

EJB Developer?

...dann aber auch Serverless Lambda Experte



airhacks.industries

**"It's not work if you like it"
...so I never worked. #java**



#185 A Cloud Migration Story: From J2EE to Serverless Java

[episode link] Listen on Apple Podcasts

LISTEN ON Spotify

Listen on Google Podcasts

[RSS]

An airhacks.fm

ZX Spectru

CPC 64, De

in 1993, usi

Lambda, Cl

clouds thei

services, n

quarkus in

the cloud h

#219 Java, CraC and Reducing Cold Start Duration with AWS Lambda SnapStart

[episode link] Listen on Apple Podcasts

LISTEN ON Spotify

Listen on Google Podcasts

[RSS]

#232 Kubernetes Was Never Supposed To Leak

[episode link] Listen on Apple Podcasts

LISTEN ON Spotify

Listen on Google Podcasts

[RSS]

An airhacks.fm conversation with Kelsey Hightower (@kelseyhightower) about:

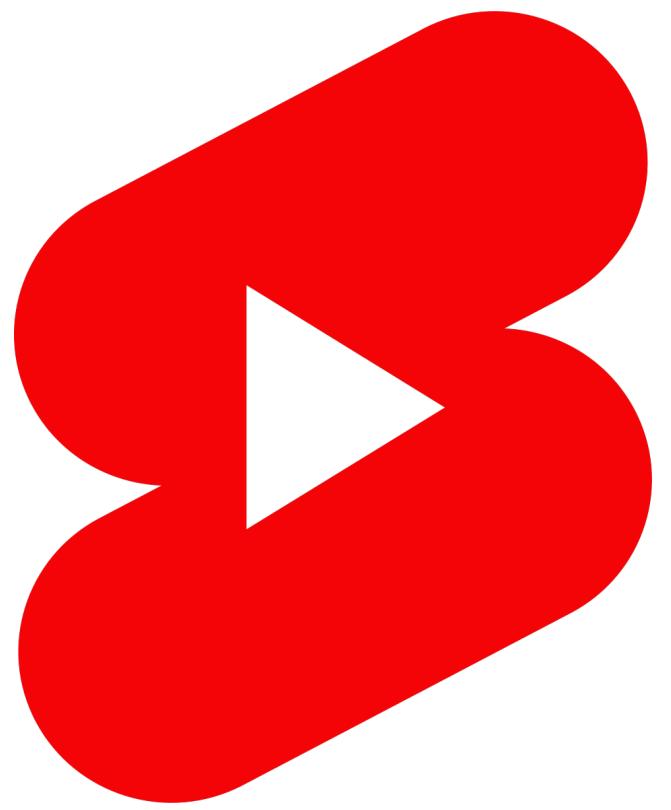
airhacks.live

HP laptop and playing Age of Empires, programming calculators with TI-BASIC, playing Metroid on NES, working at Google datacenter as contractor, bash is a pro

airhacks.TV

with the time machine, “100 episodes ago segment”

...any questions left?



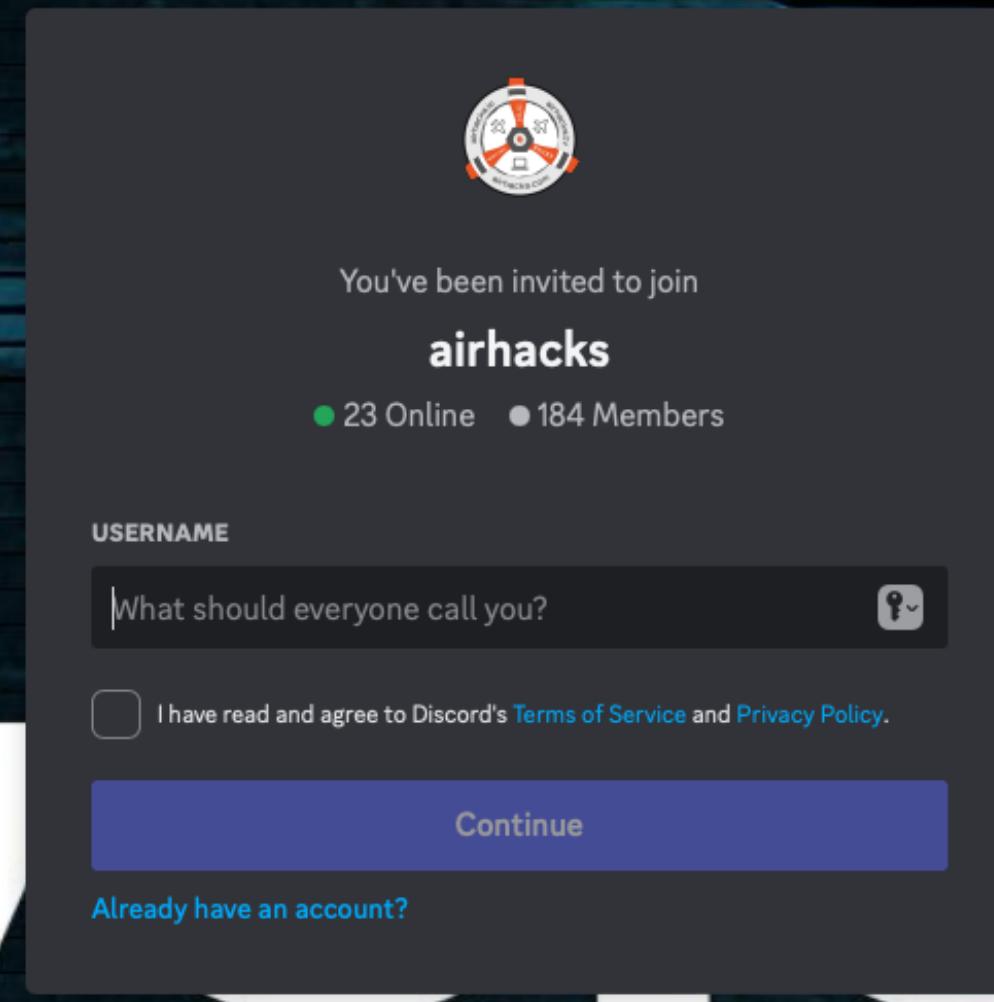
[youtube.com/
@bienadam/shorts](https://youtube.com/@bienadam/shorts)



youtube.com/bienadam/shorts

welcome to airhacks

airhacks.live



NEW <https://discord.gg/airhacks>

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Summer 2024:

Persistence Patterns for Serverless Java on AWS, July, 11th, 2024

Serverless Generative AI with Java on AWS, July, 25th, 2024

Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by Adam Bien

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io

airhacks.live

...I started with DevOps in 1995

then continued with serverless computing in 2001 ...

clouds?



