adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

27th – 29th SEPTEMBER 2021

The shift to the edge

Jakub Wądołowski, diva-e (@jwadolowski)

Web standards

HTML

JavaScript

CSS

Web standards

HTML

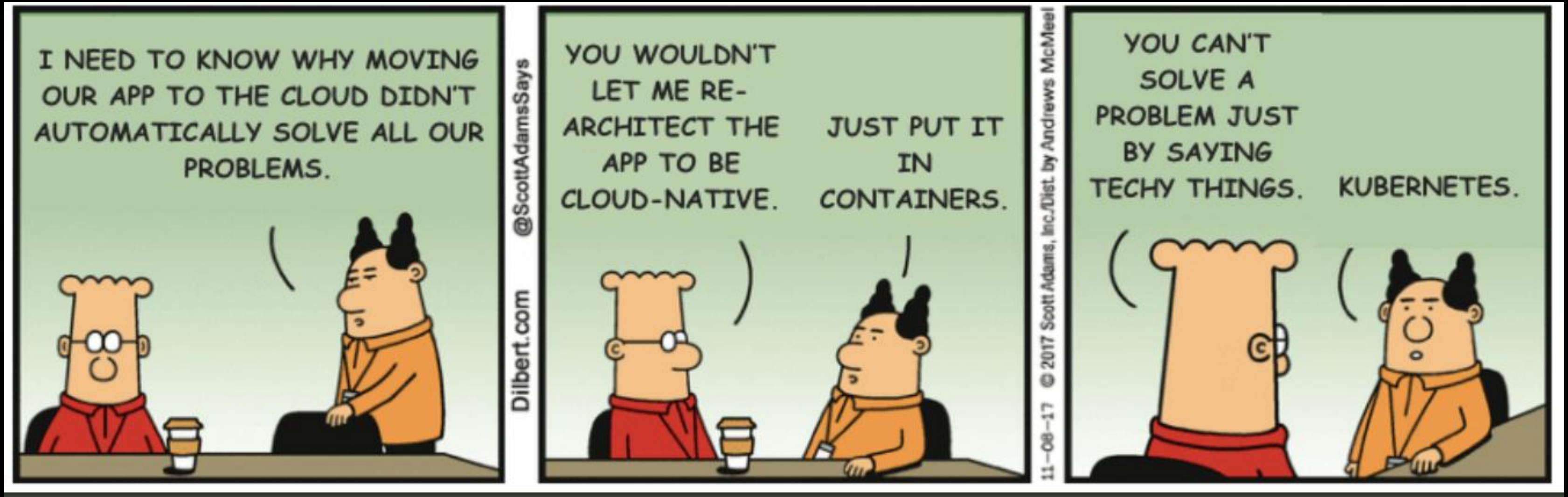
JavaScript

CSS

WebAssembly

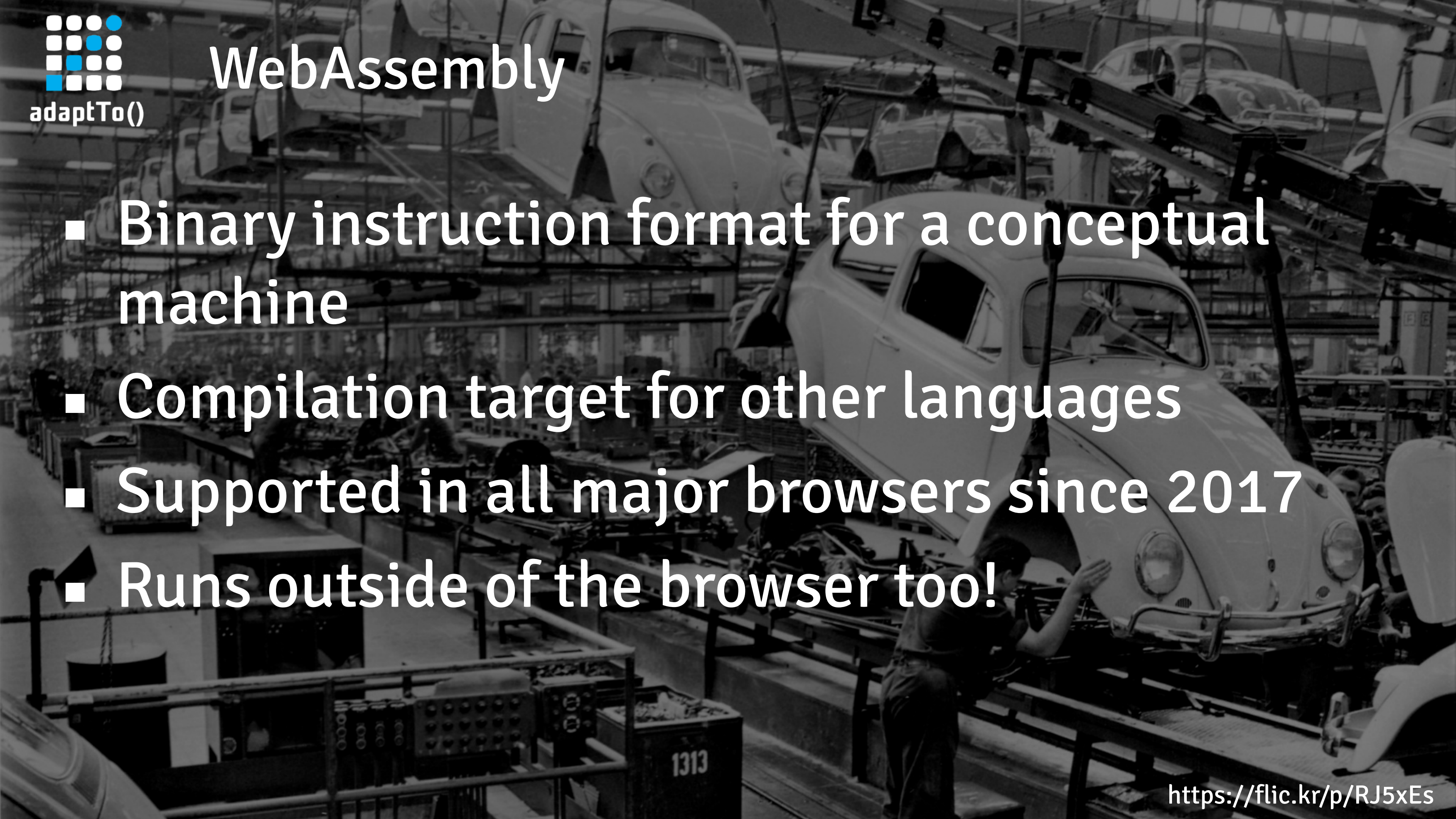
WebAssembly (WASM)

Buzzword bingo


I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
Dilbert.com
@ScottAdamsSays
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.

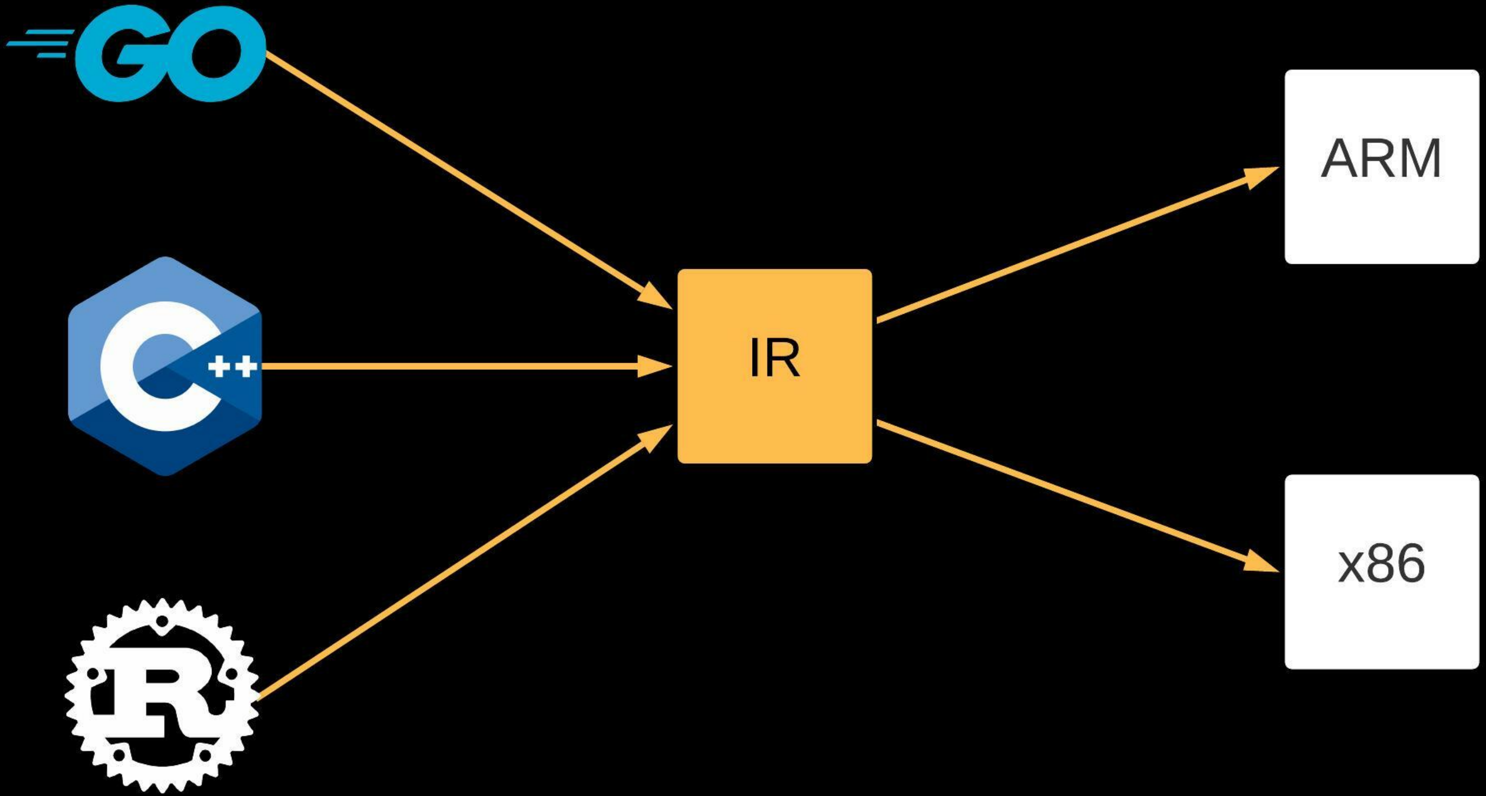
adaptTo()

