No-Compromise Cloud

🚀Enterprise Java Done Right

airhacks.industries

airhacks.live

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

youtube.com/
@bienadam

/@bienadam/shorts

airhacks.fm

#360 Java, LangChain4J and Enterprise LLMs

Listen on Apple Podcasts · Listen on Spotify · YouTube · [RSS]

An airha

journe
Micros
mode
assist
abstra
stand
Entity
integr
multi-
poten

air

#356 AI/LLM Driven Development

Listen on Apple Podcasts · Listen on Spotify · YouTube · [RSS]

An ai

bro
per
inte
arc
bro
dev
sea

#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) abo

HP laptop and playing Age of Empires, programming calculators with TI-

airhacks.TV

with the time machine, “100 episodes ago segment”

…any questions left?

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Winter 2025:

FULLY BOOKED The 50x Java Dev: Crafting Elegant Code with LLMs, 16 December 2025

...or how to write maintainable code with LLMs fast

FULLY BOOKED Architecting Applications with MCP, A2A and AI Agents, 18 December 2025

...or how to write agentic applications

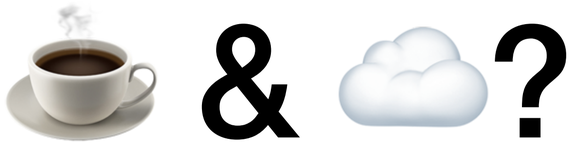
Live, Virtual Online Workshops, Winter 2026:

Effective Java Evolution: Incremental Development and Continuous Improvement with LLMs, 26 February 2026

...or how to incrementally write maintainable code with LLMs

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

by Adam Bien



Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

GraalVM™

reduces RAM footprint

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

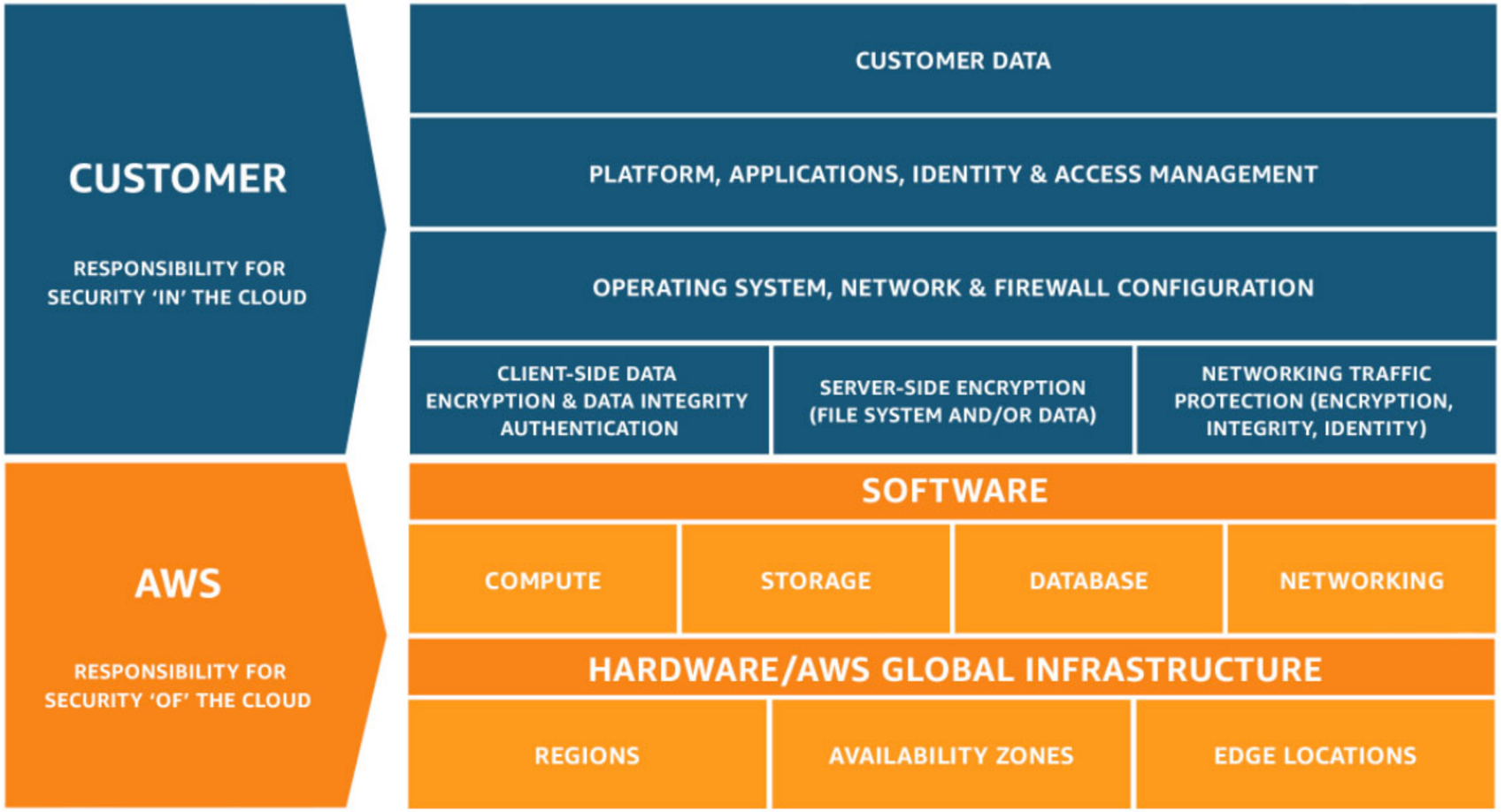
LLMs + Java = ?