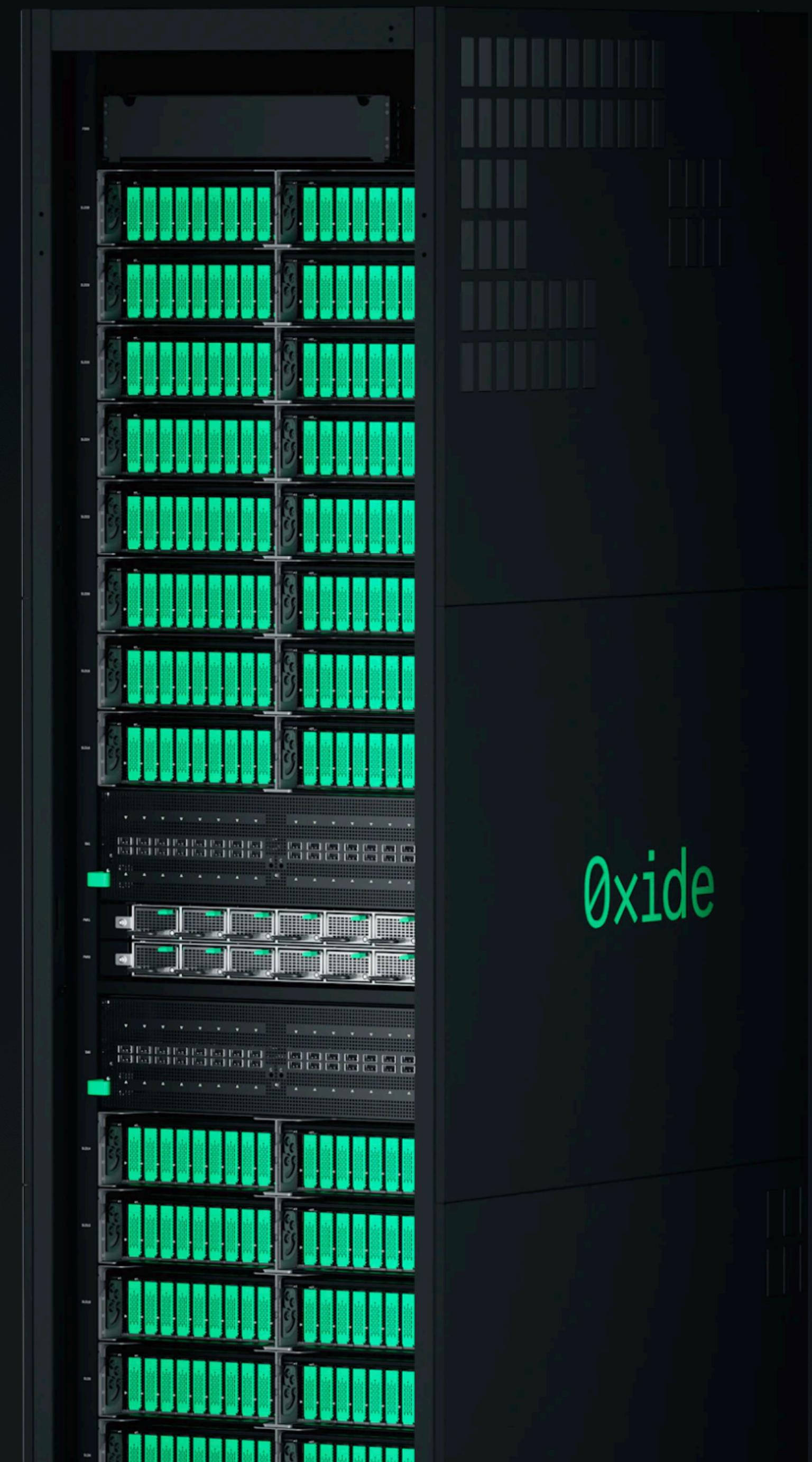
Sun Starfire 10000

Oxide Cloud Computer

No Cables. No Assembly.
Just Cloud.

CONTACT SALES

GET OUR NEWSLETTER



Oxide

Java && Clouds?

	Total		
	Energy	Time	Mb
(c) C	1.00	(c) C	1.00
(c) Rust	1.03	(c) Rust	1.04
(c) C++	1.34	(c) C++	1.56
(c) Ada	1.70	(c) Ada	1.85
(v) Java	1.98	(v) Java	1.89
(c) Pascal	2.14	(c) Chapel	2.14
(c) Chapel	2.18	(c) Go	2.83
(v) Lisp	2.27	(c) Pascal	3.02
(c) Ocaml	2.40	(c) Ocaml	3.09
(c) Fortran	2.52	(v) C#	3.14
(c) Swift	2.79	(v) Lisp	3.40
(c) Haskell	3.10	(c) Haskell	3.55
(v) C#	3.14	(c) Swift	4.20
(c) Go	3.23	(c) Fortran	4.20
(i) Dart	3.83	(v) F#	6.30
(v) F#	4.13	(i) JavaScript	6.52
(i) JavaScript	4.45	(i) Dart	6.67
(v) Racket	7.91	(v) Racket	11.27
(i) TypeScript	21.50	(i) Hack	26.99
(i) Hack	24.02	(i) PHP	27.64
(i) PHP	29.30	(v) Erlang	36.71
(v) Erlang	42.23	(i) Jruby	43.44
(i) Lua	45.98	(i) TypeScript	46.20
(i) Jruby	46.54	(i) Ruby	59.34
(i) Ruby	69.91	(i) Perl	65.79
(i) Python	75.88	(i) Python	71.90
(i) Perl	79.58	(i) Lua	82.91

GraalVM™
reduces RAM footprint

<https://sites.google.com/view/energy-efficiency-languages/results?authuser=0>

motivation / facts

Configure AWS Lambda [Info](#)



Architecture

Arm



Number of requests

2

Unit

per second



Duration of each request (in ms)

Duration is calculated from the time your code begins executing until it returns or otherwise terminates.

100

Amount of memory allocated

Enter the amount between 128 MB and 10 GB

Value

2

Unit

GB



Amount of ephemeral storage allocated

Enter the amount between 512 MB and 10,240 MB. The first 512 MB are at no additional charge, you only pay for any additional storage that you configure for the function.

