From Cloud to Edge Computing

Unleashing the power of WebAssembly at the edge

Alex Casalboni

Developer Advocate, Edgee
edgee.cloud

Agenda for today 📅

- What’s edge computing
- Cloud & edge evolution
- Wasm support @ CDNs
- Wasm @ Edge(e)

cloud academy
A QA COMPANY

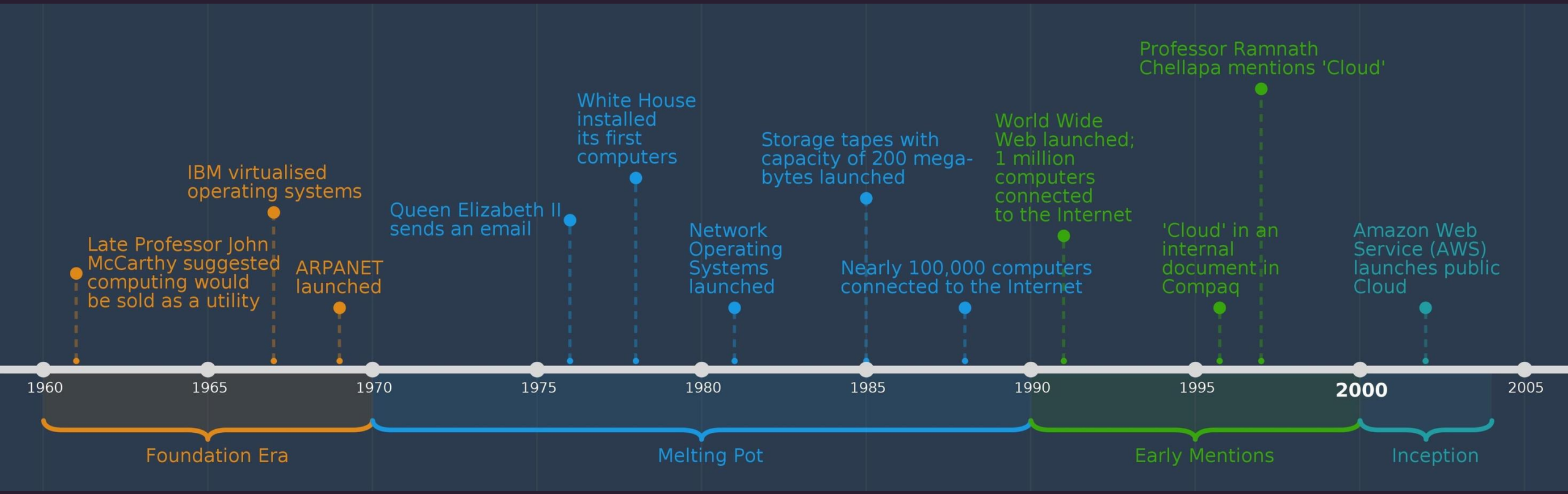
aws

edgee

Background	Web Developer & Software Engineering
Fun facts	Startupper for 5+ years, visited 41 countries
Hobbies	Reading, playing music, snowboarding
Proud of	Father of one

