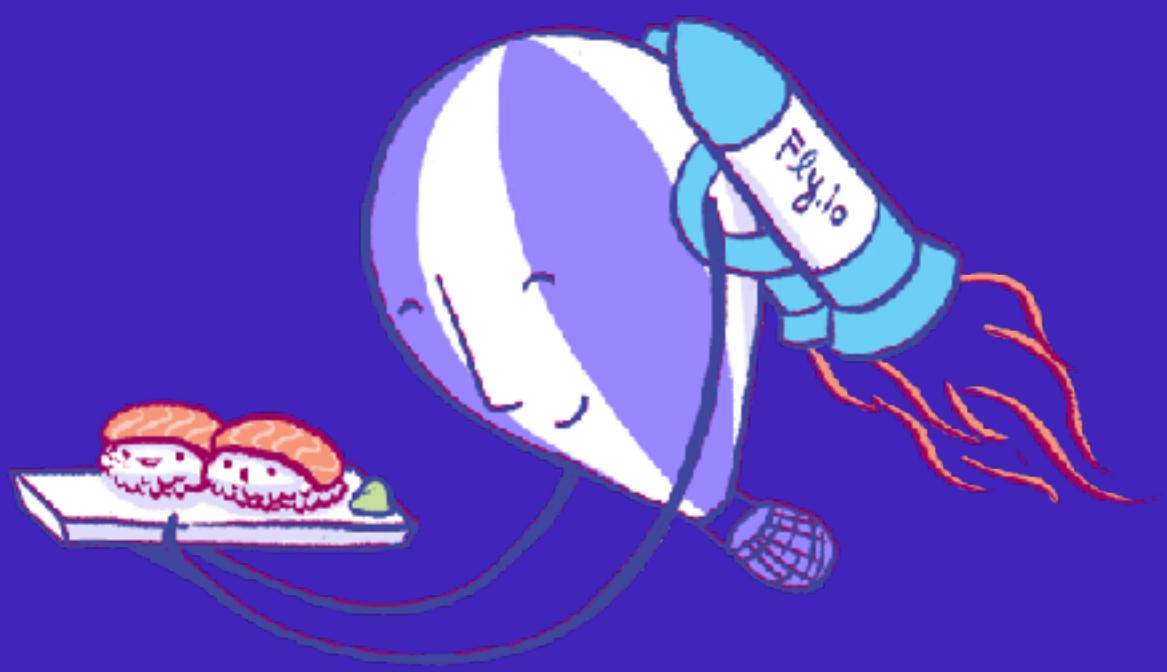
- 1. Install the isolated-vm package in a Node app.**
- 2. Start implementing APIs. Browser polyfills are a good first step.**
- 3. Build bridge for APIs that need to get out.**



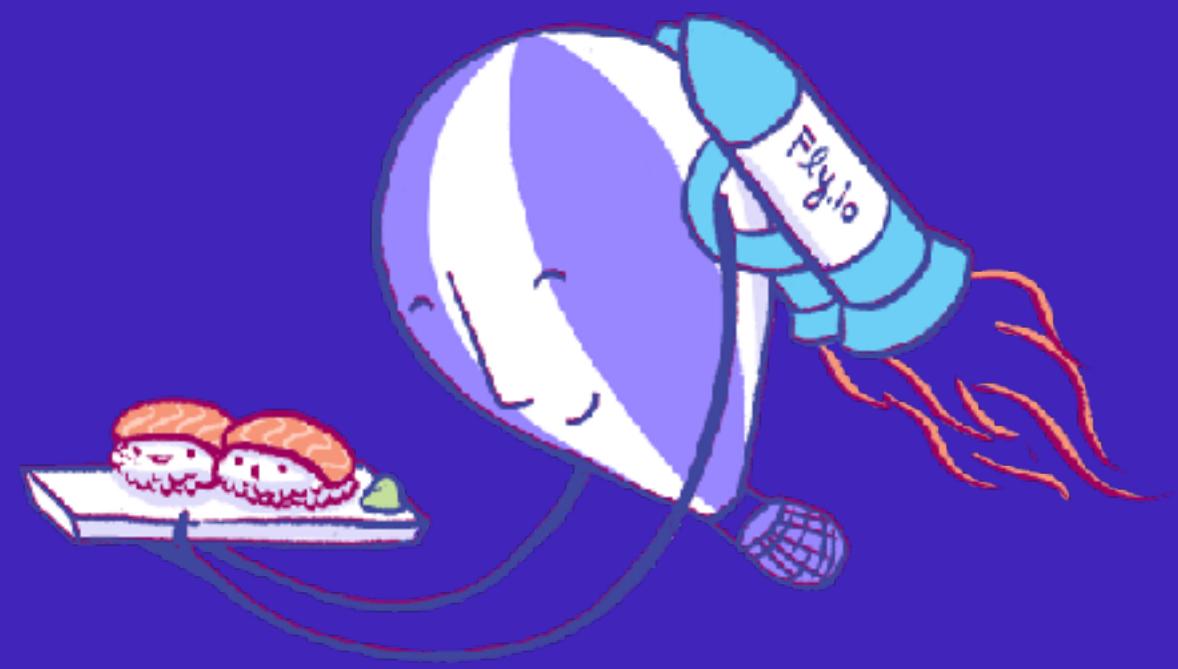
We didn't talk about
memory!