	Average	Python	Cpp	Java	JS	Go	Shell	Csharp	Dart	Elixir	Julia	Kotlin	Perl	PHP	Racket	R	Ruby	Rust	Scala	Swift	TS
Count		196	186	188	184	191	188	199	200	198	200	200	200	199	196	198	200	199	199	200	199
Current Upper Bound	*74.8*	*63.3*	*74.7*	*78.7*	*59.2*	*69.1*	*70.7*	*88.4*	*78.0*	*97.5*	*78.0*	*89.5*	*64.5*	*52.8*	*88.3*	*74.2*	*79.5*	*61.3*	*77.4*	*78.0*	*61.3*
Reasoning Mode																					
Claude Opus 4 (20250514)	**52.4**	40.3	44.1	**55.9**	38.6	37.2	**51.6**	74.9	54.0	80.3	**55.5**	72.5	44.5	28.1	**68.9**	**52.5**	61.0	38.7	50.3	50.0	**47.2**
Claude Sonnet 4 (20250514)	51.1	37.2	46.8	52.7	34.8	**41.9**	48.9	72.4	53.5	81.8	49.0	71.5	**45.0**	**34.7**	**68.9**	50.5	54.5	36.2	48.2	48.0	44.2
o3-high (20250416)	51.1	40.8	47.3	53.2	**40.8**	22.0	49.5	68.3	**55.0**	80.8	54.5	72.0	44.0	32.7	53.1	47.5	59.0	**42.2**	51.3	**59.0**	**47.2**
o4-mini (2025-04-16)	50.0	**42.3**	46.8	51.6	40.2	31.4	45.2	68.3	54.0	**82.3**	49.0	**74.0**	44.0	30.2	45.4	43.4	59.0	40.2	50.3	54.0	45.7
Grok-4	50.9	41.2	**48.7**	50.0	37.5	41.4	47.3	72.4	49.5	76.8	**55.0**	70.0	44.0	27.1	63.8	48.5	**61.5**	37.7	**52.8**	51.5	40.7
Gemini2.5 Pro	48.7	40.3	47.5	53.2	37.0	37.2	45.2	70.9	54.0	68.7	54.0	72.0	41.0	29.7	52.6	49.5	56.5	24.6	46.7	49.5	41.7
DeepSeek-R1-0528	50.2	38.8	43.6	52.7	35.9	38.7	46.8	**75.4**	52.5	77.3	52.0	70.0	39.0	28.6	56.1	50.5	58.5	37.2	51.8	55.0	41.2
Seed1.6-enabled (250615)	45.3	39.8	44.6	46.3	28.3	40.8	44.1	60.3	39.5	69.7	51.0	58.0	41.5	25.6	52.6	51.0	52.0	28.6	41.7	47.5	41.2
Seed1.6-Thinking-250715	45.0	40.3	45.2	50.0	33.2	38.2	39.9	67.3	36.5	67.7	51.0	61.0	41.0	26.1	51.0	44.9	55.5	27.6	37.2	46.5	38.7
Seed1.6-Thinking-250615	44.7	38.8	47.0	49.5	38.0	31.4	38.8	62.3	41.0	70.7	45.0	68.0	39.0	25.1	47.5	47.5	50.5	30.7	39.7	41.5	40.2
GLM-4.5-enable	46.6	41.0	43.2	47.9	34.8	37.8	43.9	70.5	42.0	72.5	47.5	66.0	43.5	28.6	50.0	45.0	54.5	31.6	41.0	46.0	42.2
GLM-4.5-Air-enable	40.8	39.3	37.6	39.4	31.0	39.8	36.7	66.3	38.0	61.5	42.0	53.0	40.5	27.1	40.3	39.0	47.0	25.1	30.5	38.5	42.7
ERNIE-X1-Turbo-32K	39.6	39.4	17.8	33.2	32.6	37.4	33.9	46.0	33.0	68.9	54.0	49.5	39.5	23.9	45.3	44.3	48.0	20.8	40.4	44.0	37.7
Qwen3-235B-A22B-Thinking-2507	47.7	37.8	41.9	48.4	39.7	39.8	45.2	71.9	46.0	79.8	48.5	58.0	40.5	29.1	56.6	49.0	55.0	35.7	40.4	46.0	44.2
Qwen3-235B-A22B	45.9	36.7	43.5	47.3	36.4	37.7	42.0	70.9	45.5	68.7	46.0	60.0	39.0	29.1	52.0	47.0	56.5	31.7	43.7	41.5	41.7
Qwen3-32B	41.7	37.8	38.7	39.9	32.6	36.1	39.4	67.8	34.5	65.2	42.5	52.0	40.5	27.6	37.8	44.9	47.0	28.1	37.2	42.0	40.2
Qwen3-14B	37.6	37.8	35.5	35.1	30.4	30.4	36.2	60.8	29.0	62.1	34.5	44.5	37.5	23.1	44.9	36.9	43.5	24.6	28.6	36.0	38.7
Qwen3-8B	28.5	28.1	22.6	21.8	28.3	29.3	27.1	52.8	21.0	43.9	29.0	36.0	35.5	18.6	13.3	30.8	37.0	12.6	21.1	22.0	37.7
Qwen3-4B	24.3	27.6	17.7	22.3	25.5	24.1	28.2	42.2	13.0	33.3	20.0	29.5	34.5	16.1	11.7	23.7	27.5	8.5	20.1	20.0	39.2
Qwen3-1.7B	11.2	16.8	5.4	4.8	12.5	9.9	11.7	19.6	7.0	20.7	11.0	9.0	19.5	7.5	5.6	9.6	21.0	0.0	2.5	10.0	19.6
Non-Reasoning Mode																					
Claude Opus 4 (20250514)	**50.9**	37.8	45.7	50.0	**38.0**	35.6	47.3	73.9	**57.0**	**82.3**	**55.0**	**75.5**	43.0	26.6	**64.8**	47.0	**54.0**	**38.2**	**46.7**	**51.5**	**46.7**
