From Cloud to Edge Computing

Unleashing the power of WebAssembly at the edge

Alex Casalboni

Developer Advocate, Edgee
edgee.cloud

Agenda for today 📆

- What is edge computing
- Quick history of cloud & edge
- Intro to server-side WebAssembly
- Wasm + edge = 💔
- Use case: Edgee

cloud academy
A QA COMPANY

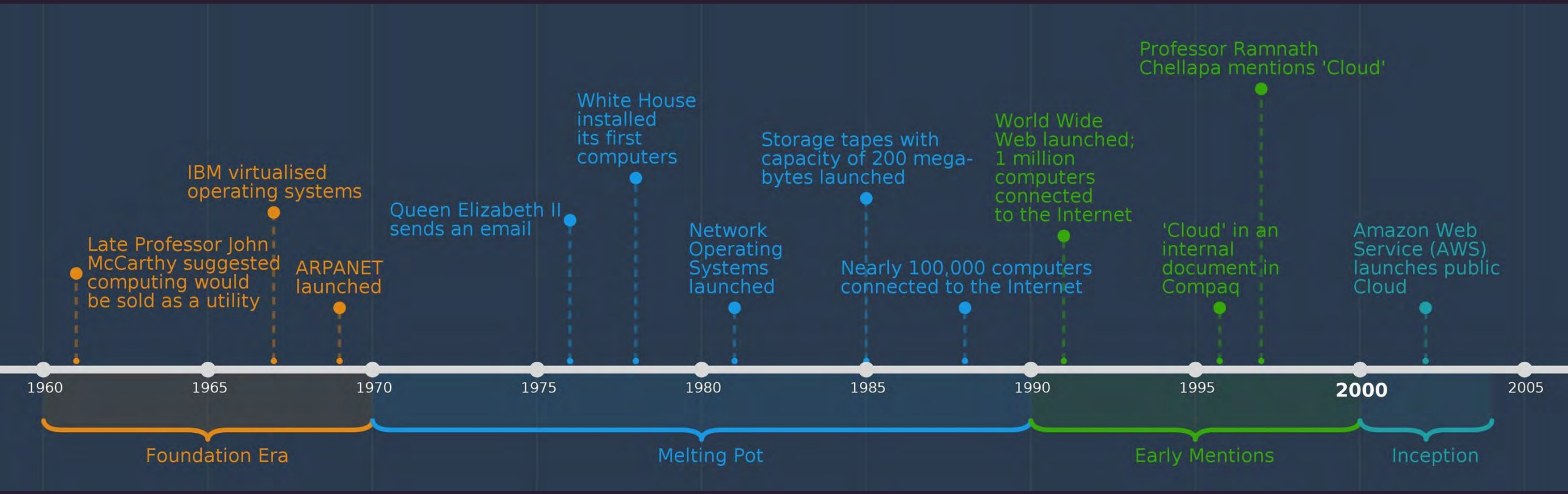
aws

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

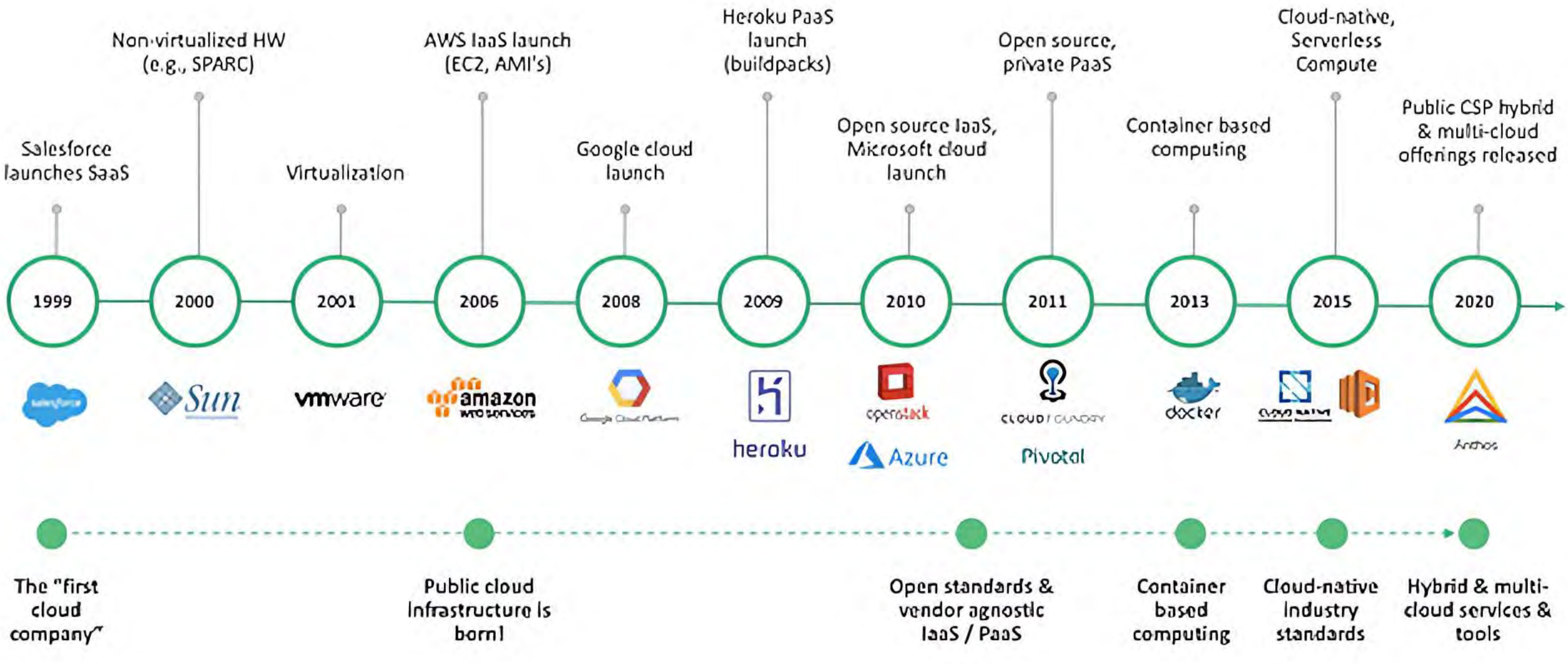
So what is exactly edge computing?