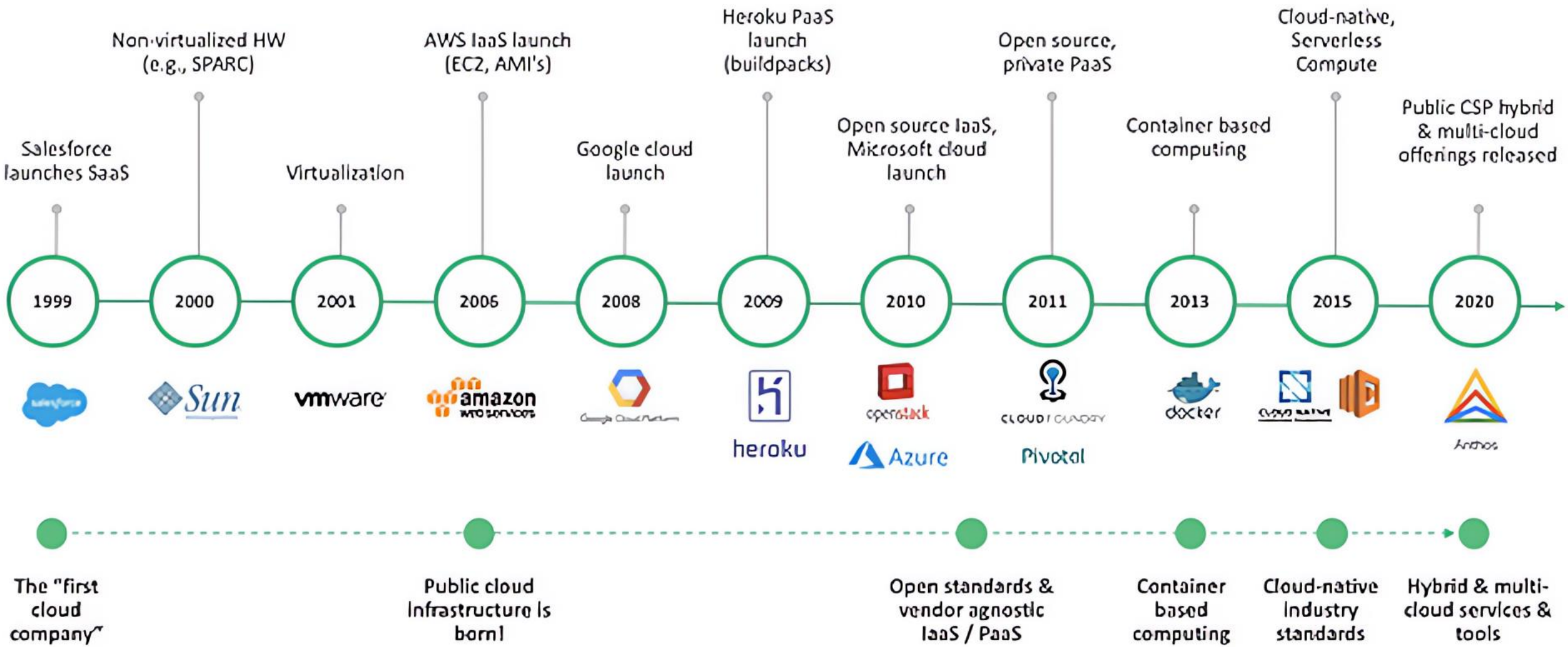
So what is exactly edge computing?

Cloud history



bcs.org/articles-opinion-and-research/history-of-the-cloud/

Cloud (recent) history



CDNs evolved too!

	Akamai	Amazon CloudFront	CLOUDFLARE	fastly
Founded	1998	2008	2009	2011

CDNs evolved too!

		 Amazon CloudFront		
Founded	1998	2008	2009	2011
Edge compute	2019 EdgeWorkers	2016 Lambda@Edge & CF Functions (2021)	2017 CF Workers	2019 Compute@Edge

CDNs evolved too!

		 Amazon CloudFront		
Founded	1998	2008	2009	2011
Edge compute	2019 EdgeWorkers	2016 Lambda@Edge & CF Functions (2021)	2017 CF Workers	2019 Compute@Edge
Data services	EdgeKV, Cloudlets, Image/Video Manager	KeyValueStore	Workers KV, Queues, D1, R2	KV/Secret/Config store, Fanout

CDNs evolved too!

	Akamai	Amazon CloudFront	CLOUDFLARE	fastly
Founded	1998	2008	2009	2011
Edge compute	2019 EdgeWorkers	2016 Lambda@Edge & CF Functions (2021)	2017 CF Workers	2019 Compute@Edge
Data services	EdgeKV, Cloudlets, Image/Video Manager	KeyValueStore	Workers KV, Queues, D1, R2	KV/Secret/Config store, Fanout
Wasm support	❌ --no-expose-wasm	Not officially… but somebody succeeded!	2018 (Wasm) 2022 (Wasi)	Built-in since launch