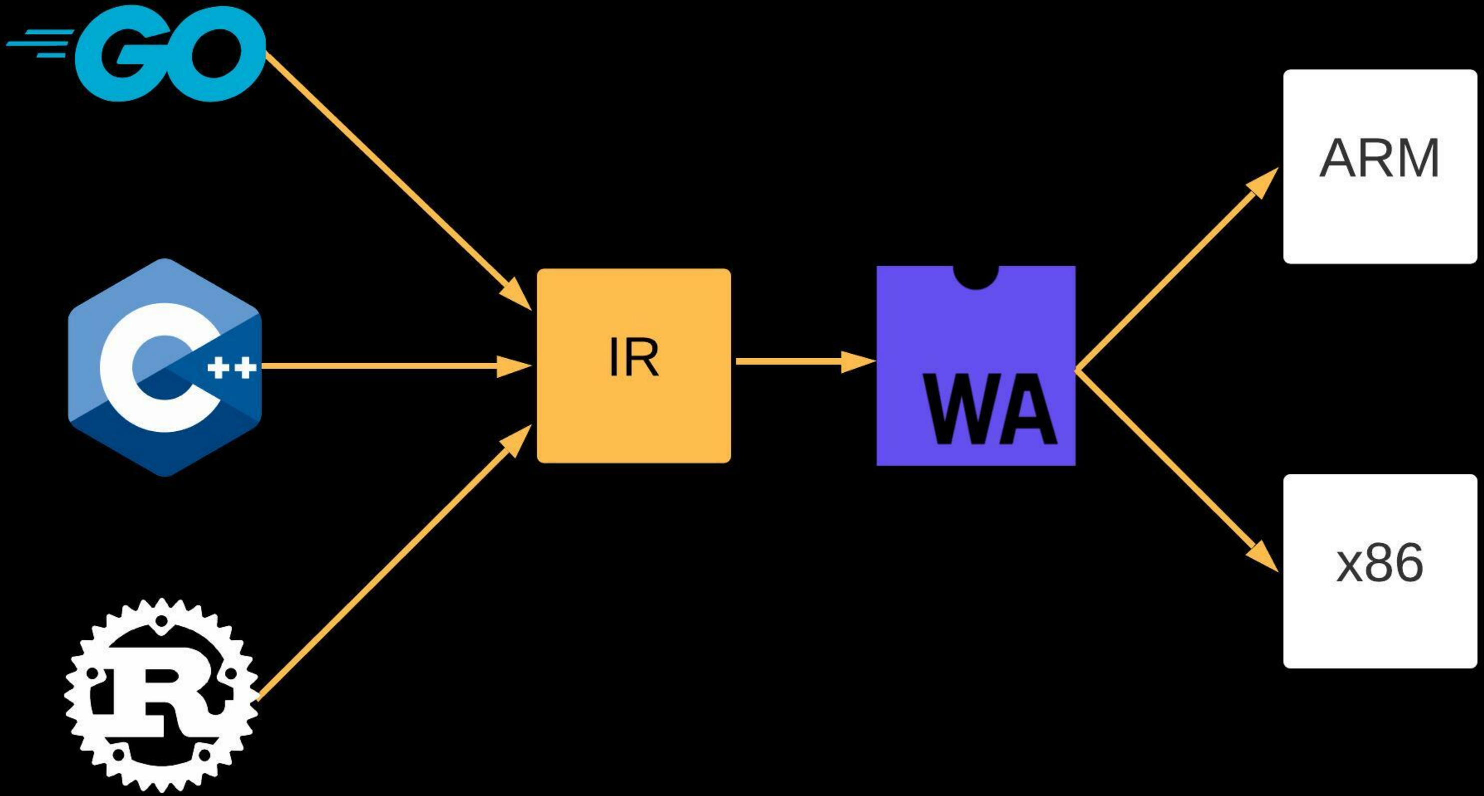
WebAssembly



- Binary instruction format for a conceptual machine
- Compilation target for other languages
- Supported in all major browsers since 2017
- Runs outside of the browser too!