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
Other services	EdgeKV, Cloudlets, Image/Video Manager	CF KeyValueStore	Workers KV, Queues, D1, R2	KV/Secret/Config store, Fanout

Meet server-side WebAssembly

WEBASSEMBLY

Java, Scala, Kotlin, Python, Ruby, JavaScript, TypeScript, PHP, C#, C++, C, Go, Rust, ...

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)...
> Show more

9:39 PM · Mar 27, 2019

30 1K 2.2K 249

webassembly.org
github.com/WebAssembly/design

wasi.dev
github.com/WebAssembly/WASI

github.com/WebAssembly/component-model

Key benefits

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

```go
package main

import "fmt"

func main() {
    fmt.Println("Hello from WebAssembly!")
}

```

```
$ GOOS=js GOARCH=wasm go build -o main.wasm
```

```
$ cp "$(go env GOROOT)/lib/wasm/wasm_exec.js" .
```

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Wasm test in go</title>
        <meta charset="utf-8"/>
        <script src="wasm_exec.js"></script>
        <script>
            const go = new Go();
            WebAssembly.instantiateStreaming(fetch("main.wasm"), go.importObject)
                .then((result) => {
                    go.run(result.instance);
                });
        </script>
    </head>
    <body></body>
</html>

```

go.dev/wiki/WebAssembly