Claude Sonnet 4 (20250514)	49.3	35.7	47.3	**52.7**	**38.0**	**37.7**	**47.9**	72.9	51.0	74.2	51.0	72.0	44.0	**30.7**	63.8	44.4	51.5	35.2	45.2	45.5	44.2
GPT4.1 (2025-04-14)	48.0	37.2	46.8	48.9	34.8	37.2	36.7	74.4	46.5	76.8	50.0	72.0	43.5	29.2	50.5	42.4	**54.0**	37.2	44.2	49.5	46.2
GPT4o (2024-11-20)	41.1	33.7	37.1	43.2	34.8	30.9	29.3	65.5	43.5	62.6	36.0	67.0	43.0	26.6	37.2	32.8	43.0	29.6	38.2	45.0	39.2
Gemini2.5 Flash	45.7	39.3	44.1	50.0	33.2	33.0	37.8	68.3	49.5	64.0	47.5	70.0	39.5	24.1	38.3	**51.5**	53.0	36.2	44.2	46.5	41.2
DeepSeek-V3-0324	48.1	36.7	**48.4**	**52.7**	31.5	34.6	37.8	72.9	48.0	75.8	49.0	69.0	42.5	28.1	59.2	45.0	52.5	37.2	**46.7**	48.0	43.7
DeepSeek-Coder-V2	37.7	29.1	34.9	34.0	27.7	29.8	31.4	63.8	33.5	60.6	37.5	58.5	35.5	25.1	41.8	35.4	45.0	22.6	33.2	38.0	35.7
DeepSeek-Coder-33B-Instruct	28.5	25.0	24.2	29.3	24.5	29.8	22.3	54.8	17.5	67.7	14.5	52.0	29.5	19.1	28.1	18.7	33.0	8.0	24.1	18.0	29.1
DeepSeek-Coder-6.7B-Instruct	20.5	18.9	12.9	19.7	19.6	21.5	16.0	44.2	11.5	47.5	15.5	45.5	21.5	10.6	15.3	13.1	27.5	6.0	11.1	8.0	23.6
Kimi-K2-0711-Preview	47.8	38.8	42.5	47.9	37.5	31.4	40.4	**75.9**	50.0	80.3	52.0	68.0	41.5	28.1	57.1	40.4	52.5	36.7	42.7	47.0	42.2
Hunyuan-TurboS-20250716	43.8	34.2	34.9	47.9	32.6	34.6	38.3	64.8	44.5	70.7	47.0	62.0	42.0	30.2	45.9	39.9	53.0	30.7	39.2	39.5	42.2
Hunyuan-Coder-7B-Preview	33.4	25.5	26.9	34.0	27.2	21.5	29.3	54.8	36.0	60.1	30.5	57.0	34.0	22.2	37.8	27.3	40.5	11.1	31.2	27.6	31.2
Seed1.6-disabled (250615)	42.9	35.2	40.3	46.8	32.6	34.6	35.1	70.9	42.5	69.7	45.0	62.0	39.5	23.1	49.5	40.4	46.5	28.1	32.7	40.0	42.7
Seed-Coder-8B-Instruct	32.3	23.5	23.7	33.5	28.8	22.5	20.7	54.8	30.5	57.1	33.0	52.5	34.0	25.1	36.7	28.3	35.5	15.6	29.6	28.0	31.2
Qwen3-Coder-480B-A35B-Instruct	44.8	**39.4**	41.1	51.1	27.9	31.4	41.1	63.0	36.5	73.7	49.5	63.1	41.0	27.2	56.3	42.7	51.5	25.4	42.1	47.5	41.9
Qwen3-235B-A22B-Instruct-2507	43.1	35.7	38.2	49.5	29.3	33.5	40.4	67.3	39.5	59.1	46.0	59.5	**44.5**	26.1	49.5	44.0	46.5	24.6	37.7	46.0	43.2
Qwen3-32B	31.0	26.5	21.5	29.5	28.0	25.5	24.0	59.3	27.5	52.0	28.0	45.0	34.5	21.6	28.6	22.7	36.5	16.1	26.6	30.0	35.2
Qwen3-14B	28.6	24.5	22.6	32.4	27.2	16.8	23.9	50.8	21.0	42.4	22.5	42.0	34.5	24.6	28.1	26.3	33.5	17.1	20.6	26.5	32.7
Qwen3-8B	23.3	22.4	11.3	25.0	22.8	18.3	22.3	42.2	17.0	41.4	18.5	36.0	29.5	18.1	23.5	16.2	27.0	7.5	19.1	17.5	29.1
Qwen3-1.7B	7.9	8.7	1.1	2.7	8.2	3.7	11.7	9.5	3.5	17.2	6.0	11.5	15.5	7.5	4.6	7.6	14.5	0.5	3.0	6.0	14.6
Qwen2.5-Coder-32B	35.8	29.6	27.4	33.0	29.9	23.0	29.3	58.3	34.5	59.6	35.5	56.0	38.5	26.1	35.7	31.3	40.0	23.1	29.6	39.0	35.2
Qwen2.5-72B	34.3	33.2	24.7	31.9	28.8	19.9	31.9	54.3	34.0	65.7	30.5	50.5	30.0	22.6	39.3	27.3	42.0	23.1	32.7	29.5	33.2
Qwen2.5-Coder-7B	22.5	19.9	8.6	22.3	21.2	12.6	21.3	38.7	18.5	47.0	18.0	39.0	27.5	15.1	24.0	17.7	29.5	7.0	19.1	17.0	24.6
Qwen2.5-Coder-1.5B	10.3	12.2	2.7	4.8	12.0	7.3	14.4	17.6	6.5	35.4	4.0	16.5	15.0	5.0	7.1	7.6	15.5	1.0	4.5	5.0	11.1
OpenCoder-8B	19.3	14.3	9.7	19.1	12.5	17.3	21.3	33.2	15.5	34.3	15.5	30.0	26.5	14.6	15.8	17.2	29.5	6.0	17.1	15.0	21.1