https://flic.kr/p/RJ5xEs

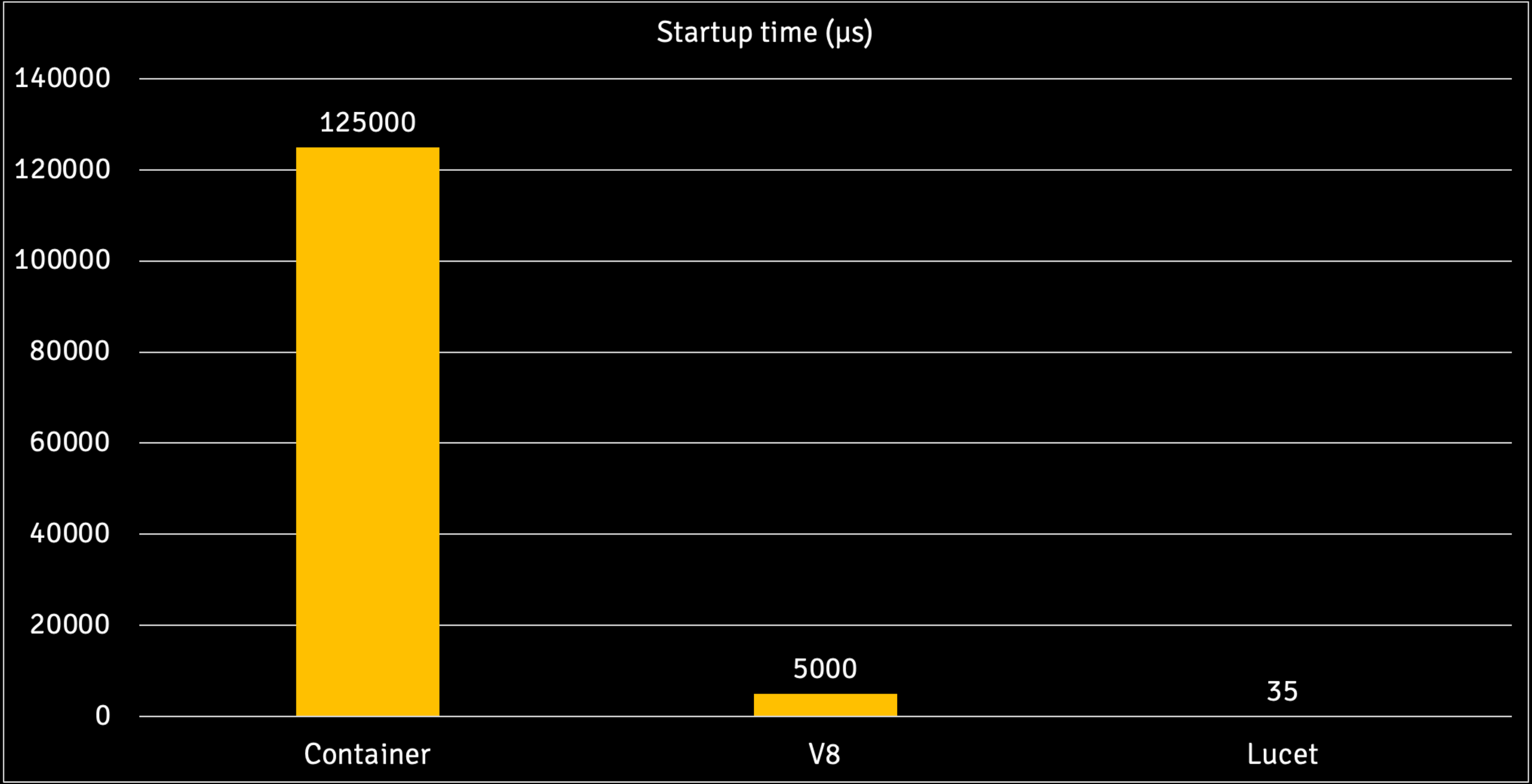

IR
ARM
x86


GO
C++
R
IR
WA
ARM
x86

WASM features

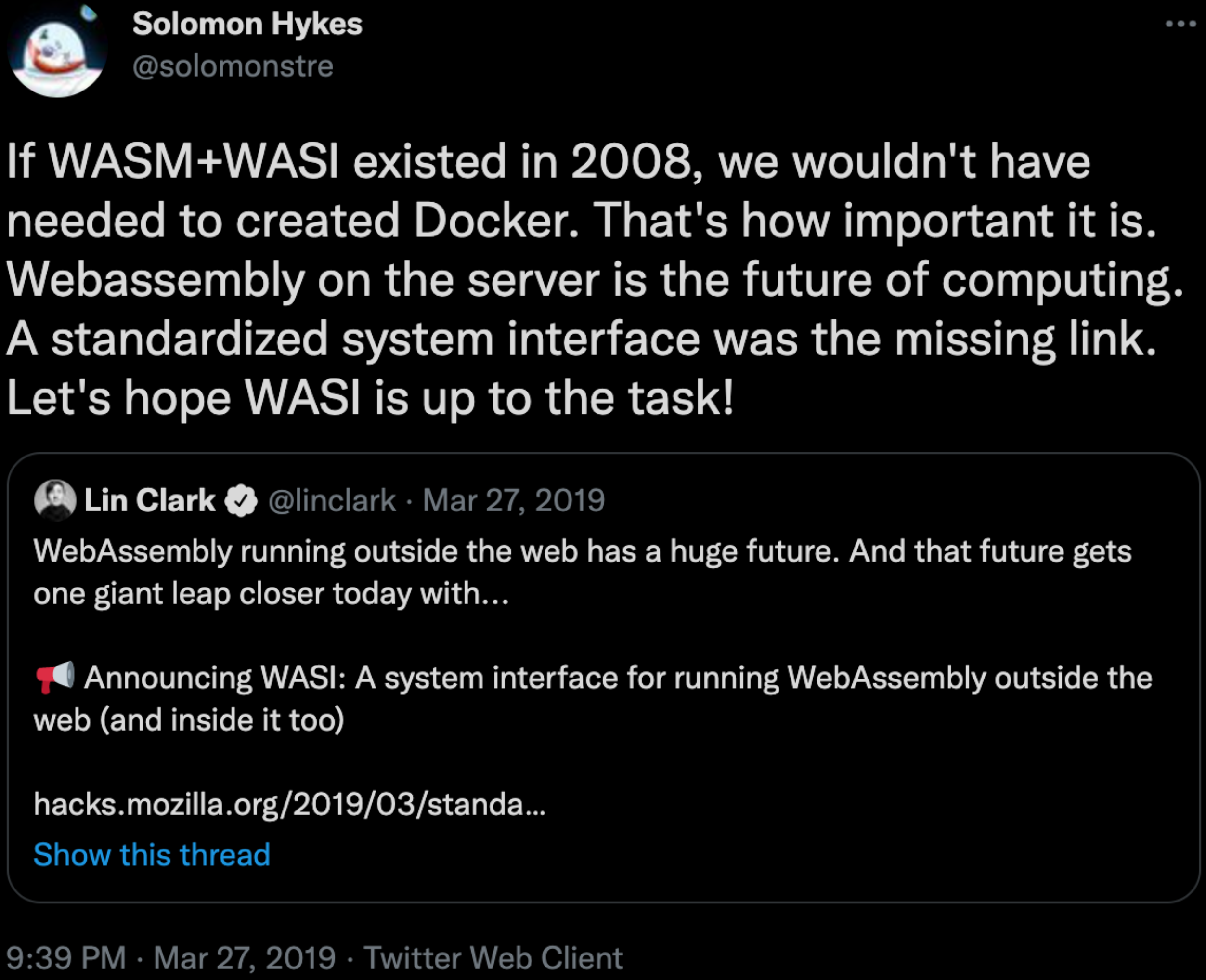
- Simple
- Small & portable
- Streamable
- Fast & secure

https://flic.kr/p/e7noPK


Startup time (µs)
140000
120000
100000
80000
60000
40000
20000
0
125000
5000
35
Container
V8
Lucet

WebAssembly Text (WAT)

```
(module
  (func $add (param $lhs i32) (param $rhs i32) (result i32)
    local.get $lhs
    local.get $rhs
    i32.add)
  (export "add" (func $add))
)
```



Solomon Hykes
@solomonstre

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

Lin Clark @linclark · Mar 27, 2019
WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

hacks.mozilla.org/2019/03/standa...
Show this thread

9:39 PM · Mar 27, 2019 · Twitter Web Client

adaptTo()

WASM in the wild

envoy

unity

ebay

fastly®

CLOUDFLARE®

1Password

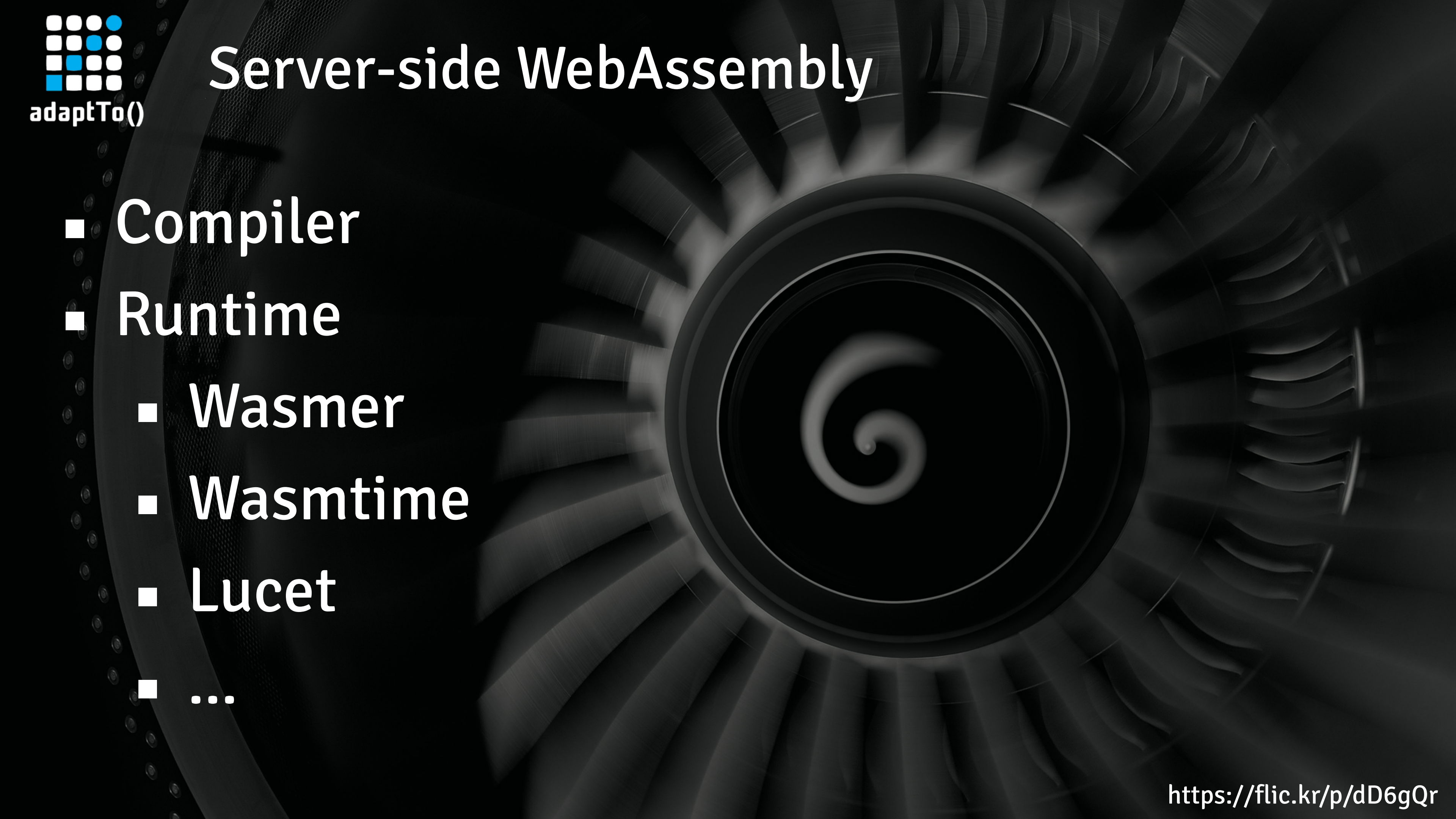
shopify

adaptTo()

WebAssembly beyond the browser

Server-side WebAssembly

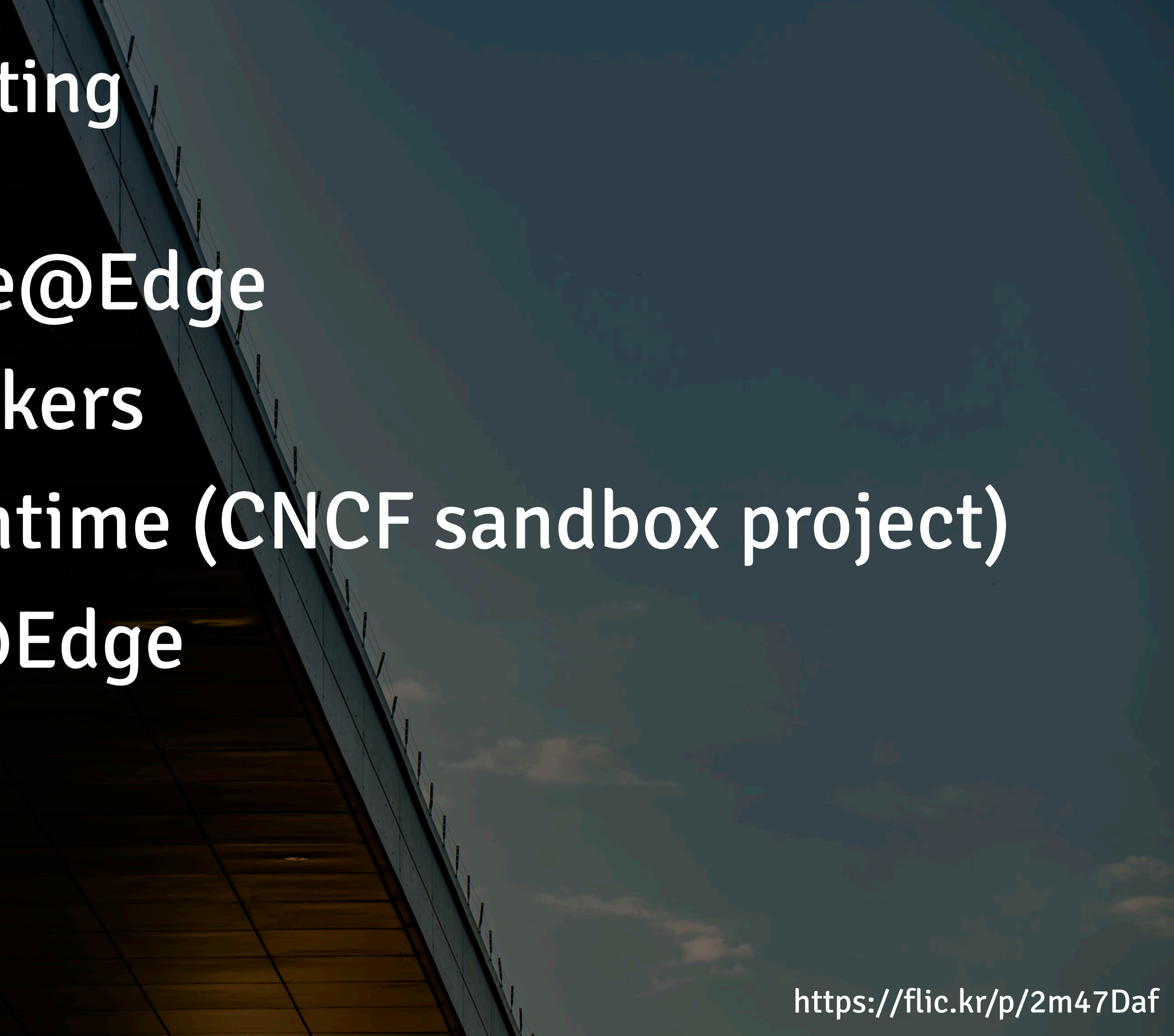
- Compiler
- Runtime
  - Wasmer
  - Wasmtime
  - Lucet
  - ...