Value

512

Unit

MB



► Show calculations

Provisioned Concurrency [Info](#)

Total Upfront cost: 0.00 USD

Total Monthly cost: 15.07 USD

Show Details ▾

Save and view summary

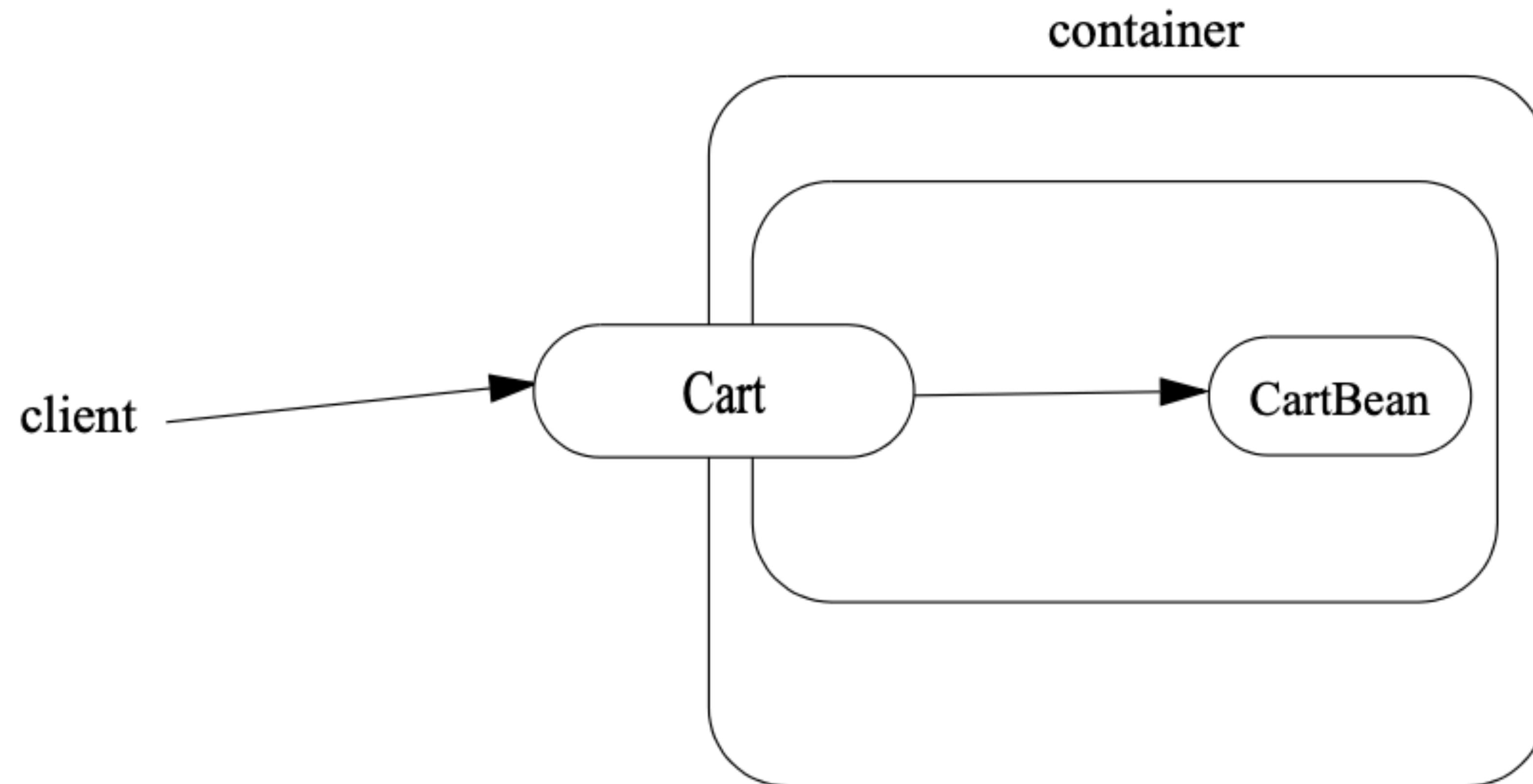
Save and add service

the great, lightweight EJBs :-)

EJBs

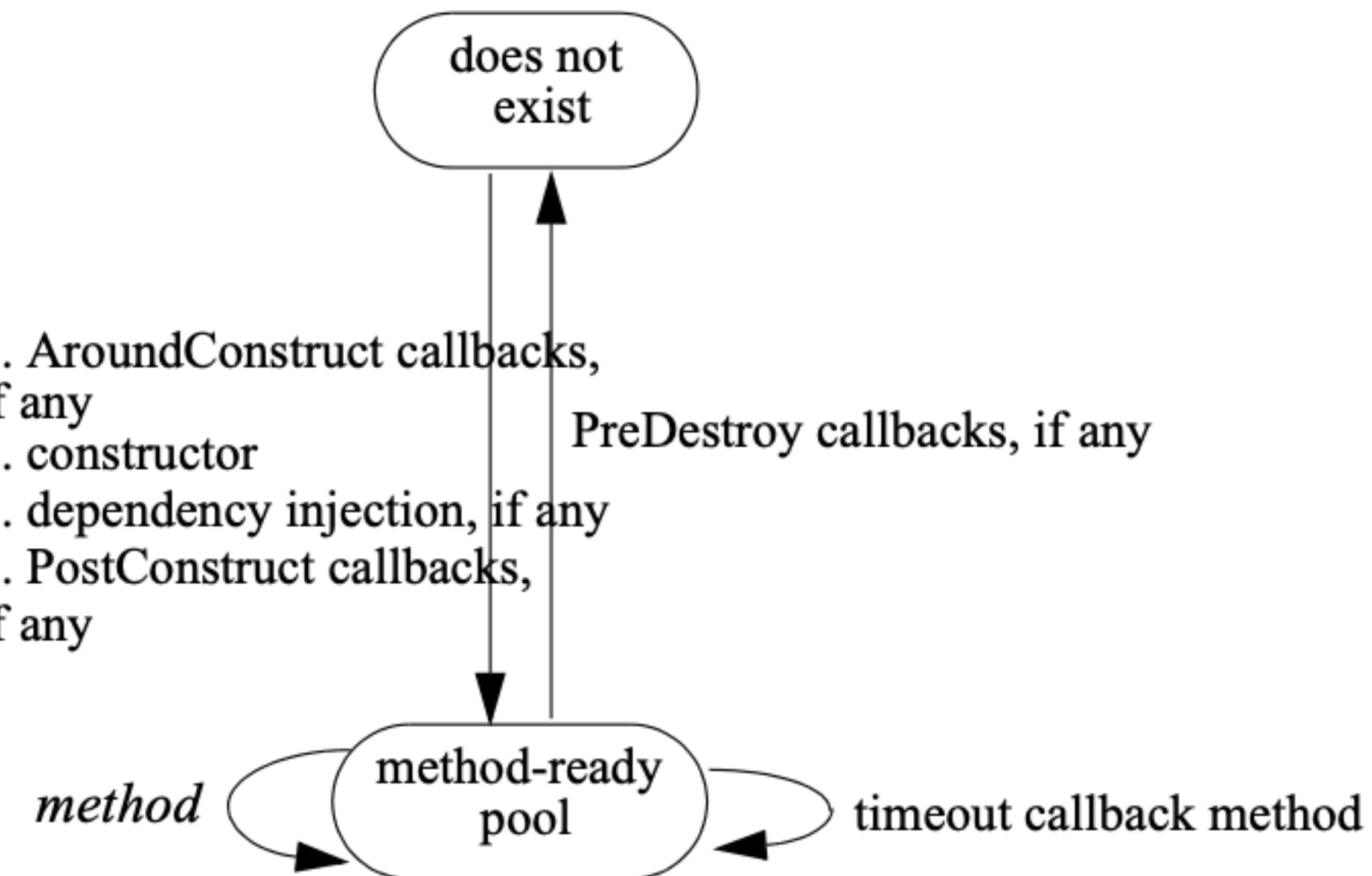
- lifecycle
- concurrency
- bulkheads
- state
- optimization patterns
- the meaning of “synchronized” and “static”
- packaging