Table 4: Pass@1 (%) performance of different models for AutoCodeBench. ***Current Upper Bound*** represents the Pass@1 value calculated by taking the union of problems correctly solved by all models.

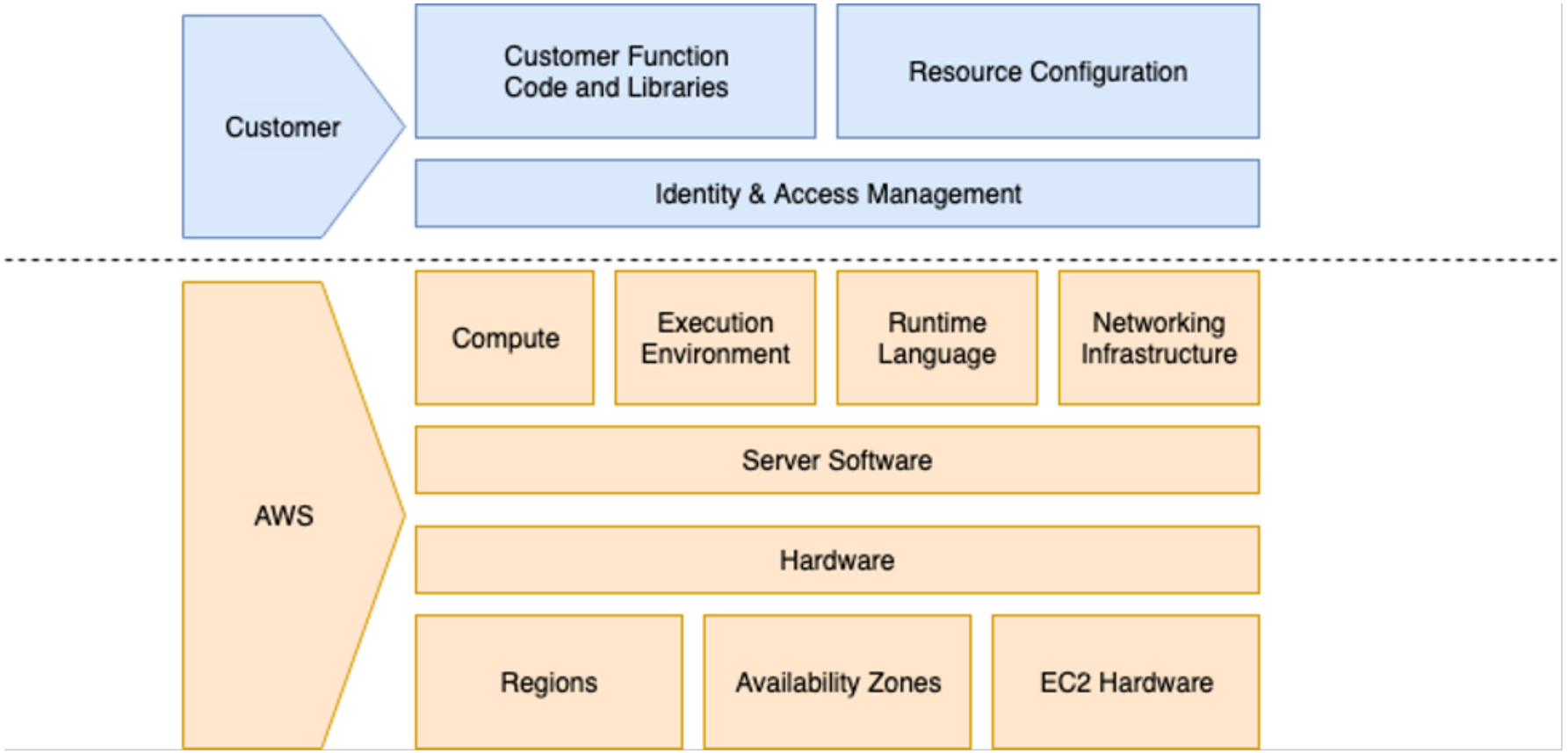
https://github.com/Tencent-Hunyuan/AutoCodeBenchmark/tree/main