https://flic.kr/p/dD6gQr

adaptTo()

Edge computing



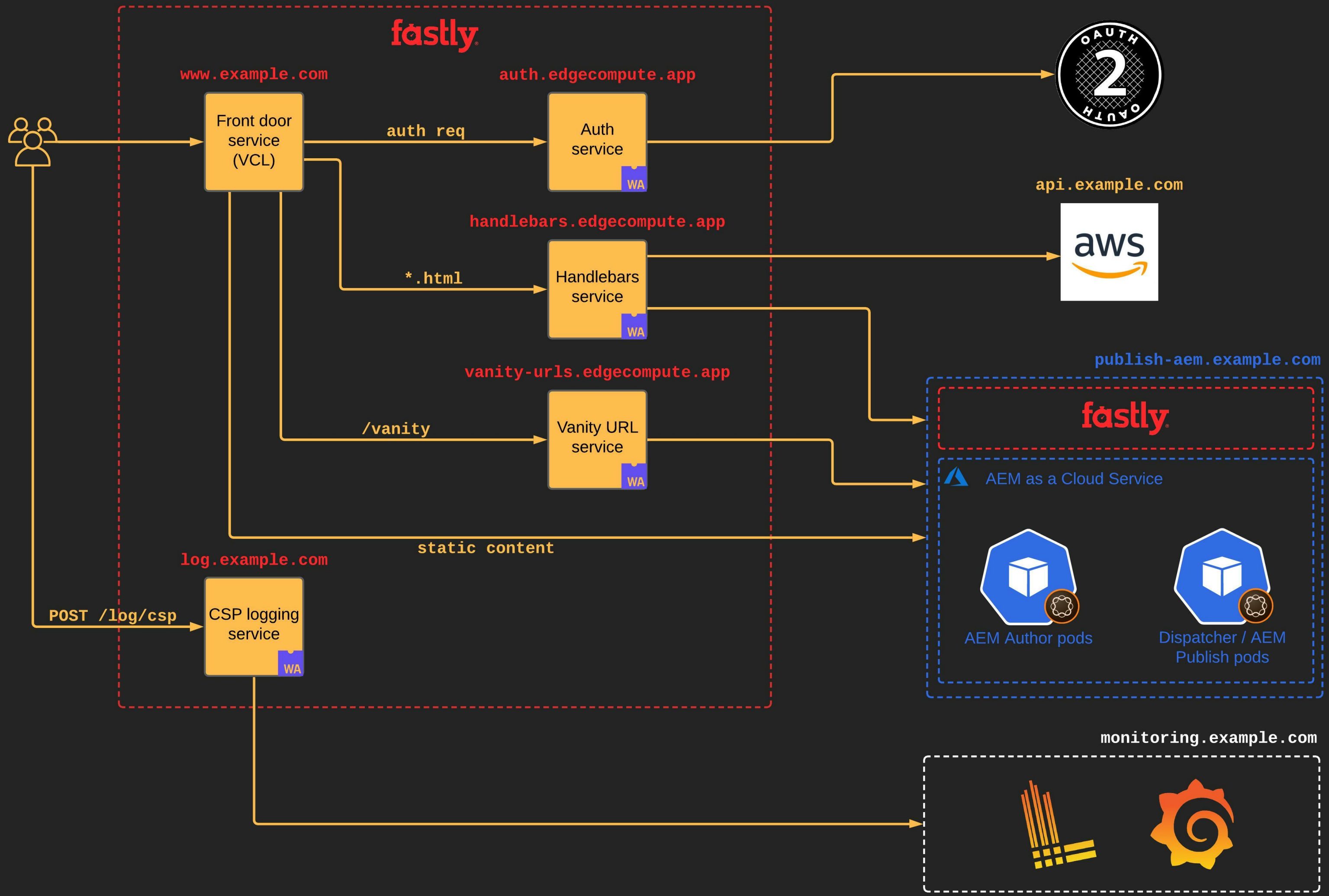
- Fastly Compute@Edge
- Cloudflare Workers
- WasmEdge Runtime (CNCF sandbox project)
- AWS Lambda@Edge
- ...