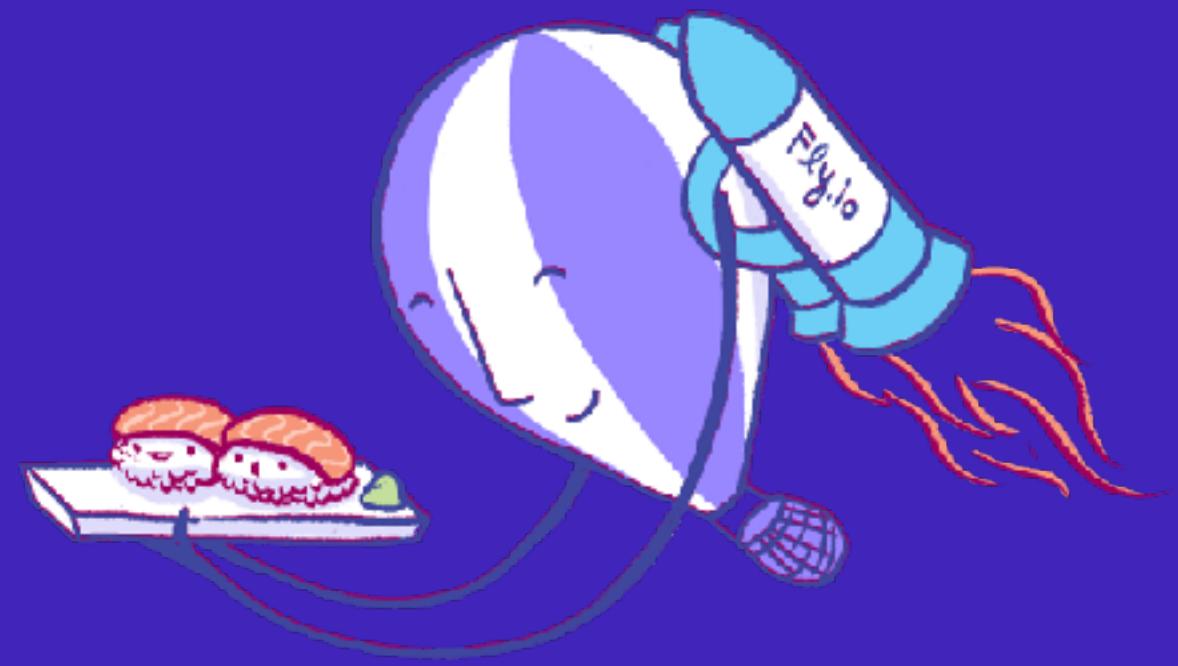
Danger: cross the boundary
of an isolate and you can
screw up garbage collection.



JavaScript: You should get comfy with Buffers, SharedArrayBuffers, and text encoding.