traditional…



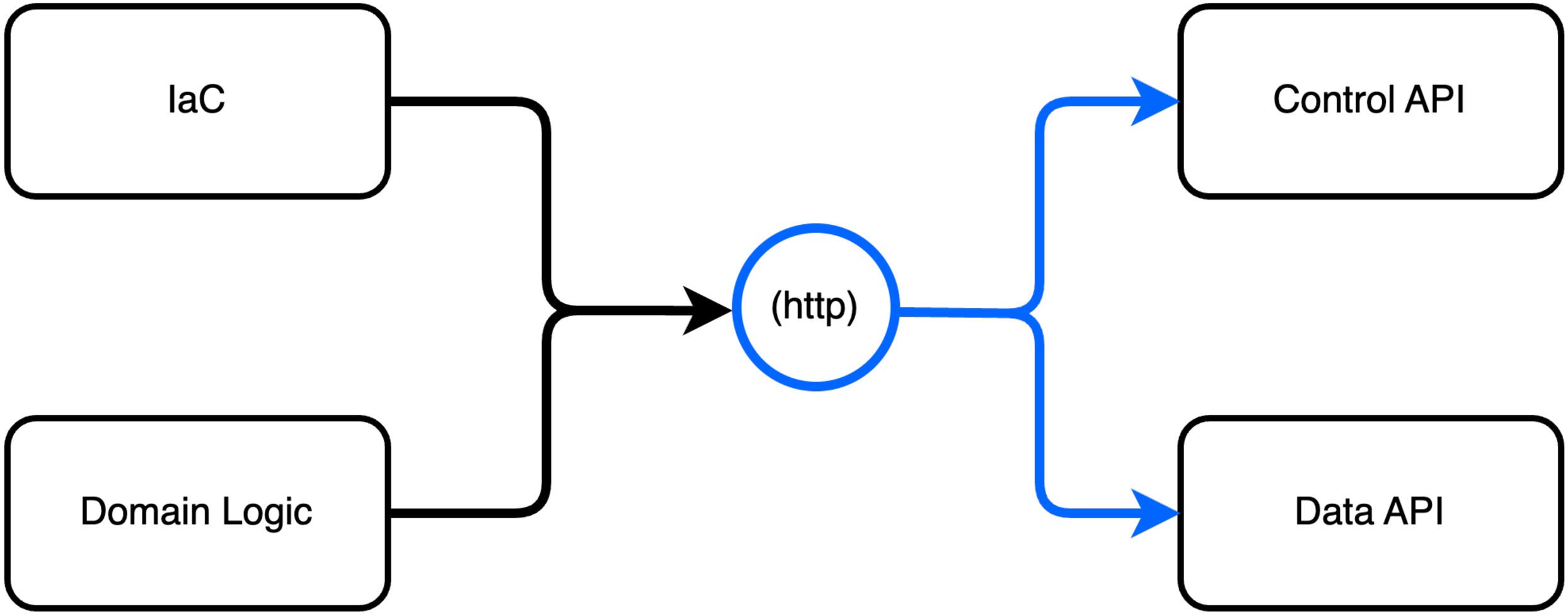
https://aws.amazon.com/compliance/shared-responsibility-model/