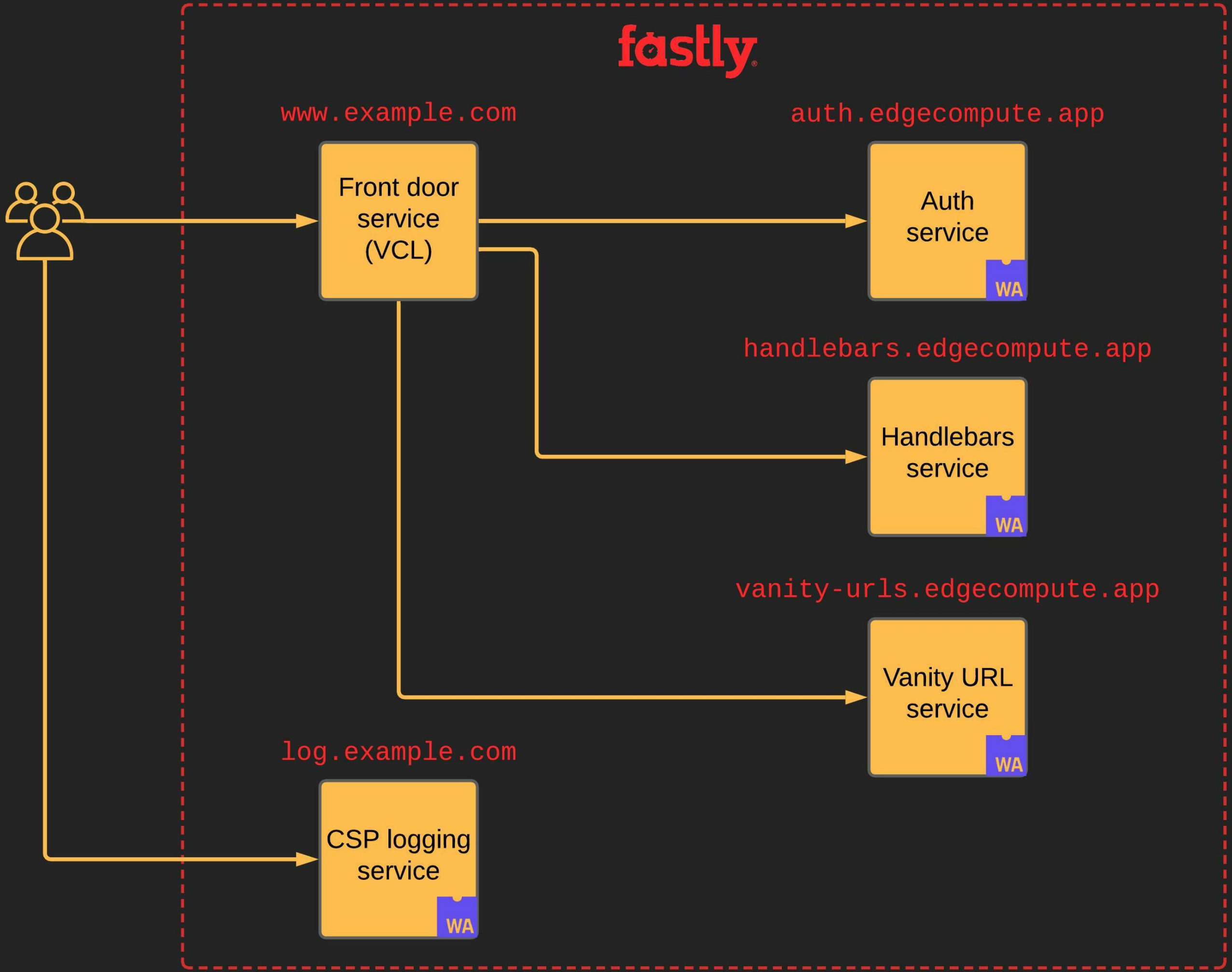
https://flic.kr/p/2m47Daf

AEM as a Cloud Service

- Paradigm shift
- Mindset change
- New challenges and opportunities

https://flic.kr/p/FWq4ez

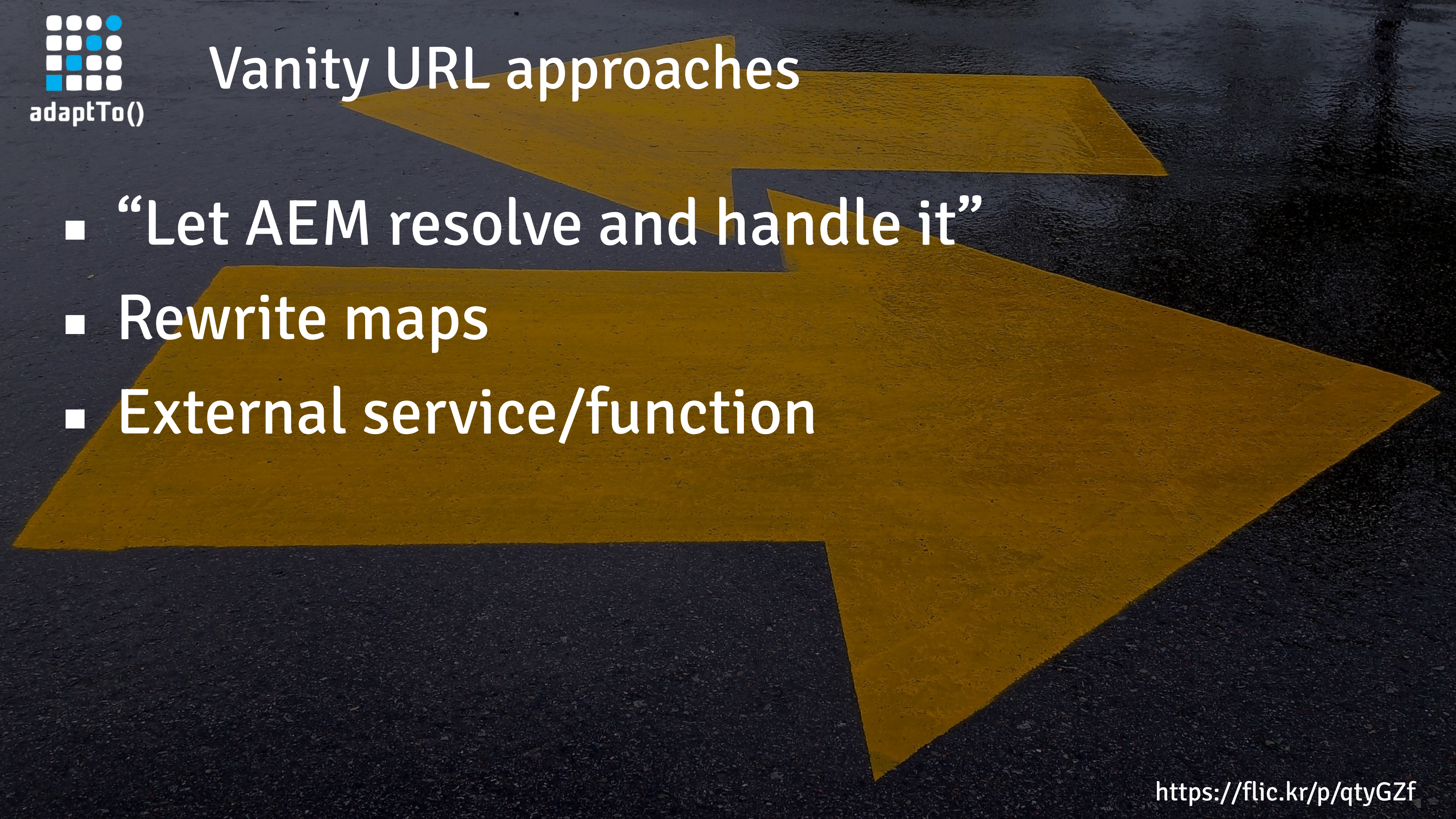

fastly
www.example.com
auth.edgecompute.app
Front door service (VCL)
auth req
Auth service
WA
handlebars.edgecompute.app
*.html
Handlebars service
WA
vanity-urls.edgecompute.app
/vanity
Vanity URL service
WA
static content
log.example.com
POST /log/csp
CSP logging service
WA
OAUTH 2
api.example.com
aws
publish-aem.example.com
fastly
AEM as a Cloud Service
AEM Author pods
Dispatcher / AEM Publish pods
monitoring.example.com


fastly
www.example.com
Front door service (VCL)
auth.edgecompute.app
Auth service
WA
handlebars.edgecompute.app
Handlebars service
WA
vanity-urls.edgecompute.app
Vanity URL service
WA
log.example.com
CSP logging service
WA

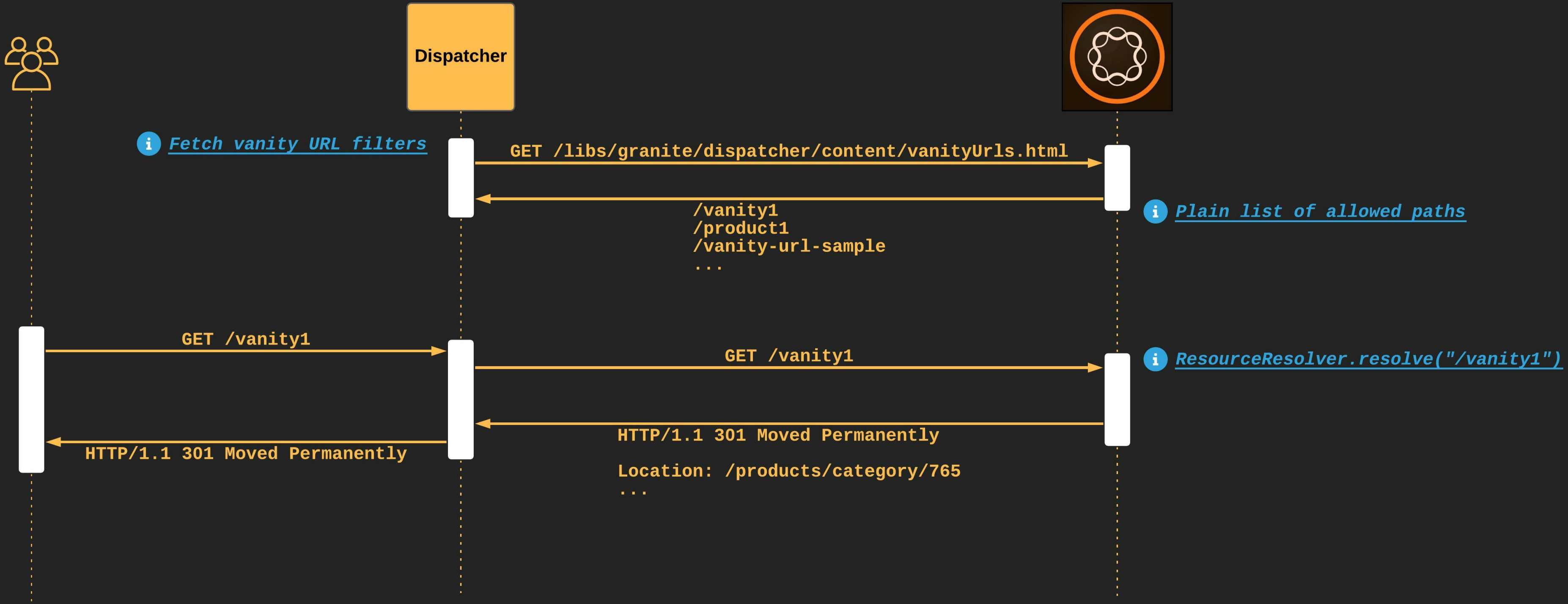
adaptTo()

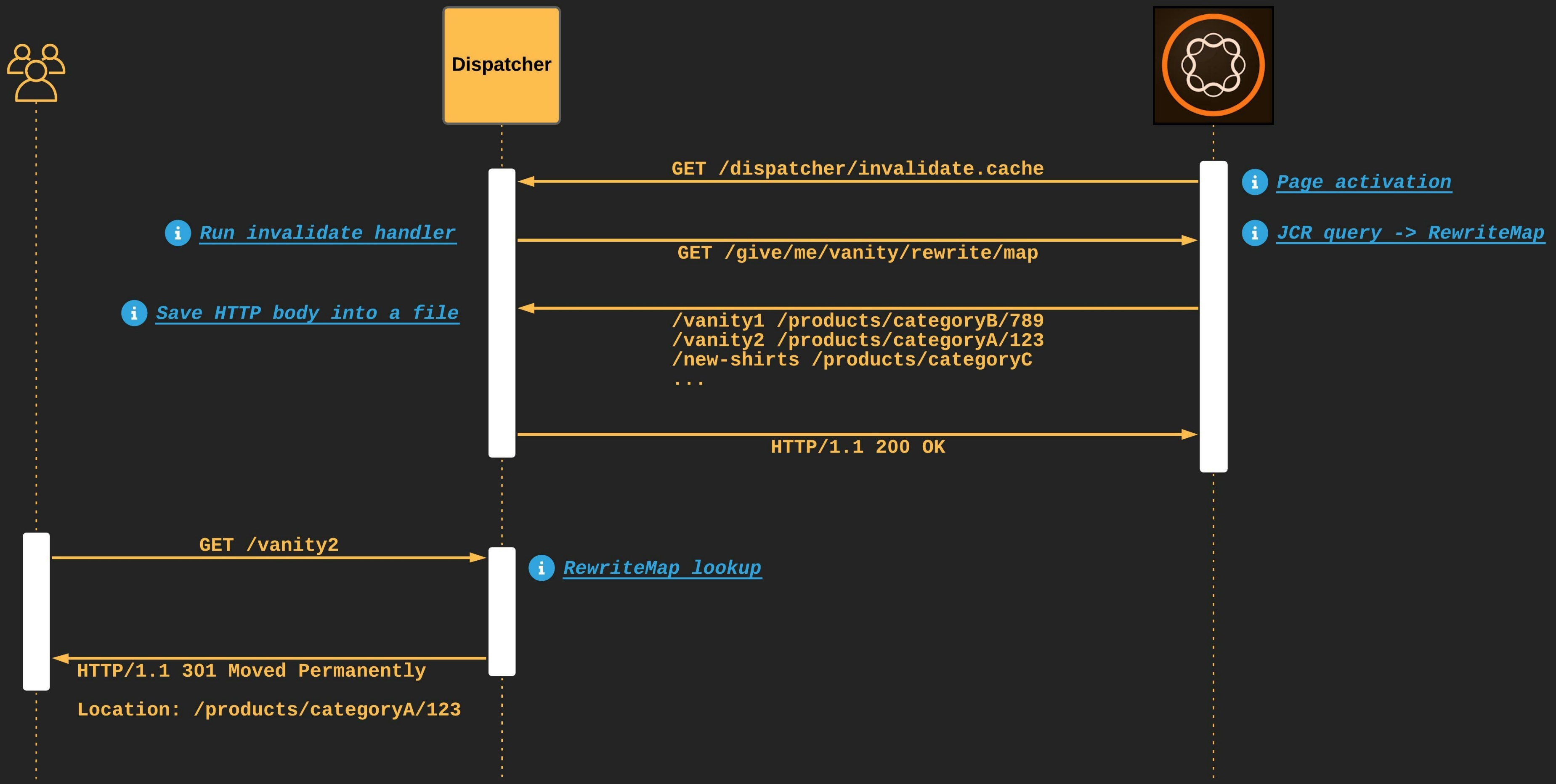
Vanity URL approaches

- “Let AEM resolve and handle it”
- Rewrite maps
- External service/function



https://flic.kr/p/qtyGZf


Dispatcher
Fetch vanity URL filters
GET /libs/granite/dispatcher/content/vanityUrls.html
/vanity1
/product1
/vanity-url-sample
...
Plain list of allowed paths
GET /vanity1
GET /vanity1
ResourceResolver.resolve("/vanity1")
HTTP/1.1 301 Moved Permanently
Location: /products/category/765
...
HTTP/1.1 301 Moved Permanently