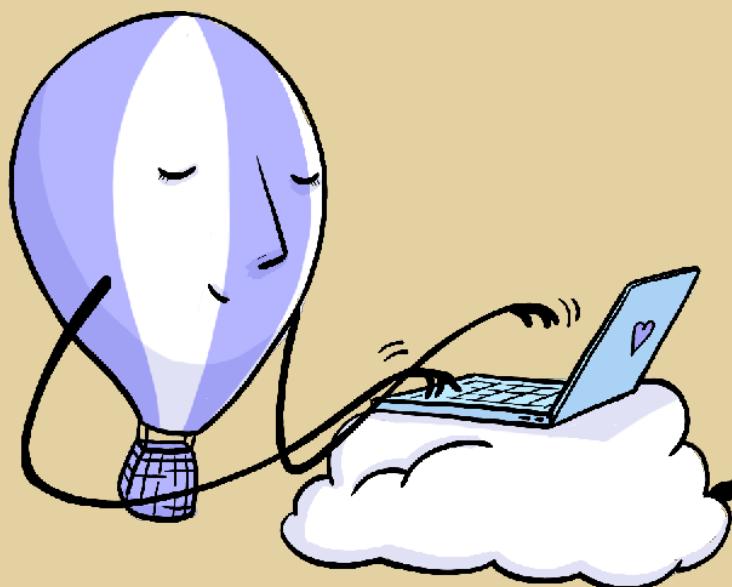
Execution Control: isolated-vm
takes care of a lot of this for us.
But you should understand what's
happening.





Want to know more?

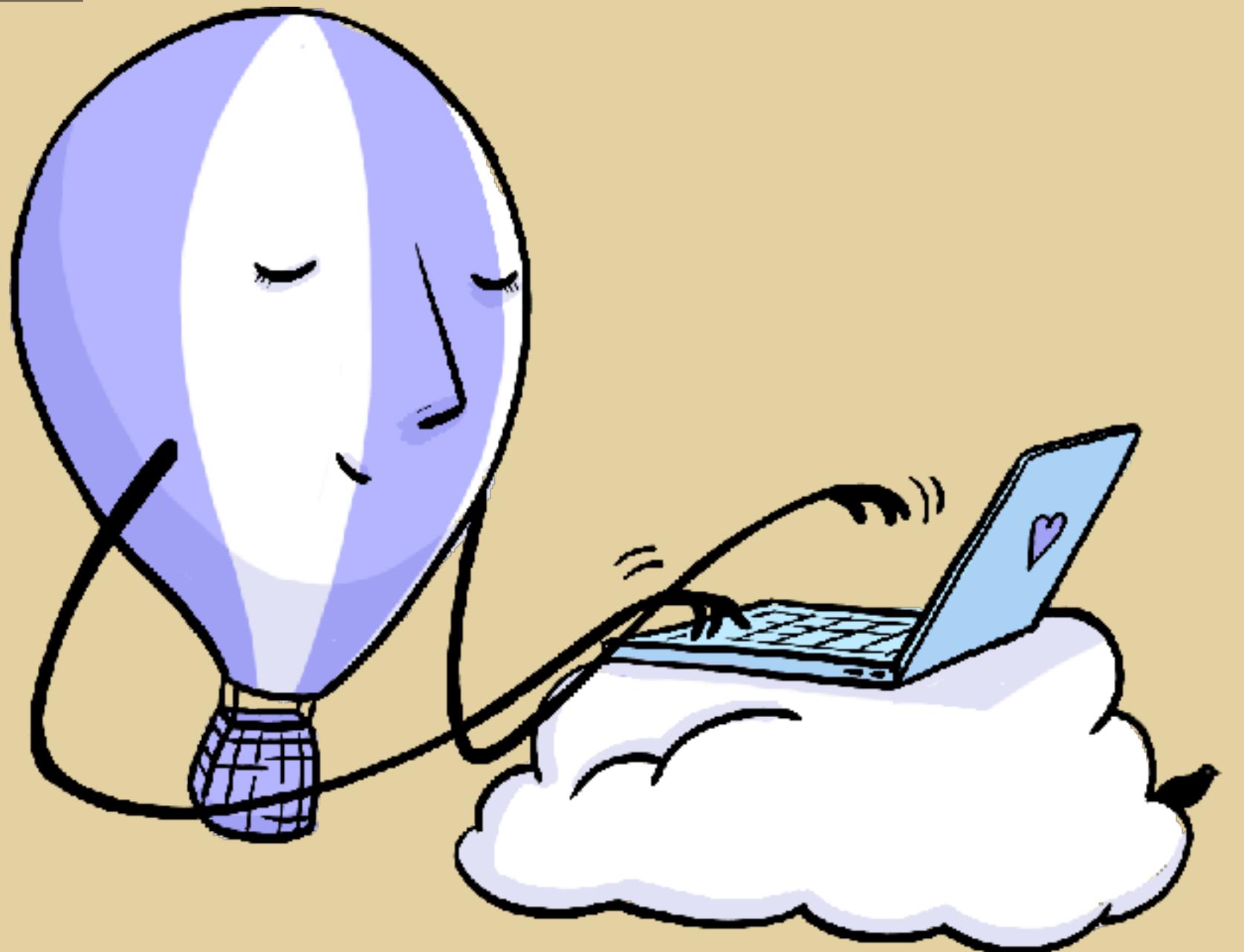
- ▶ isolated-vm: github.com/laverdet/isolated-vm
- ▶ v8 embedding guide: github.com/v8/v8/wiki/Getting%20Started%20with%20Embedding
- ▶ fetch API: developer.mozilla.org/en-US/docs/Web/API/Fetch_API