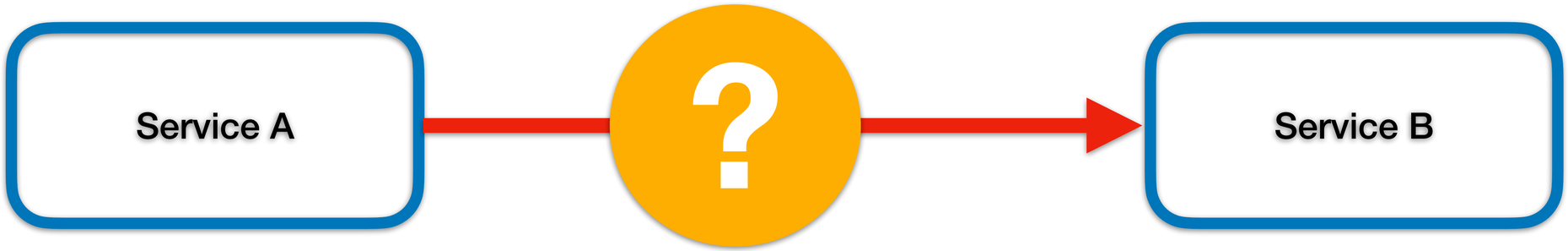
…optimal



https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/the-shared-responsibility-model.html

“cloud native”


IaC
Domain Logic
(http)
Control API
Data API


Service A
?
Service B

cloud & security

Kriterienkatalog C5

Bild-Dokument für das Frontend

Der Kriterienkatalog C5 (Cloud Computing Compliance Criteria Catalogue) spezifiziert Mindestanforderungen an sicheres Cloud Computing und richtet sich in erster Linie an professionelle Cloud-Anbieter, deren Prüfer und Kunden.

Der Kriterienkatalog C5 wurde im Jahr 2016 durch das Bundesamt für Sicherheit in der Informationstechnik erstmalig veröffentlicht und hat sich in den letzten Jahren erfolgreich im Markt durchgesetzt: Nach Kenntnis des BSI wurden bereits mehr als ein Dutzend Testate für nationale, europäische und weltweite Cloud-Anbieter sowie eine breite Palette an Cloud-Diensten erstellt. Mittlerweile gibt es auch mittelständische und kleinere Anbieter, die den Katalog anwenden. Der C5 bietet Cloud-Kunden eine wichtige Orientierung für die Auswahl eines Anbieters. Er bildet die Grundlage, um ein kundeneigenes Risikomanagement durchführen zu können. Im Jahr 2019 wurde der C5 grundlegend überarbeitet, um auf aktuelle Entwicklungen einzugehen und die Qualität noch weiter zu erhöhen.

Cloud Computing Compliance Criteria Catalogue

(C5)

Übersicht

Der Cloud Computing Compliance Criteria Catalog (C5) ist ein von deutschen Behörden gefördertes Zertifizierungsschema. Dieses Schema wurde in Deutschland durch das Bundesamt für Sicherheit in der

https://aws.amazon.com/de/compliance/bsi-c5/

Cloud Computing-Konformitätskriterienkatalog (C5)

C5 – Überblick

Im Jahr 2016 erstellte das Bundesamt für Sicherheit in der Informationstechnik (BSI) den Anforderungskatalog „Cloud Computing Compliance Controls Catalog“ (C5). Das BSI hat den Leitfaden im Jahr 2020 als Cloud Computing Compliance Criteria Catalog (C5:2020) überarbeitet. Der C5 ist ein geprüfter Standard, der verbindliche Mindestanforderungen für die Cloudsicherheit und die Einführung von

https://learn.microsoft.com/de-de/compliance/regulatory/offering-c5-germany

Lambda & Security

- Firecracker microVMs: virtualization technology providing hardware-level isolation for each Lambda execution environment
- microVMs launch in ~125ms
- Minimal attack surface: Firecracker runs only 50k
- Multi-tenant isolation: each function execution runs in its own microVM with dedicated kernel, preventing cross-tenant data leakage or side-channel attacks
- combines microVM isolation with cgroups, namespaces, seccomp filters, and IAM policies for layered security

AWS Lambda

AWS Lambda does not include Log4j2 in its managed runtimes or base container images. These are therefore not affected by the issue described in CVE-2021-44228 and CVE-2021-45046.

For cases where a customer function includes an impacted Log4j2 version, we have applied a change to the Lambda Java managed runtimes and base container images (Java 8, Java 8 on AL2, and Java 11) that helps to mitigate the issues in CVE-2021-44228 and CVE-2021-45046. Customers using managed runtimes will have the change applied automatically. Customers using container images will need to rebuild from the latest base container image, and redeploy.

Independent of this change, we strongly encourage all customers whose functions include Log4j2 to update to the latest version. Specifically, customers using the aws-lambda-java-log4j2 library in their functions should update to version 1.4.0 and redeploy their functions. This version updates the underlying Log4j2 utility dependencies to version 2.16.0. The updated aws-lambda-java-log4j2 binary is available at the Maven repository and its source code is available in Github.