Dispatcher
GET /dispatcher/invalidate.cache
Page activation
Run invalidate handler
GET /give/me/vanity/rewrite/map
JCR query -> RewriteMap
Save HTTP body into a file
HTTP/1.1 200 OK
GET /vanity2
RewriteMap lookup
HTTP/1.1 301 Moved Permanently

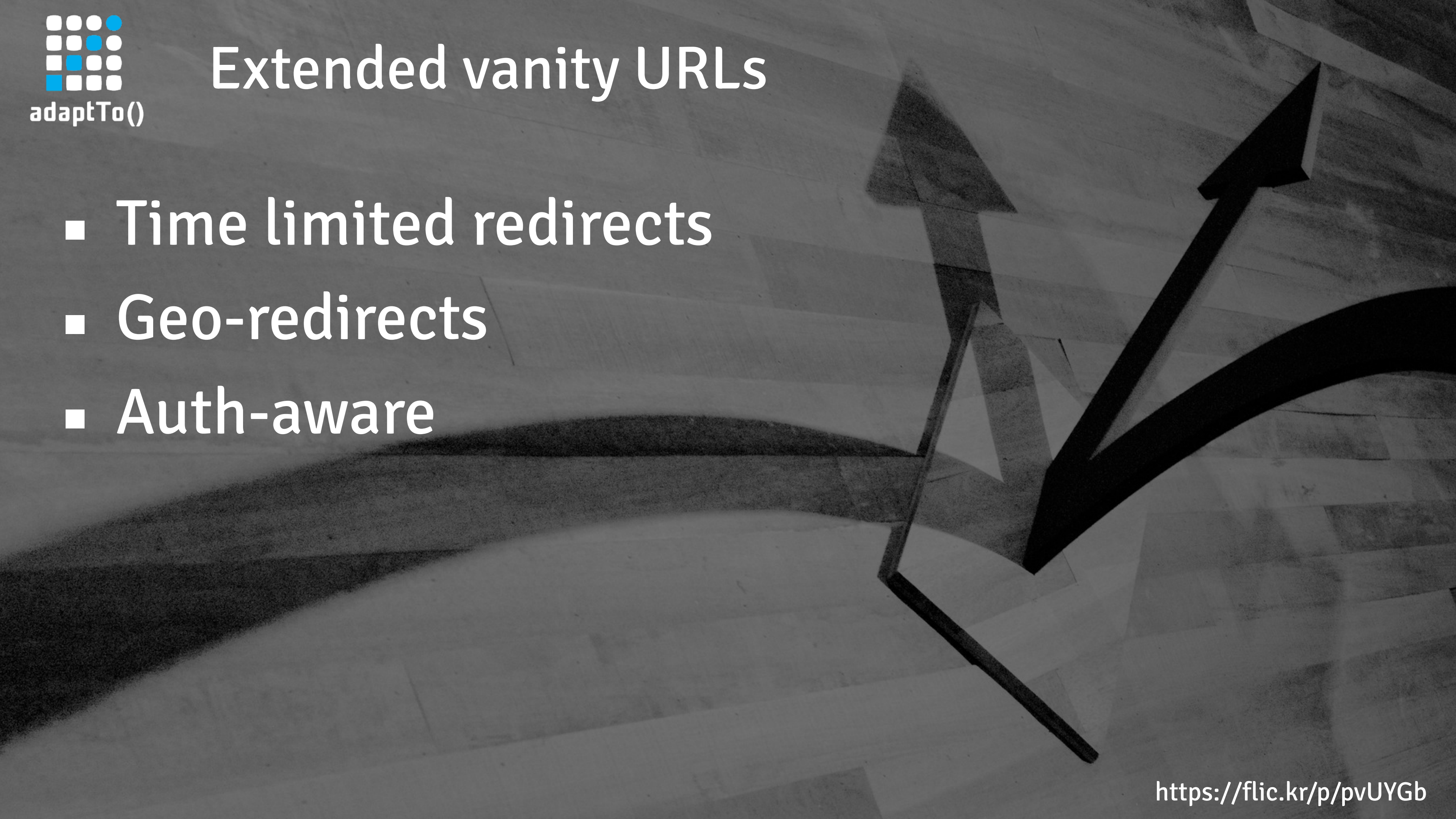
```
/vanity1 /products/categoryB/789
/vanity2 /products/categoryA/123
/new-shirts /products/categoryC
...
```

```
Location: /products/categoryA/123
```

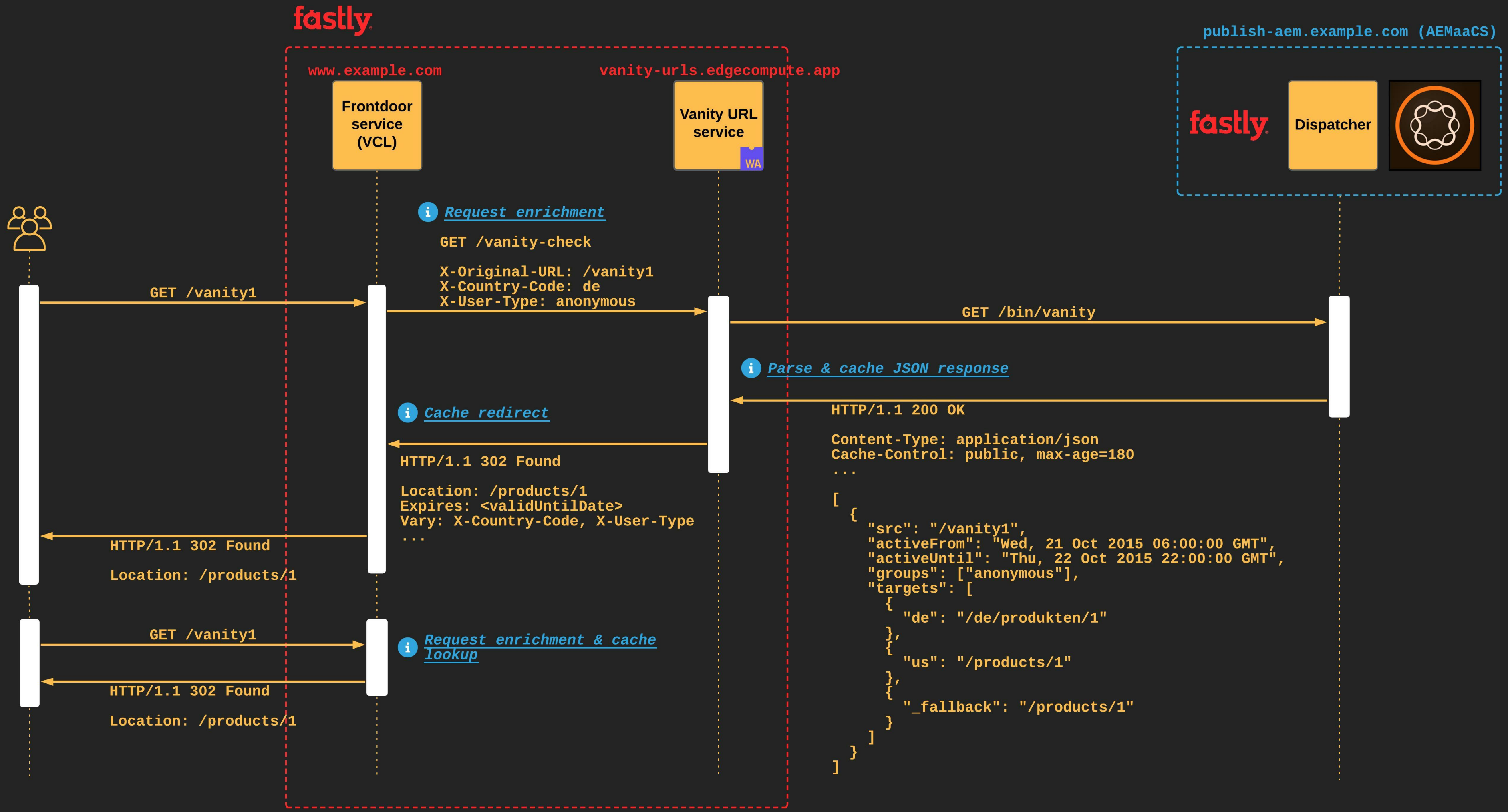
adaptTo()

Extended vanity URLs

- Time limited redirects
- Geo-redirects
- Auth-aware



https://flic.kr/p/pvUYGb


fastly
www.example.com
Frontdoor service (VCL)
vanity-urls.edgecompute.app
Vanity URL service
WA
publish-aem.example.com (AEMaaCS)
fastly
Dispatcher
Request enrichment
Parse & cache JSON response
Cache redirect
Request enrichment & cache lookup

```
GET /vanity1
```

```
GET /vanity-check

X-Original-URL: /vanity1
X-Country-Code: de
X-User-Type: anonymous
```

```
GET /bin/vanity
```

```
HTTP/1.1 200 OK

Content-Type: application/json
Cache-Control: public, max-age=180
...

[
  {
    "src": "/vanity1",
    "activeFrom": "Wed, 21 Oct 2015 06:00:00 GMT",
    "activeUntil": "Thu, 22 Oct 2015 22:00:00 GMT",
    "groups": ["anonymous"],
    "targets": [
      {
        "de": "/de/produkten/1"
      },
      {
        "us": "/products/1"
      },
      {
        "_fallback": "/products/1"
      }
    ]
  }
]
```

```
HTTP/1.1 302 Found

Location: /products/1
Expires: <validUntilDate>
Vary: X-Country-Code, X-User-Type
...
```

```
HTTP/1.1 302 Found

Location: /products/1
```