Lifecycle



JSR 345: Enterprise JavaBeansTM, Version 3.2

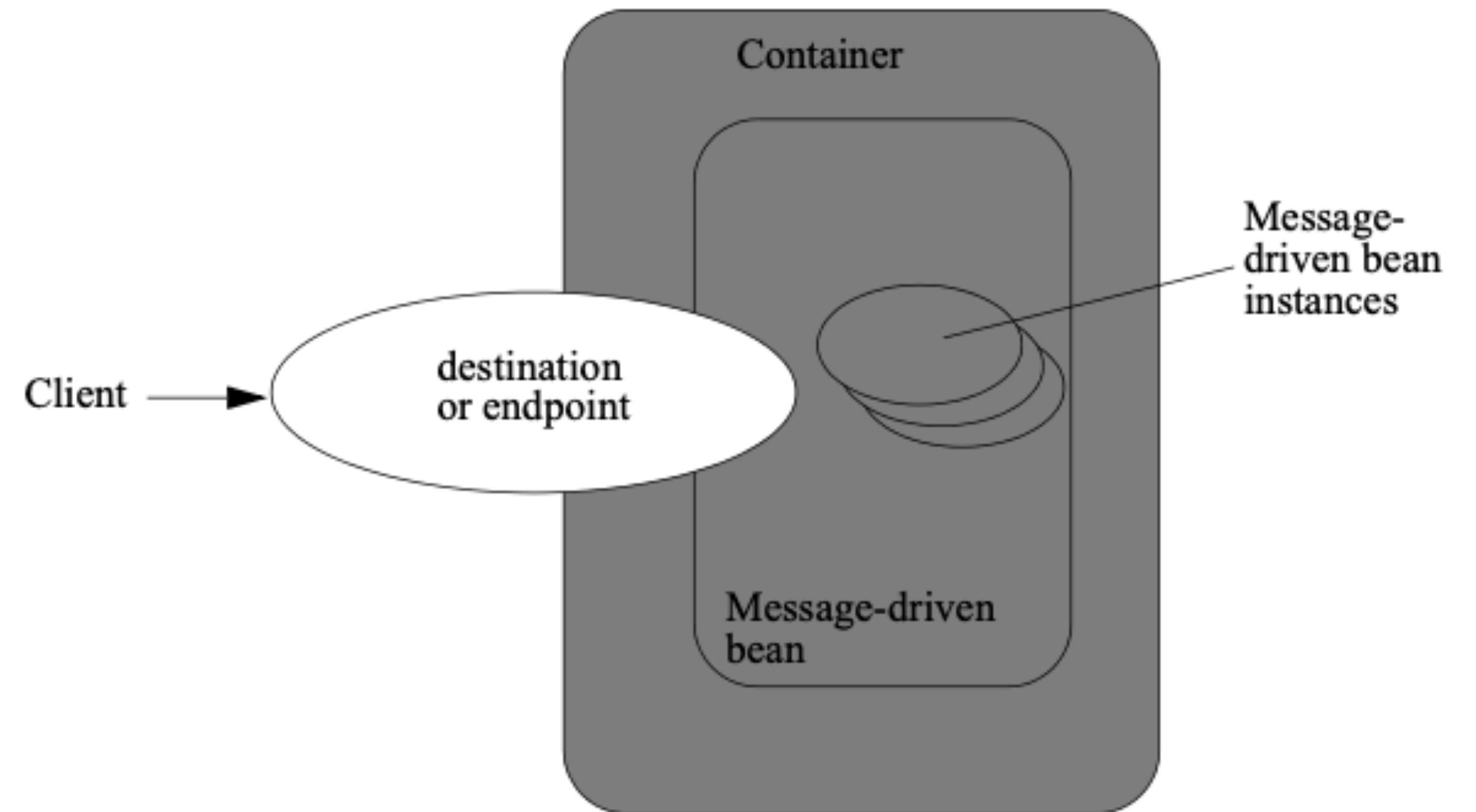
Stateless EJBs



JSR 345: Enterprise JavaBeansTM, Version 3.2

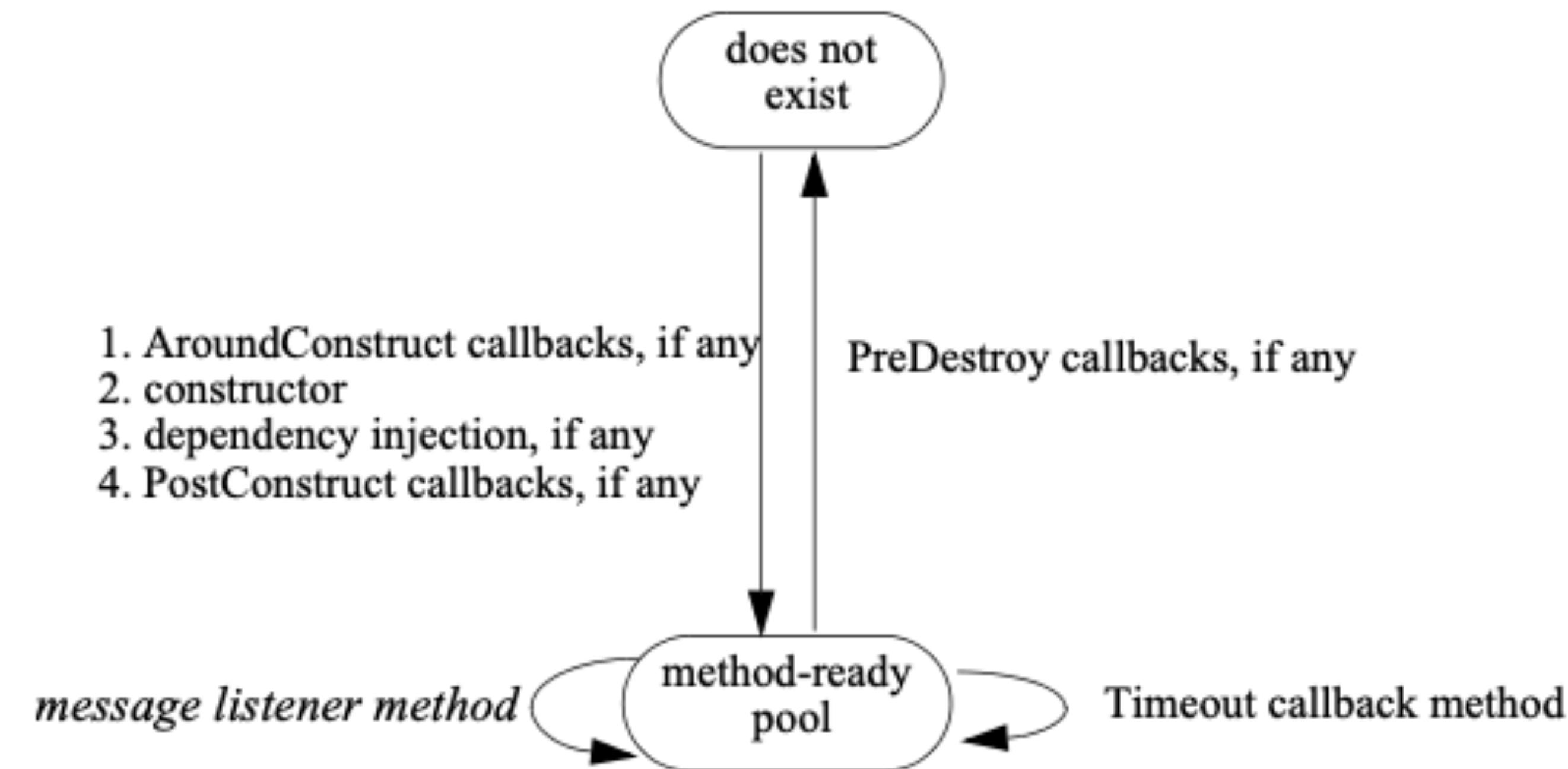
<i>method()</i> constructor	action initiated by client action initiated by container
--------------------------------	---

Message Driven Beans



JSR 345: Enterprise JavaBeansTM, Version 3.2

MDB Lifecycle



JSR 345: Enterprise JavaBeansTM, Version 3.2

EJB Programming Restrictions ↗ Lambda Best Practices

An enterprise bean must not use read/write static fields. Using read-only static fields is allowed. Therefore, it is recommended that all static fields in the enterprise bean class be declared as final.

An enterprise bean must not use thread synchronization primitives to synchronize execution of multiple instances, unless it is a singleton session bean with bean-managed concurrency.

An enterprise bean should exercise caution when using the Java I/O package to attempt to access files and directories in the file system.

An enterprise bean must not attempt to listen on a socket, accept connections on a socket, or use a socket for multicast.

(...) The enterprise bean must not attempt to use the Reflection API to access information that the security rules of the Java programming language make unavailable.

The enterprise bean must not attempt to create a class loader; set the context class loader; set security manager; create a new security manager; stop the JVM; or change the input, output, and error streams.