Help improve this talk

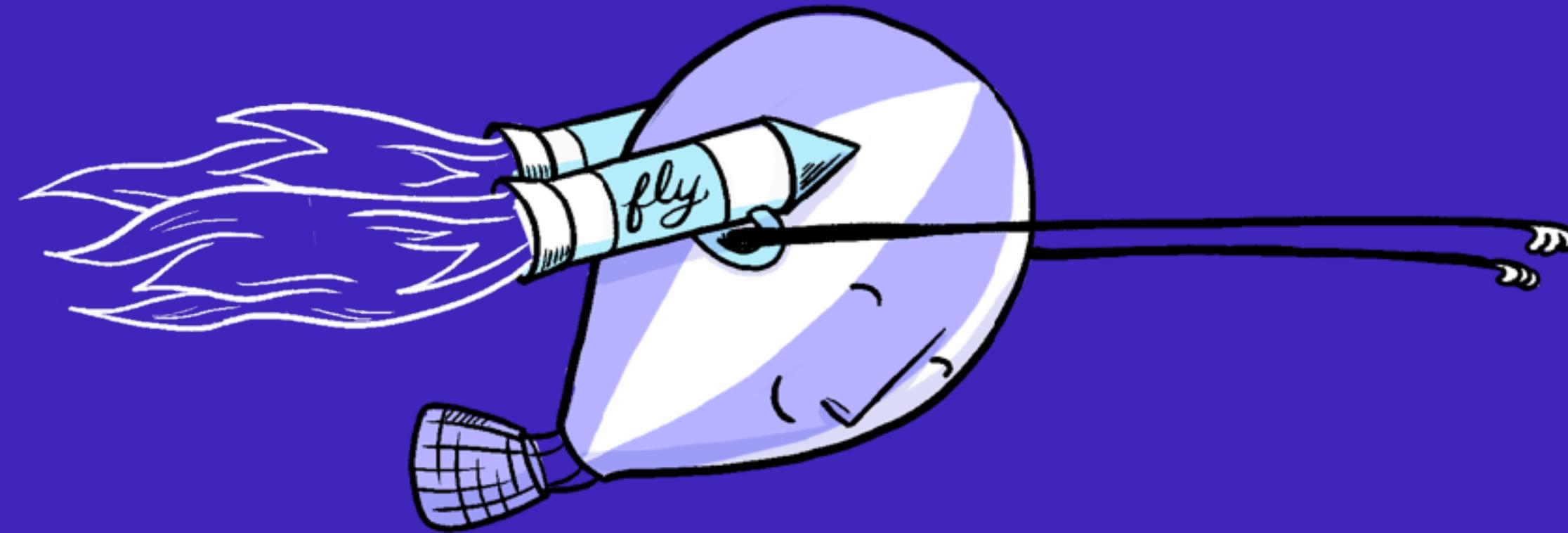
- ▶ Email: kurt@fly.io
- ▶ Twitter: @mrkurt
- ▶ Talk to me after, I love notes, corrections, pointers, and cutting sarcasm.



Thank you!

Thank you!

Go try Fly: <https://fly.io>



Build and test locally. Deploy close to all your users the world, instantly.