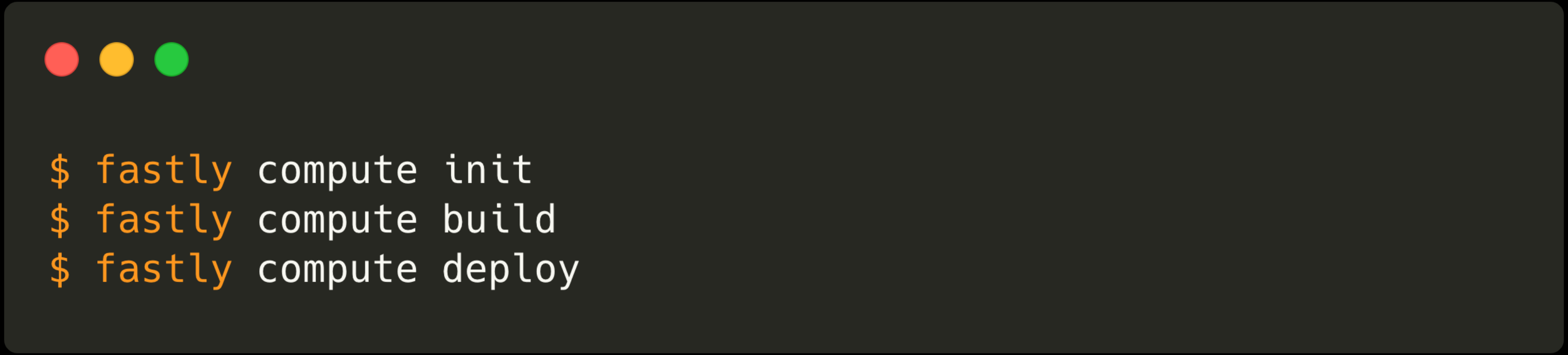
```
GET /vanity1
```

```
HTTP/1.1 302 Found

Location: /products/1
```

adaptTo()

WASM service development flow



```
$ fastly compute init
$ fastly compute build
$ fastly compute deploy
```

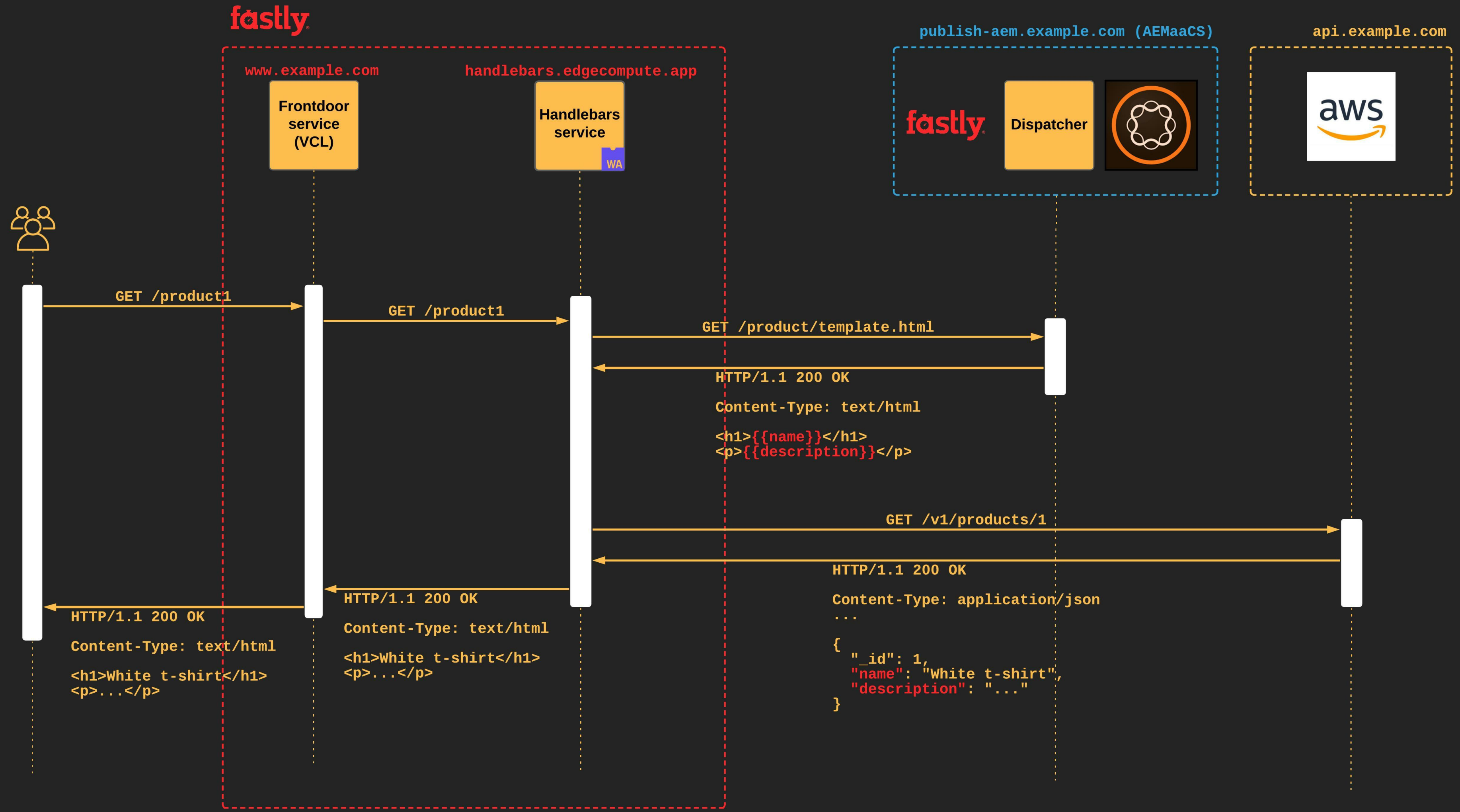
WASM service deployment demo

https://flic.kr/p/27Py5HK

Content stitching at the edge

- Edge Side Includes (ESI)
- Templating libraries
  - Handlebars
  - Tera (~Jinja2)
  - Liquid
  - ...

https://flic.kr/p/opCnxv



```
HTTP/1.1 200 OK

Content-Type: text/html

<h1>{{name}}</h1>
<p>{{description}}</p>
```

```
HTTP/1.1 200 OK

Content-Type: application/json
...

{
  "_id": 1,
  "name": "White t-shirt",
  "description": "..."
}
```

```
HTTP/1.1 200 OK

Content-Type: text/html

<h1>White t-shirt</h1>
<p>...</p>
```

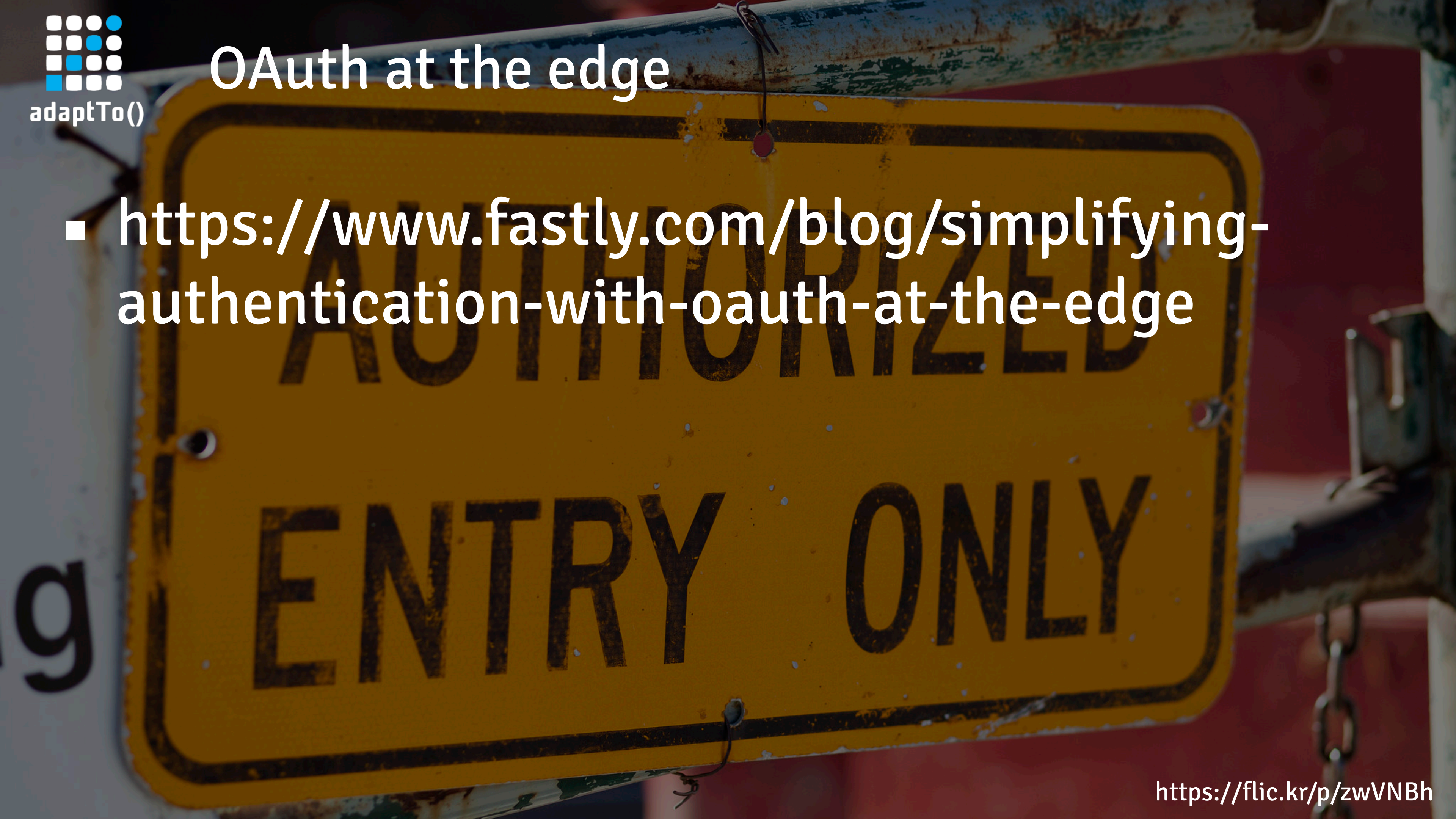
```
HTTP/1.1 200 OK

Content-Type: text/html

<h1>White t-shirt</h1>
<p>...</p>
```