The enterprise bean must not attempt to set the socket factory used by ServerSocket, Socket, or the stream handler factory used by URL.

The enterprise bean must not attempt to manage threads. The enterprise bean must not attempt to start, stop, suspend, or resume a thread, or to change a thread's priority or name. The enterprise bean must not attempt to manage thread group.

...and the future of clouds is...

...how it should be

- no activity, no charges
- requests are proportional to costs
- Cost Driven Architecture (CDA :-))
- costs incur per transactions / request / invocation

Event-driven architecture (EDA) is a software architecture paradigm promoting the production, detection, consumption of, and reaction to events.

Serverless computing

Serverless computing is a [cloud computing execution model](#) in which the cloud provider allocates machine resources on demand, taking care of the [servers](#) on behalf of their customers. Serverless computing does not hold resources in volatile memory; computing is rather done in short bursts with the results persisted to storage. When an app is not in use, there are no computing resources allocated to the app. Pricing is based on the actual amount of resources consumed by an application.^[1] It can be a form of [utility computing](#). "Serverless" is a [misnomer](#) in the sense that servers are still used by cloud service providers to execute code for developers. However, developers of serverless applications are not concerned with [capacity planning](#), configuration, management, maintenance, [fault tolerance](#), or scaling of containers, [VMs](#), or physical servers.