EC 2 / Nitro

- Hardware and hypervisor-level: security through dedicated hardware cards
- IAM roles attached via instance metadata service (IMDS)
- applications retrieve temporary credentials from IMDS to sign API requests
- compute, storage, and networking functions offloaded to dedicated Nitro Cards
- isolated compute environments for sensitive data processing
- no AWS operator access to customer instances

EC 2 / Nitro


aws
NVIDIA
GB200 with
AWS Nitro
aws
NVIDIA
GB200 with
AWS Nitro
CAUTION
CAUTION
CAUTION

S3

- Access control layers
- Encryption at rest
- Encryption in transit
- Access logging & auditing
- Object Lock & versioning
- AWS used "lightweight formal methods" to validate correctness

“max cloud” 👉 serverless

DEFINITION

NoOps (no operations)

By **Rahul Awati**

What is NoOps (no operations)?

NoOps (no operations) is a concept that an IT environment can become so automated and abstracted from the underlying infrastructure that there's no need for a dedicated team to manage software in-house.

First coined by research and advisory company Forrester, the term describes the goal of NoOps as to "improve the process of deploying applications" so that "application developers will never have to speak with an operations professional again." Forrester's Mike Gualtieri called DevOps "a step backward" even though, in his short 2011 article, he wrote that he admires its goal to improve application release deployment processes.

DevOps Advantages

Reduced Operational Overhead

- No server provisioning, patching, or maintenance
- Automatic scaling and high availability
- Built-in monitoring and logging

Reduced Operational Overhead

- Accelerated development (deploy on git push)
- Faster deployment cycles (seconds vs. hours)
- Focus on business logic, not infrastructure
- Environment parity (dev = prod)

Cost (Power) Optimization

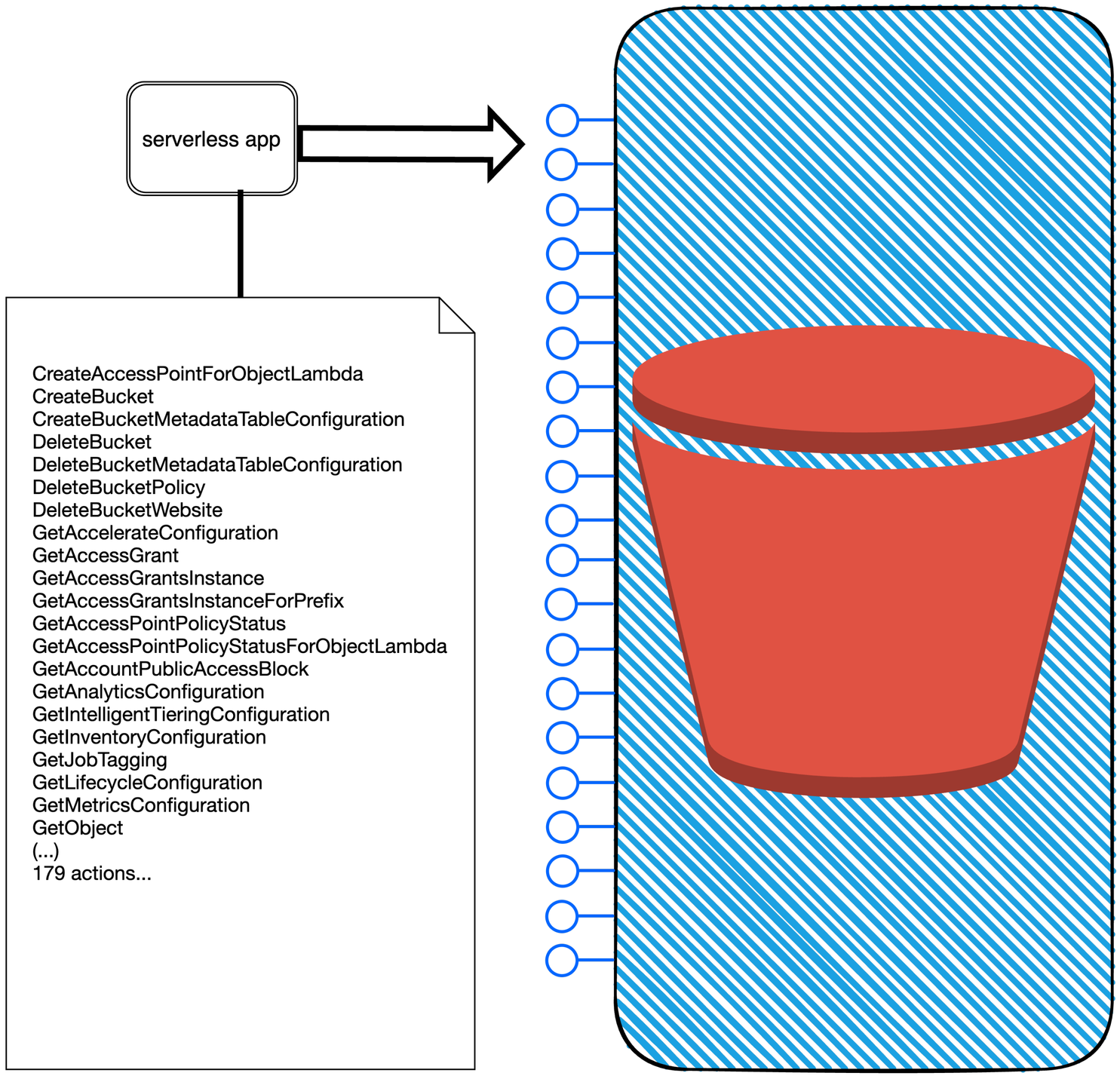
- Pay-per-execution model
- No idle capacity costs
- Automatic resource optimization