A closer look

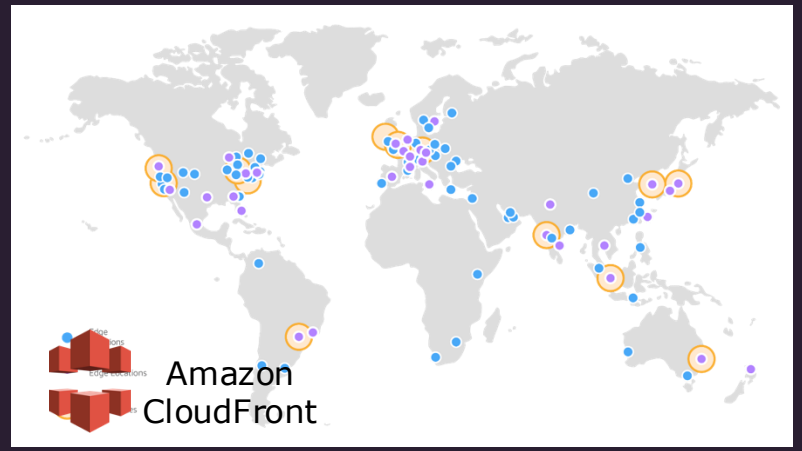
Wasm @ Cloudflare Workers:

- Beta support for Python (via Pyodide)
  - Optimized imports with linear memory snapshots
  - Bindings exposed via `JsProxy`
- Beta support for Rust (`workers-rs` crate)
  - `wasm32-unknown-unknown` target

Wasm @ Fastly Compute:

- Official SDKs for JS/TS, Rust, and Go
- Unofficial SDKs for Zig, Swift, Ruby, and .NET
  - Public `.witx` definitions to build your own SDK and package Wasm binaries
- Only `wasip1`

Globally distributed by default



Resource and performance constraints

On Fastly Compute:

- Max CPU time: 50ms
- Max memory: 128MB
- Max # of lookups: 16
- Max # of backend reqs: 32

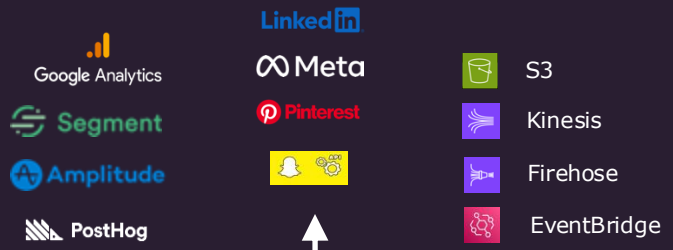
On Cloudflare Workers:

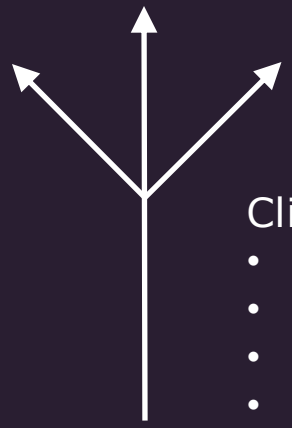
- Max CPU time: 10ms (free), 30s (paid)
- Max memory: 128MB
- Max # of lookups: 50 (free), 1000 (paid)
- Max # of sub-reqs: 50 (free), 1000 (paid)

(Well-known) Wasm benefits

- Cross-platform portability
- Near-native performance
- Lightweight & fast initialization
- Secure sandboxing
- Multi-language support & versatility

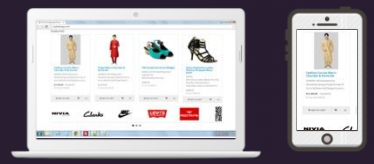
Wasm @ Edge(e)