https://en.wikipedia.org/wiki/Serverless_computing

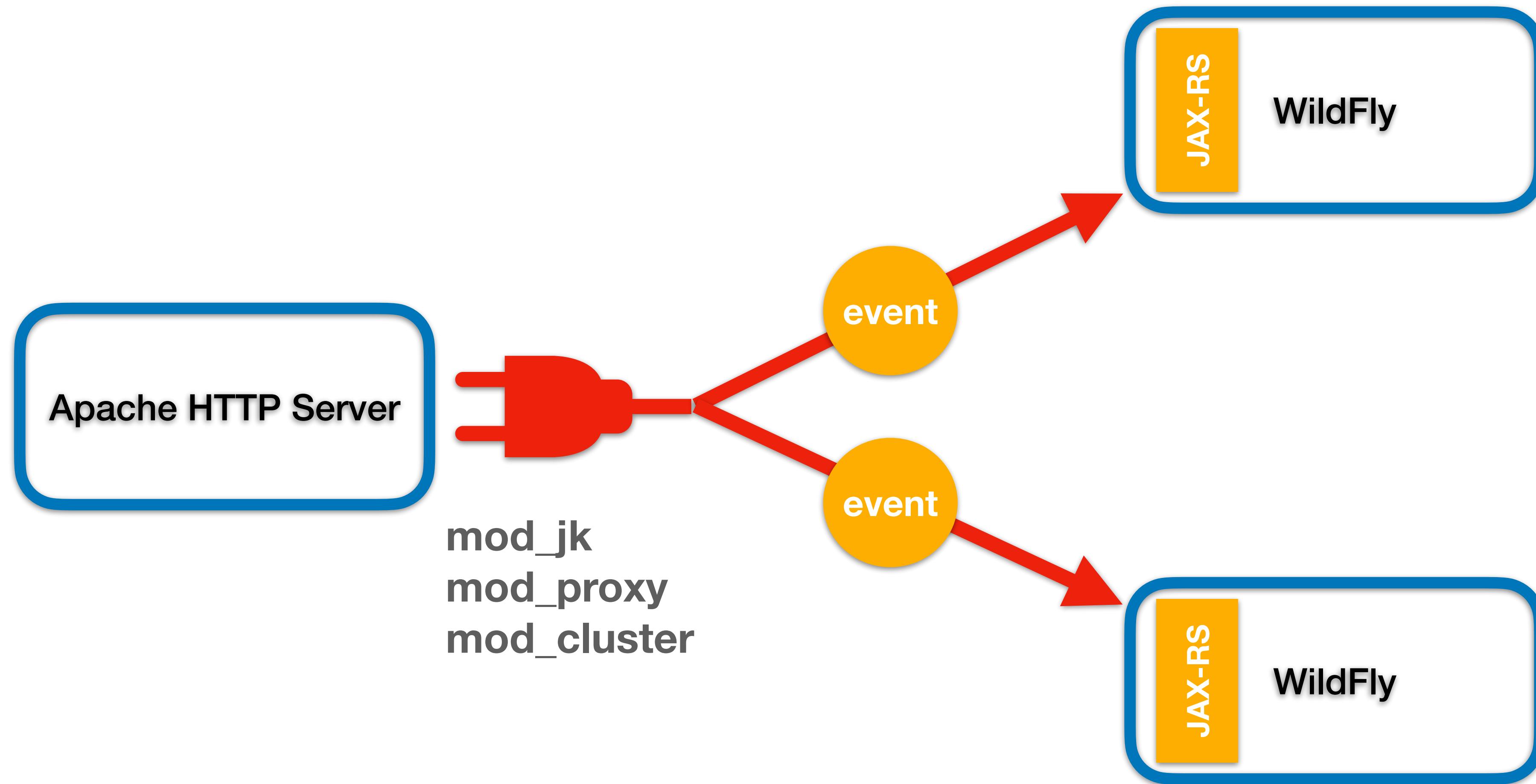
AWS Lambda

AWS Lambda is an [event-driven, serverless computing](#) platform provided by [Amazon](#) as a part of [Amazon Web Services](#). It is a computing service that runs code in response to [events](#) and automatically manages the computing resources required by that code. It was introduced in November 2014.^[1]

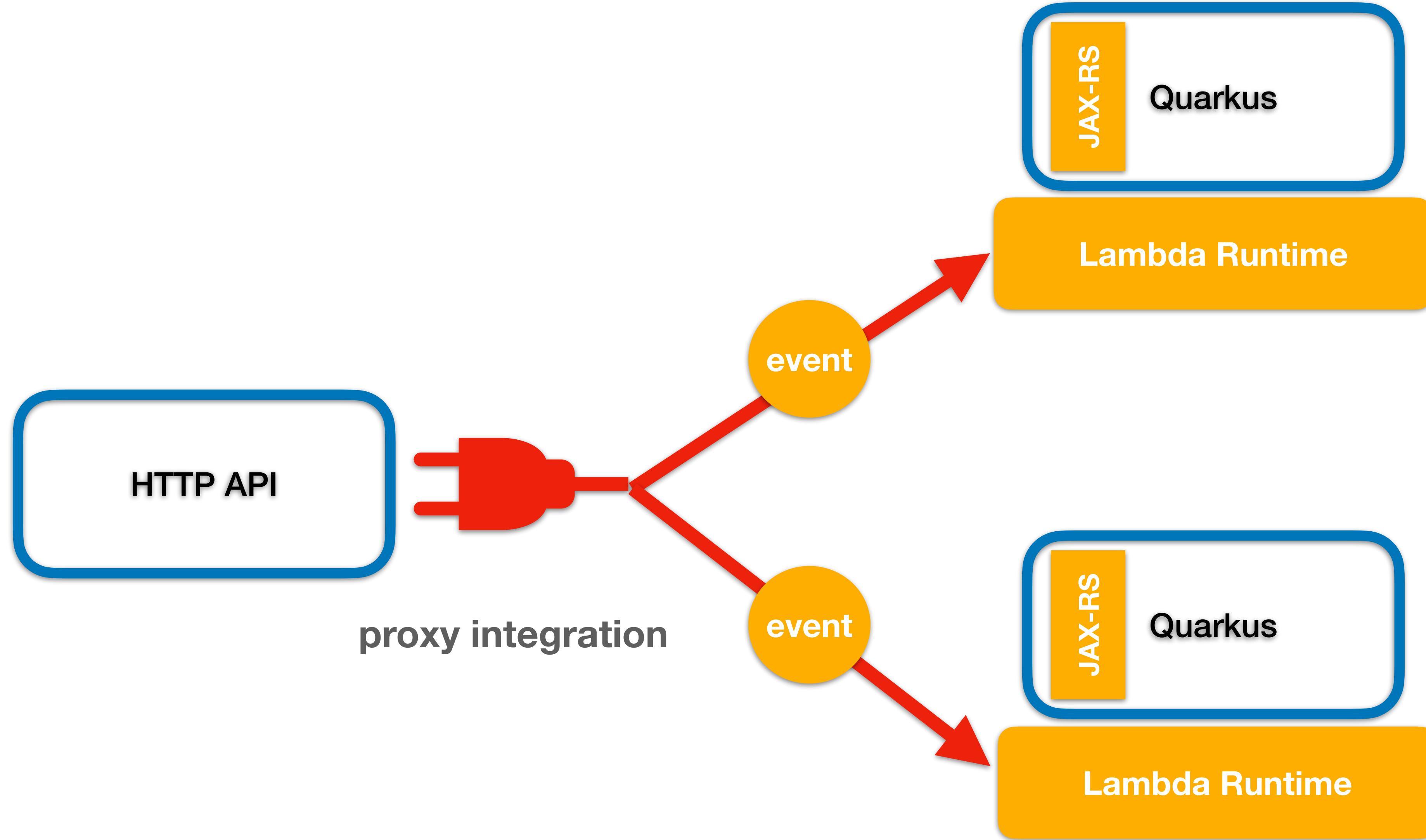
[Node.js](#), [Python](#), [Java](#), [Go](#),^[2] [Ruby](#),^[3] and [C#](#) (through [.NET](#)) are all officially supported as of 2018. In late 2018, custom runtime support^[4] was added to AWS Lambda.