```
$ GOOS=wasip1 GOARCH=wasm go build -o main.wasm
```

```
$ GOOS=wasip2 GOARCH=wasm tinygo build -o main.wasm
```

```
$ wit-bindgen-go generate -o internal/ ./wit
```

tinygo.org/docs/guides/webassembly/wasi/

API	Repository
Clocks	https://github.com/WebAssembly/wasi-clocks
Random	https://github.com/WebAssembly/wasi-random
Filesystem	https://github.com/WebAssembly/wasi-filesystem
Sockets	https://github.com/WebAssembly/wasi-sockets
CLI	https://github.com/WebAssembly/wasi-cli
HTTP	https://github.com/WebAssembly/wasi-http

wasi.dev/interfaces

Application Development

Announcing Wasm support in Go 1.24

February 14, 2025

Cameron Balahan
Group Product Manager

Arman Rye
Product Manager

Earlier this week, we released Go 1.24, the latest version of Google's open-source programming language for productively building scalable, production-ready backend and cloud-based systems.

There's a lot to love about Go 1.24, including support for post-quantum cryptography, a weak pointer implementation, and substantial performance improvements to the Go runtime. Go 1.24 also significantly expands its capabilities for WebAssembly (Wasm), a binary instruction format that provides for the execution of high-performance, low-level code at speeds approaching native performance. With a new `go:wasmexport` compiler directive and the ability to build a reactor for the WebAssembly System Interface (WASI), developers can now export functions from their Go code to Wasm — including in long-running applications — fostering deeper integrations with Wasm hosts and unlocking new possibilities for Go-based Wasm applications.

These additions represent a significant step forward in Go's Wasm story. For some types of applications, like those running at the edge, Wasm is critical to serving performance-critical use cases. Now, developers can leverage Go's signature capabilities to ensure that these use cases are also scalable, secure, and production-ready.

cloud.google.com/blog/products/application-development/go-1-24-expands-support-for-wasm

Application Development

Announcing Wasm support in Go 1.24

February 14, 2025

Cameron Balahan
Group Product Manager

Arman Rye
Product Manager

Run Wasm at the edge with Google Cloud

Starting today, you can now run Go compiled Wasm plugins for applications built on Google Cloud at the edge. To do so, you need to leverage Service Extensions with Google Cloud's Application Load Balancers. Service Extensions allows you to run your own custom code directly in the request/response path in a fully managed Google environment with optimal latency, so you can customize load balancers to meet your business requirements. All you need to do is provide the code — Google Cloud manages the rest.