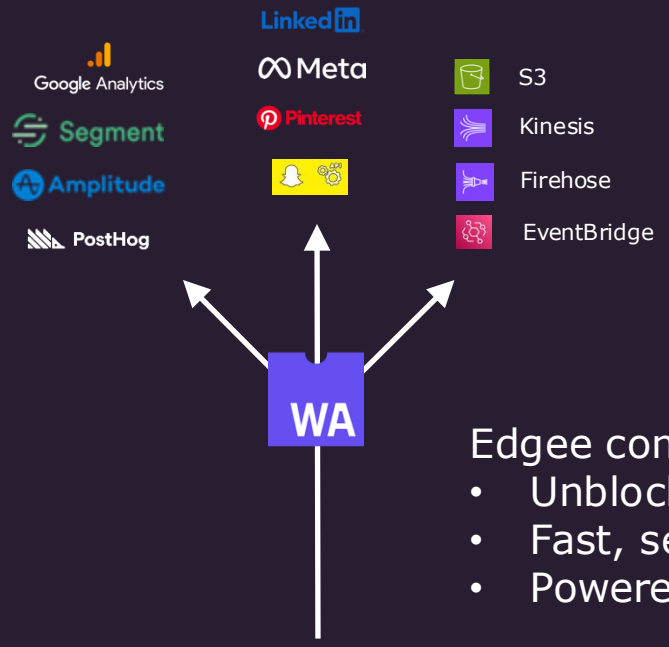
Client-side SDKs:

- Slow down performance
- Increased security risk
- Privacy compliance issues
- Lead to inaccurate analytics



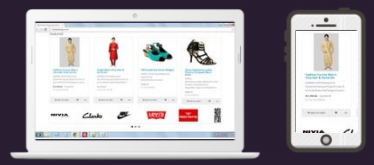
Client device

Cloud / website



Edgee components:

- Unblockable
- Fast, secure, compliant
- Powered by Wasm



edgee

Client device

Cloud / website

Vision: empower developers to build the next generation of high-performance, sustainable applications.

Mission: create an ecosystem of open source, interoperable components that can run where they're most effective.

Challenges:

- Run business logic efficiently after each HTTP request
- Support untrusted code (3rd-party components)
- Define domain-specific interfaces for many use cases
- Handle versioning of components and WIT interfaces

Meet the Edgee Component Registry

edgee.cloud/registry

Categories

- All
- Analytics
- Consent Mapping
- Conversion API
- Warehouse
- Native Cookies

Analytics

View all (6)

- edgee/amplitude — analytics — 1.0.0
- eulerian-technologies/eulerian-analytics — analytics — 1.0.9
- edgee/google-analytics — analytics — 1.0.0
- edgee/piano-analytics — analytics — 1.0.0

Consent Mapping

View all (3)

- edgee/axeptio-consent-mapping — consent mapping — 1.0.0
- edgee/didomi-consent-mapping — consent mapping — 1.0.0
- edgee/onetrust-consent-mapping — consent mapping — 1.0.0

Conversion API

View all (4)

- edgee/linkedin-capi — conversion api — 1.0.0
- edgee/meta-capi — conversion api — 1.0.0
- edgee/pinterest-capi — conversion api — 1.0.0
- edgee/snapchat-capi — conversion api — 1.0.0

Warehouse

View all (3)

- edgee/amazon-data-firehose — warehouse — 1.1.0
- edgee/amazon-kinesis — warehouse — 1.1.0
- edgee/amazon-s3 — warehouse — 1.2.0

Native Cookies

View all (3)

- edgee/amplitude-native-cookies
- edgee/google-analytics-native-cookies
- edgee/piano-analytics-native-cookies

Many more components and types coming soon

(CMP, A/B testing, security, AI inference, and more!)

roadmap.edgee.cloud/roadmap/data-collection-components

Technical overview

- Edge reverse proxy implemented in Rust
  - Domain-specific layer on top of Wasmtime
  - Shared codebase (proxy & CLI) /edgee-cloud/edgee

Technical overview

- Edge reverse proxy implemented in Rust
  - Domain-specific layer on top of Wasmtime
  - Shared codebase (proxy & CLI) /edgee-cloud/edgee
- Wasip2 & WIT
  - `package edgee:components` /edgee-cloud/edgee-wit