https://en.wikipedia.org/wiki/Serverless_computing

EJBs and Events



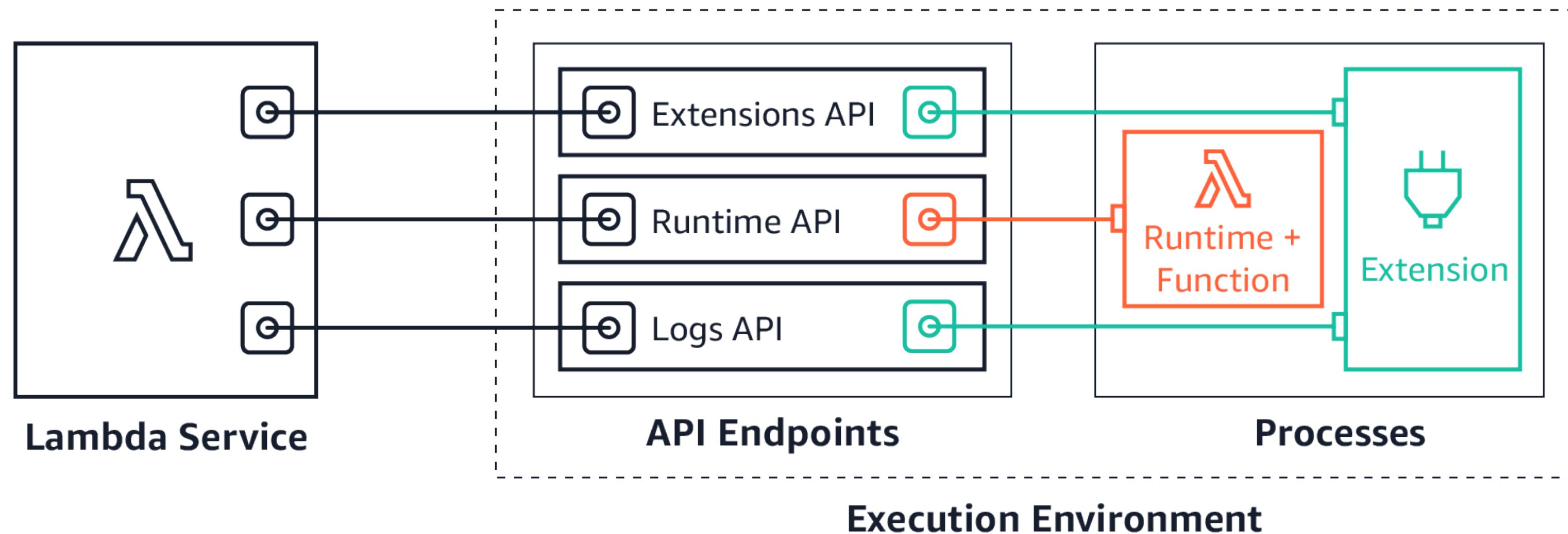
AWS Lambda



serverless for enterprise

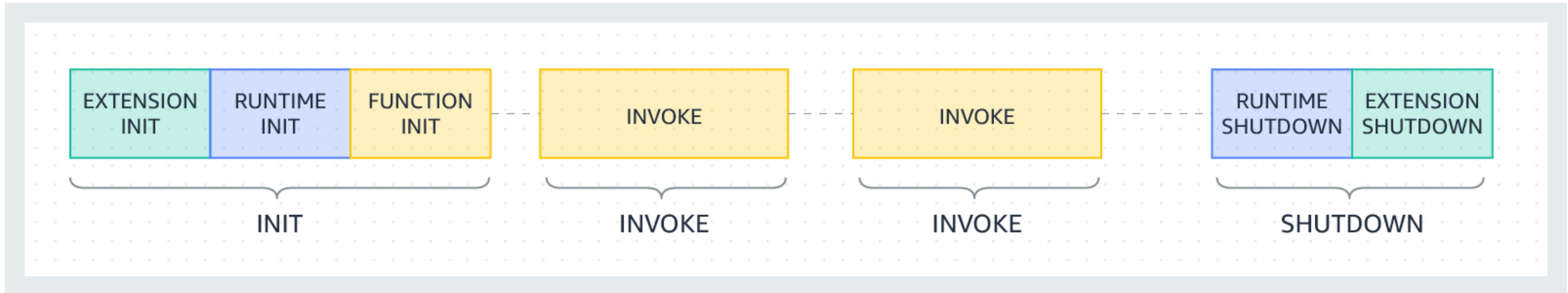
- better target tracking autoscaling
- precise, infinite autoscaling (1000 lambdas are the default)
- scale to zero - no traffic, no costs
- free staging
- AB deployment
- built-in rolling updates