OAuth at the edge

- https://www.fastly.com/blog/simplifying-authentication-with-oauth-at-the-edge

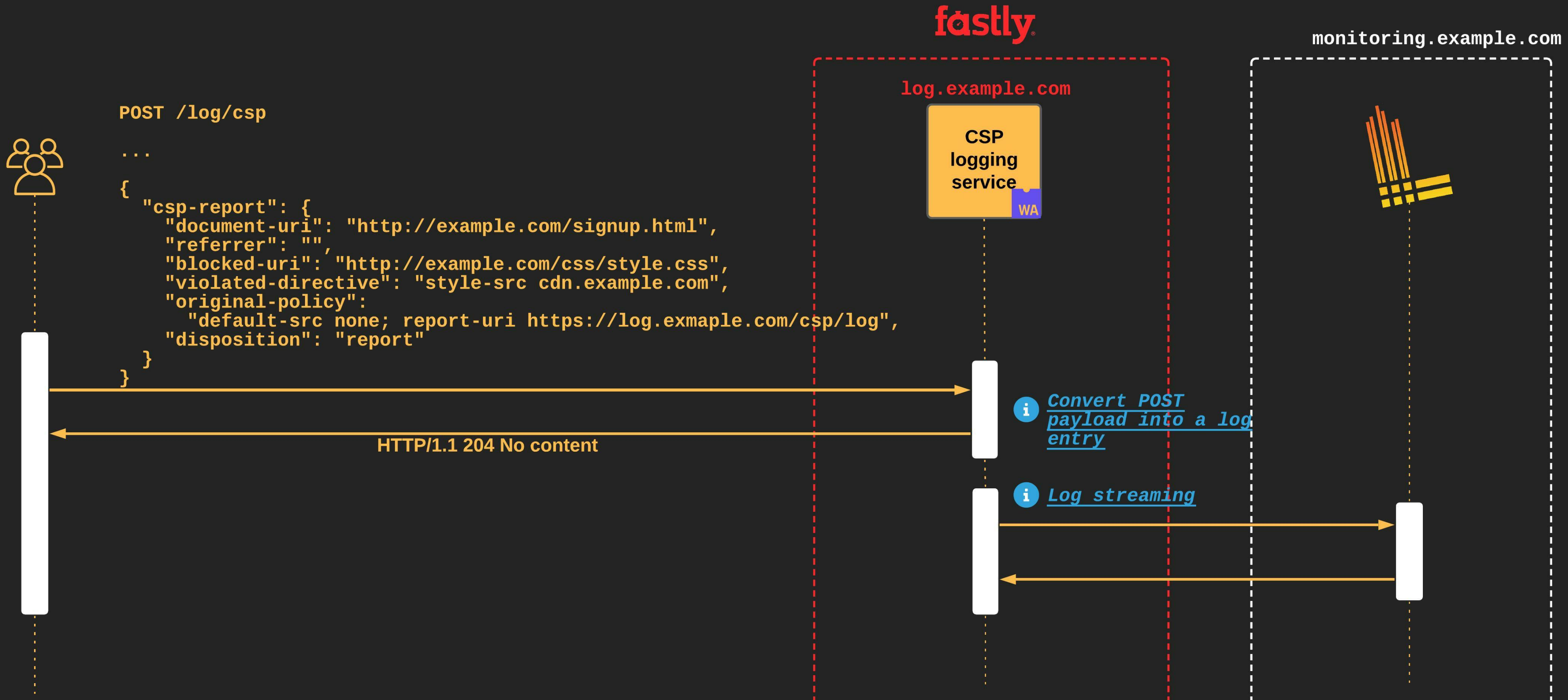


https://flic.kr/p/zwVNBh

Content Security Policy (CSP) logging

- Browser security feature
- Violation == POST request
- Log collection

https://flic.kr/p/iYxHLh



```
POST /log/csp

...

{
  "csp-report": {
    "document-uri": "http://example.com/signup.html",
    "referrer": "",
    "blocked-uri": "http://example.com/css/style.css",
    "violated-directive": "style-src cdn.example.com",
    "original-policy":
      "default-src none; report-uri https://log.exmaple.com/csp/log",
    "disposition": "report"
  }
}
```

adaptTo()

Compute@Edge use cases

- Content stitching (incl. SSR)
- A/B testing
- Authentication
- Personalisation
- Ad targeting
- Games?!

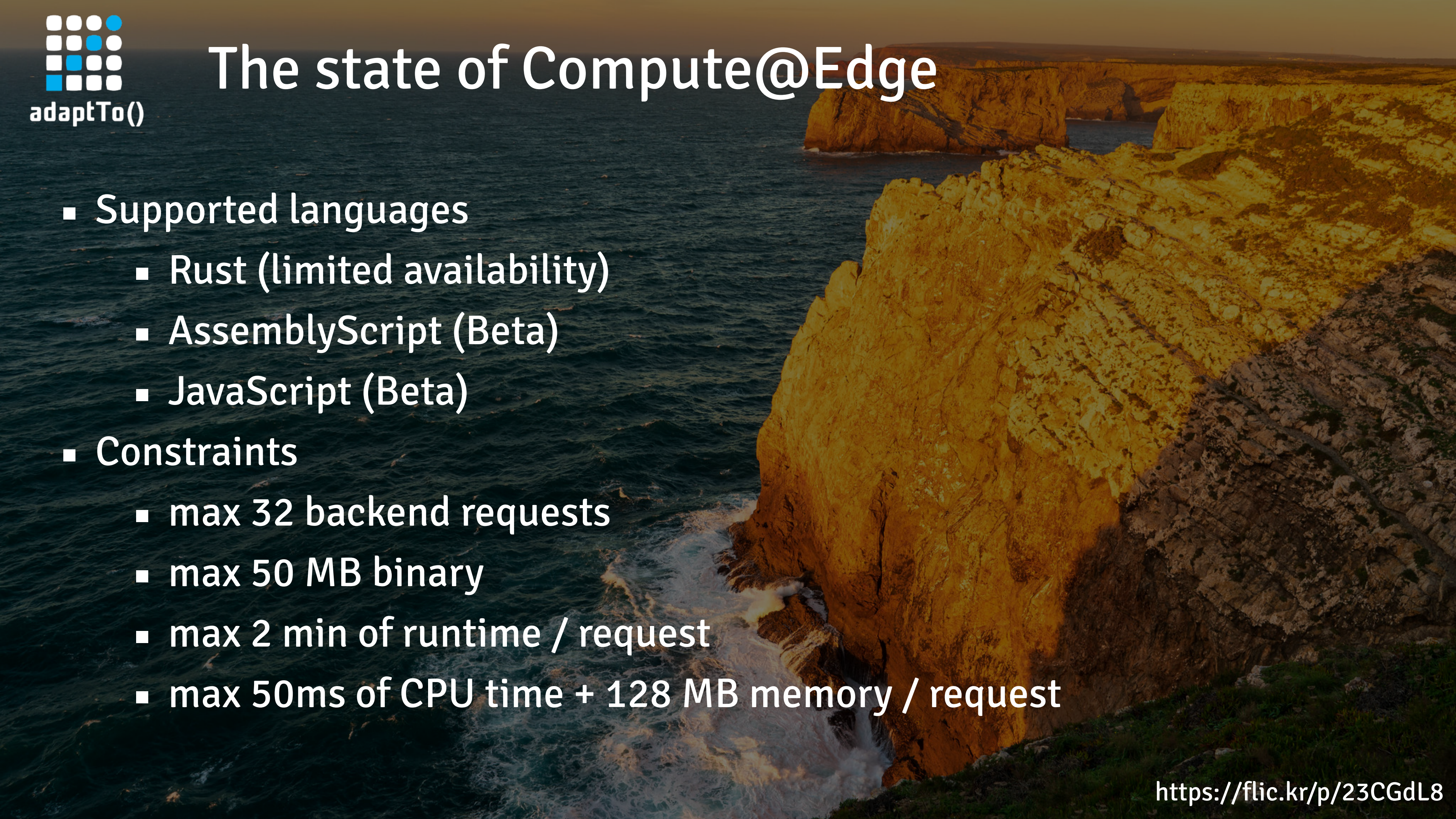
https://flic.kr/p/XqGdcM

DOOM

A port of the original DOOM to Compute@Edge. This demo was created to push the boundaries of the platform and inspire new ideas!

Try it out

DOOM was a game developed in 1993 by id software and released in December of that year. Id software had made a living developing high quality 2D games, but with Wolfenstein in 1992 and then DOOM the following year, they made a historic leap into 3D, taking advantage of the quickly evolving PC hardware



adaptTo()

The state of Compute@Edge

- Supported languages
  - Rust (limited availability)
  - AssemblyScript (Beta)
  - JavaScript (Beta)
- Constraints
  - max 32 backend requests
  - max 50 MB binary
  - max 2 min of runtime / request
  - max 50ms of CPU time + 128 MB memory / request

https://flic.kr/p/23CGdL8

Thank you!