Technical overview

- Edge reverse proxy implemented in Rust
  - Domain-specific layer on top of Wasmtime
  - Shared codebase (proxy & CLI) /edgee-cloud/edgee
- Wasip2 & WIT
  - `package edgee:components` /edgee-cloud/edgee-wit
- Supported languages
  - Rust, Go, C, C#, Python, JavaScript, TypeScript
  - Templates: /edgee-cloud/example-{LANG}-component

Technical overview

- Edge reverse proxy implemented in Rust
  - Domain-specific layer on top of Wasmtime
  - Shared codebase (proxy & CLI) /edgee-cloud/edgee
- Wasip2 & WIT
  - `package edgee:components` /edgee-cloud/edgee-wit
- Supported languages
  - Rust, Go, C, C#, Python, JavaScript, TypeScript
  - Templates: /edgee-cloud/example-{LANG}-component
- Fully integrated with the Edgee managed service

```
package edgee:native;

world data-collection {
  export edgee:components/data-collection;
}
```

```
package edgee:components;

interface data-collection {
    type dict = list<tuple<string,string>>;
    enum event-type { page, track, user }
    enum consent { pending, granted, denied }
    enum http-method { GET, PUT, POST, DELETE }
    record event {...}

    variant data {}
    record page-data {...}
    record user-data {...}
    record track-data {...}

    record context {...}
    record client {...}
    record campaign {...}
    record session {...}

    record edgee-request {
        method: http-method,
        url: string,
        headers: dict,
        forward-client-headers: bool,
        body: string,
    }

    page:  func(e: event, settings: dict) -> result<edgee-request, string>;
    track: func(e: event, settings: dict) -> result<edgee-request, string>;
    user:  func(e: event, settings: dict) -> result<edgee-request, string>;
}
```

Meet the Edgee CLI

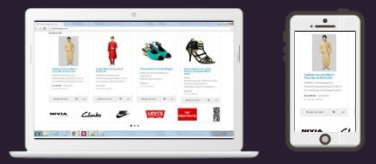
```
$ curl https://install.edgee.cloud | sh
```

```
$ brew tap edgee-cloud/edgee
$ brew install edgee
```

```
$ cargo binstall edgee
```

github.com/edgee-cloud/edgee

```
> edgee component build
 INFO Running: npm install && npm run generate && npm run build

added 122 packages, and audited 123 packages in 572ms

31 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

> generate
> npx @bytecodealliance/jco types .edgee/wit -o types/

  Generated Type Files:

 - types/interfaces/edgee-components-data-collection.d.ts  2.59 KiB
 - types/wit.d.ts                                          0.16 KiB
```

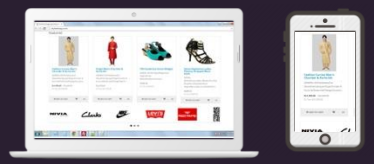


fastly

Edgee proxy

Client device

Yourwebsite.com



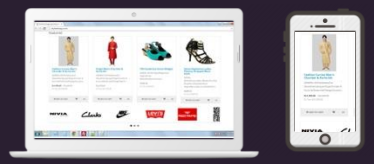
Client device

fastly Edgee proxy

Yourwebsite.com

aws event.edgee.app

WA WA WA
WA WA WA



Client device

fastly Edgee proxy

Yourwebsite.com

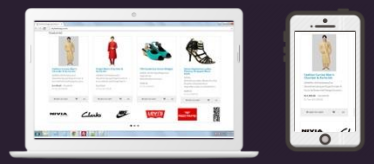
aws event.edgee.app

eu-west-1

eu-west-2

us-east-1

...



Client device

fastly Edgee proxy

Yourwebsite.com

aws api.edgee.app

edgee.cloud

Edgee CLI

Other API consumers

aws event.edgee.app

eu-west-1

eu-west-2

us-east-1

...

Takeaways & next steps

- CDNs aren’t just for caching & DDoS protection anymore
- Edge computing is a new way of building new apps & features
- Wasm + wasip2 at the edge still limited, but promising
- We wish WIT had built-in versioning capabilities (host)

My call to actions for you:

- Join our open community
- Check out the Edgee Component Registry

Thank you!

Alex Casalboni

Developer Advocate, Edgee

linkedin.com/in/alexcasalboni