To get started with Service Extensions plugins and Go, take a look at our growing samples repository with a local testing toolkit and follow our quickstart guide in the documentation.

cloud.google.com/blog/products/application-development/go-1-24-expands-support-for-wasm

Wasmtime

A fast and secure runtime for WebAssembly

A Bytecode Alliance project

wasmtime.dev

Overview | Repositories 96 | Projects 12 | Packages | People 32

Bytecode Alliance

Unfollow

Verified

1.6k followers | https://bytecodealliance.org | @bytecodeallies | hello@bytecodealliance.org

Pinned

wasmtime Public
A lightweight WebAssembly runtime that is fast, secure, and standards-compliant
Rust 16k 1.4k

wasm-micro-runtime Public
WebAssembly Micro Runtime (WAMR)
C 5.2k 669

wasm-tools Public
CLI and Rust libraries for low-level manipulation of WebAssembly modules
Rust 1.4k 267

wit-bindgen Public
A language binding generator for WebAssembly interface types
Rust 1.1k 210

People

View all

Top languages

Rust C JavaScript CSS HTML

github.com/bytecodealliance

Members


amazon


ANACONDA.


arm


intel


JAF Labs


Microsoft


midokura
A Sony Group Company


moz://a


NGINX
Part of F5


cosmonic


DFINITY


docker


nor2


shopify


SIEMENS


fastly


FERMYON


FUTUREWEI
Technologies


SingleStore


StackBlitz


Stellar


InfinyOn


igalia

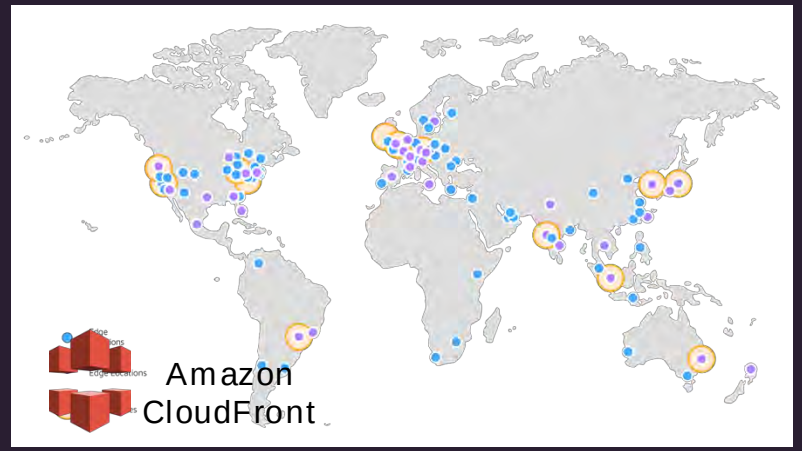

imec


UC San Diego


SNT

Wasm + Edge = 

Globally distributed by default



More resource and performance constraints

On Fastly Compute:

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

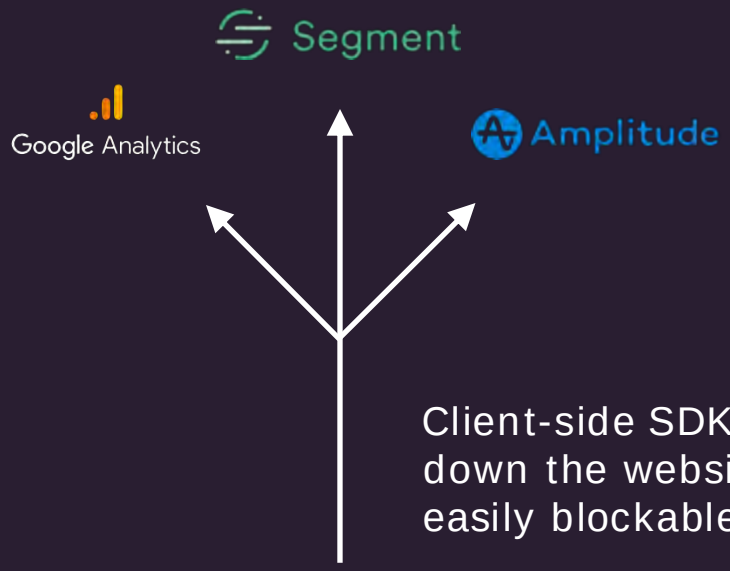
On CloudFlare Workers:

- 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)

Key benefits

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

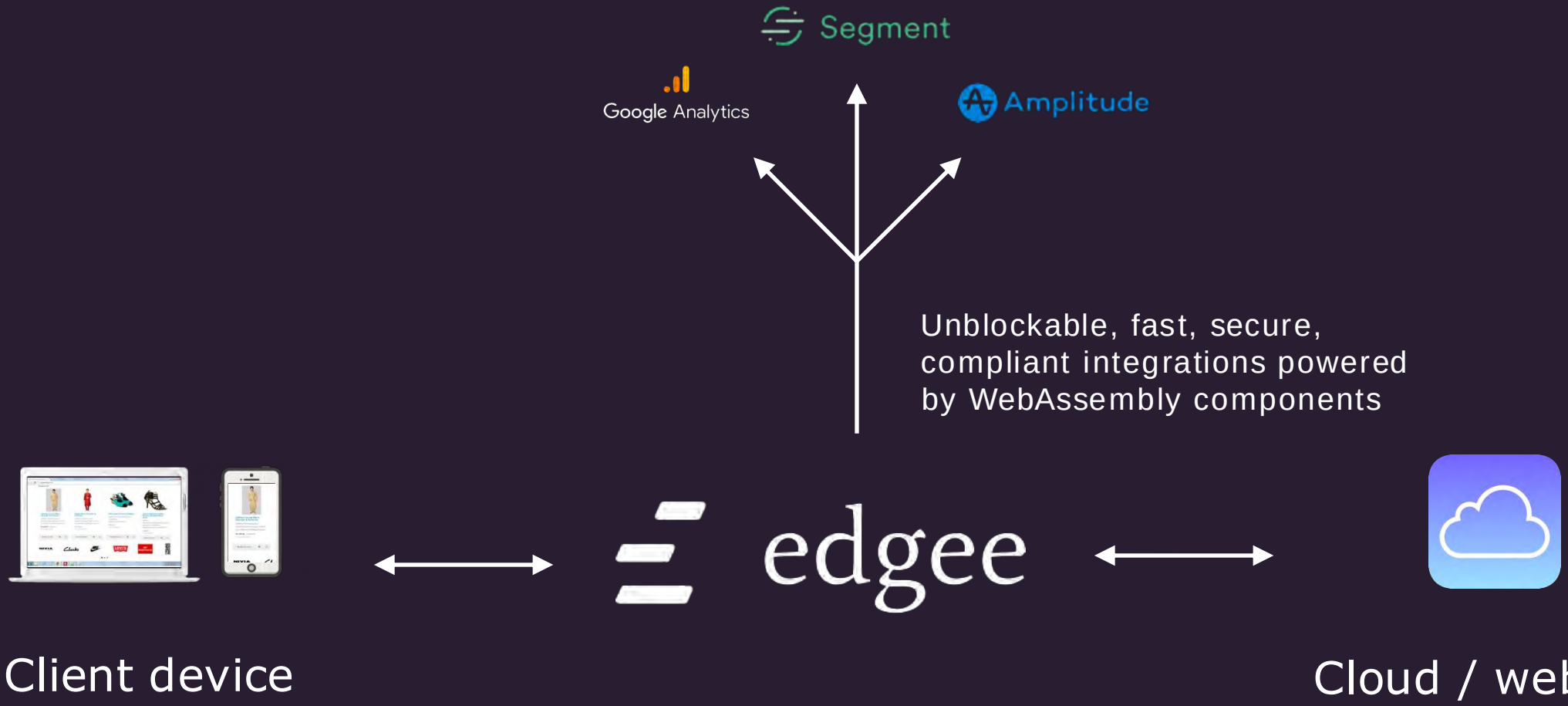
Meet  edgee



Client-side SDKs, slowing down the website and easily blockable

Client device

Cloud / website


Segment
Google Analytics
Amplitude
Unblockable, fast, secure,
compliant integrations powered
by WebAssembly components
edgee
Client device
Cloud / website

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

Meet the Edgee CLI

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

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

github.com/edgee-cloud/edgee

Client device

fastly

CLOUDFLARE

Edgee proxy

Yourwebsite.com

edgee.cloud

Edgee CLI

Other API consumers

aws event.edgee.app

WA WA WA WA WA WA

GO aws api.edgee.app

Takeaways

- CDNs aren’t just for caching & DDoS protection anymore
- The edge provides a new way of building global apps/features
- WebAssembly + WASI have a bright future
- Join our community and push to the Edgee Component Registry

Thank you!

Alex Casalboni

Developer Advocate, Edgee
linkedin.com/in/alexcasalboni
linkedin.com/company/edgee-cloud