Lambda Execution Environment



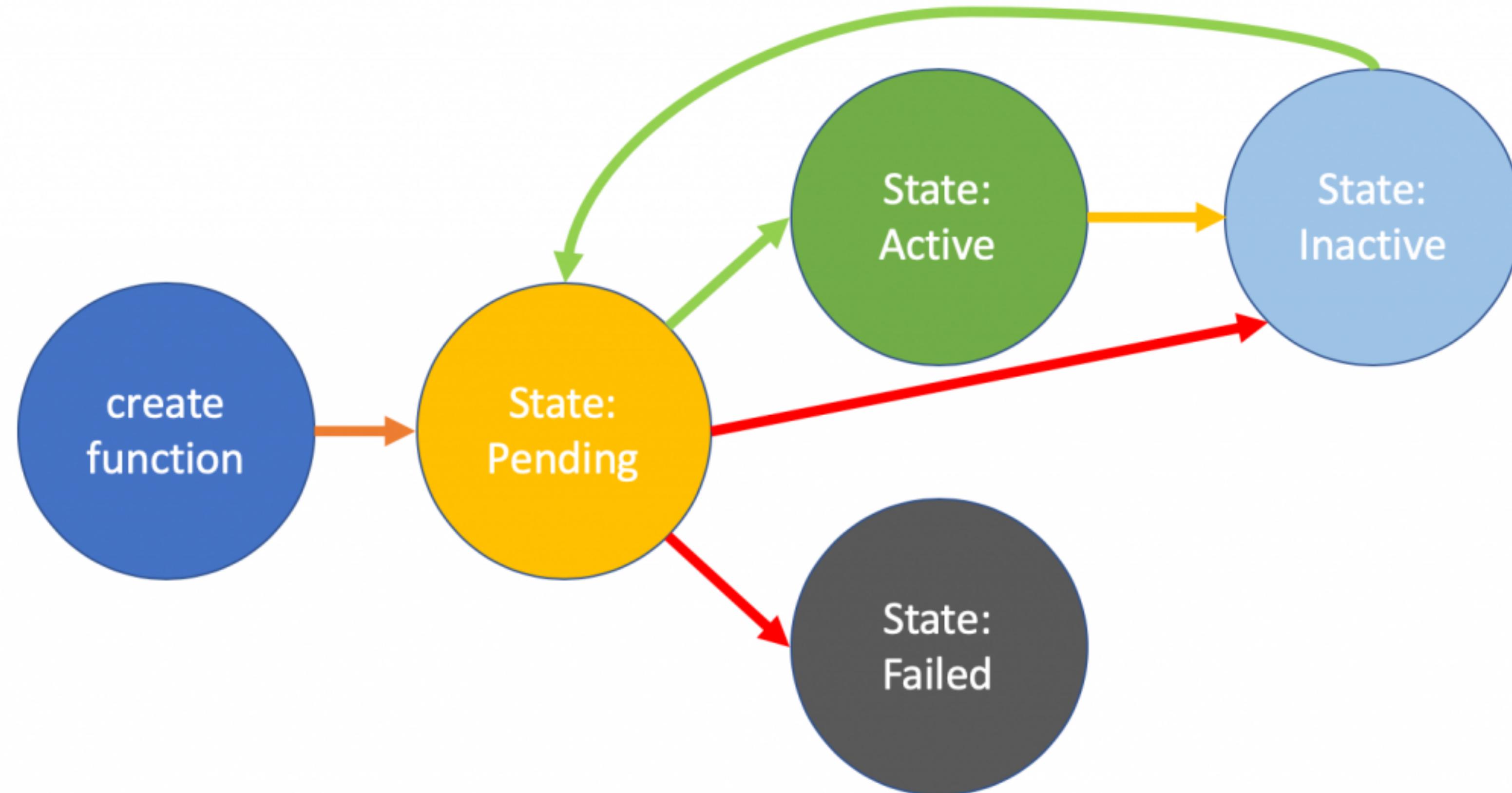
<https://docs.aws.amazon.com/lambda/latest/dg/runtimes-context.html>

Execution environment lifecycle



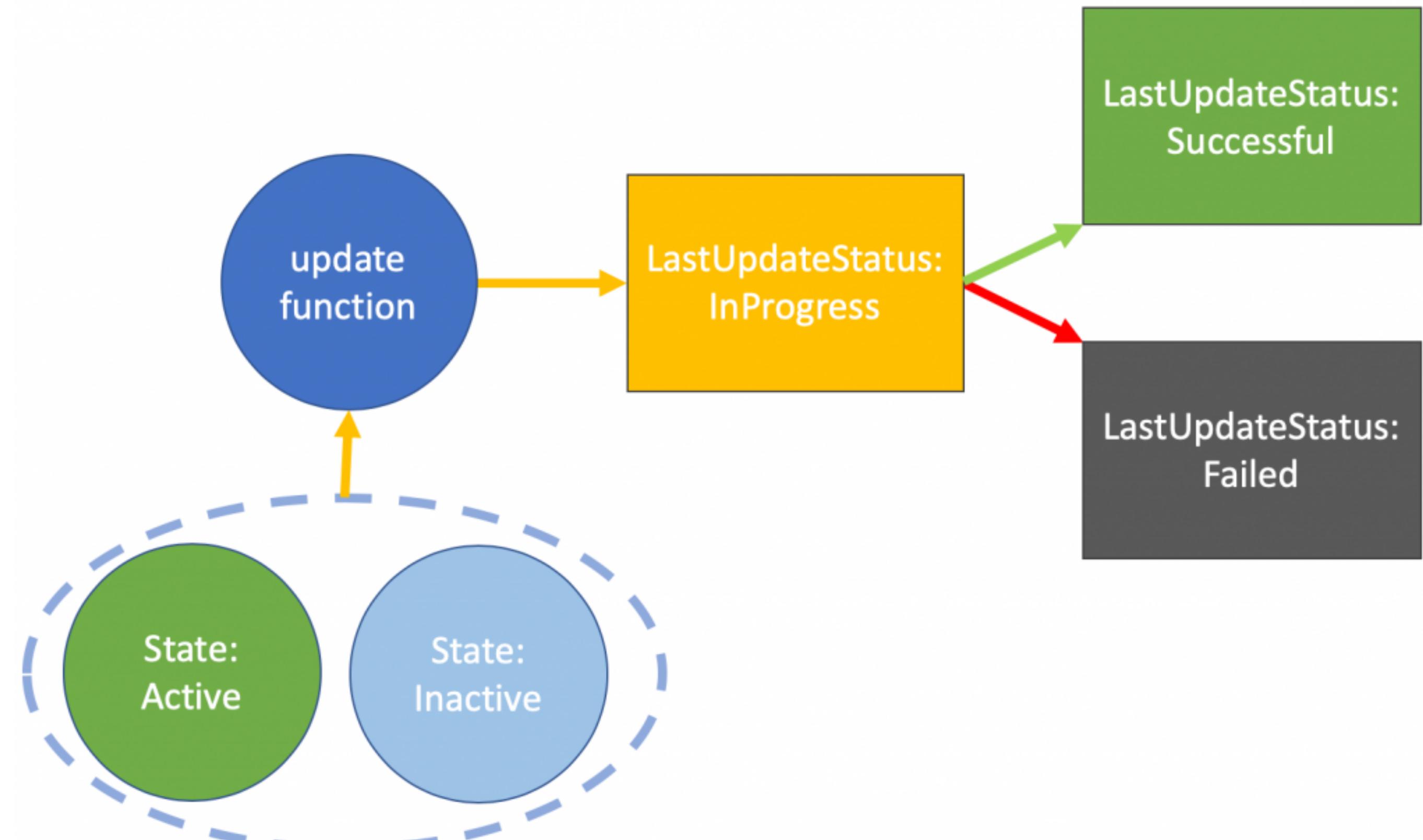
<https://docs.aws.amazon.com/lambda/latest/dg/runtimes-context.html>

Lambda States



<https://aws.amazon.com/blogs/compute/tracking-the-state-of-lambda-functions/>

Lambda States (Updates)



<https://aws.amazon.com/blogs/compute/tracking-the-state-of-lambda-functions/>

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Summer 2024:

Persistence Patterns for Serverless Java on AWS, July, 11th, 2024

Serverless Generative AI with Java on AWS, July, 25th, 2024

Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by Adam Bien

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io

airhacks.live



Thank YOU!



airhacks.industries