Infrastructure as Code (IaC) Advantages

Infrastructure as Code (IaC)

- Functions = deployable units
- Infrastructure defined alongside code
- Single source of truth (in Java CDK)
- Self-provisioned runtimes

fine grained permissions



Enhanced Reproducibility

- Immutable deployments
- Drift detection
- Provision on “git push”

on-premise

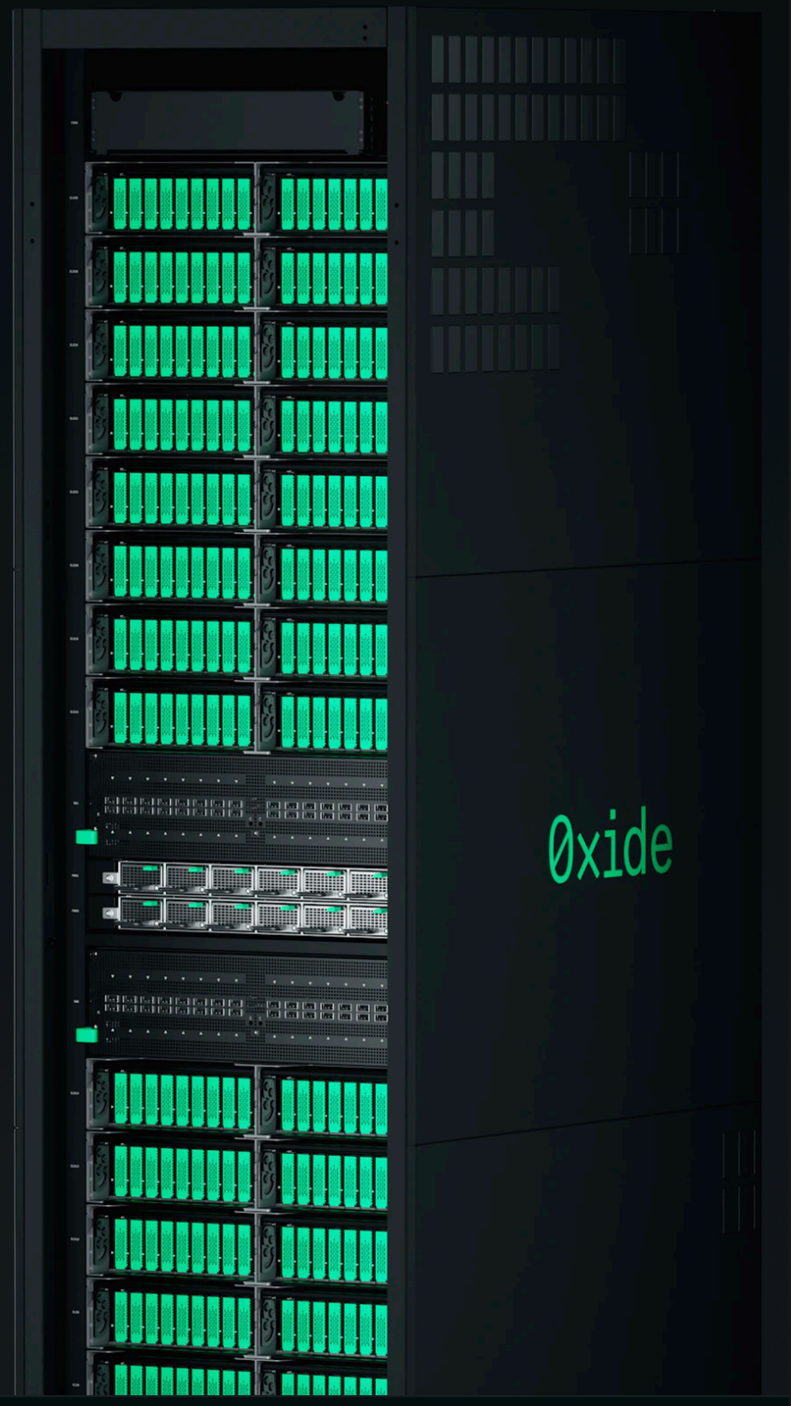
v2

Oxide Cloud Computer

No Cables. No Assembly.
Just Cloud.

CONTACT SALES

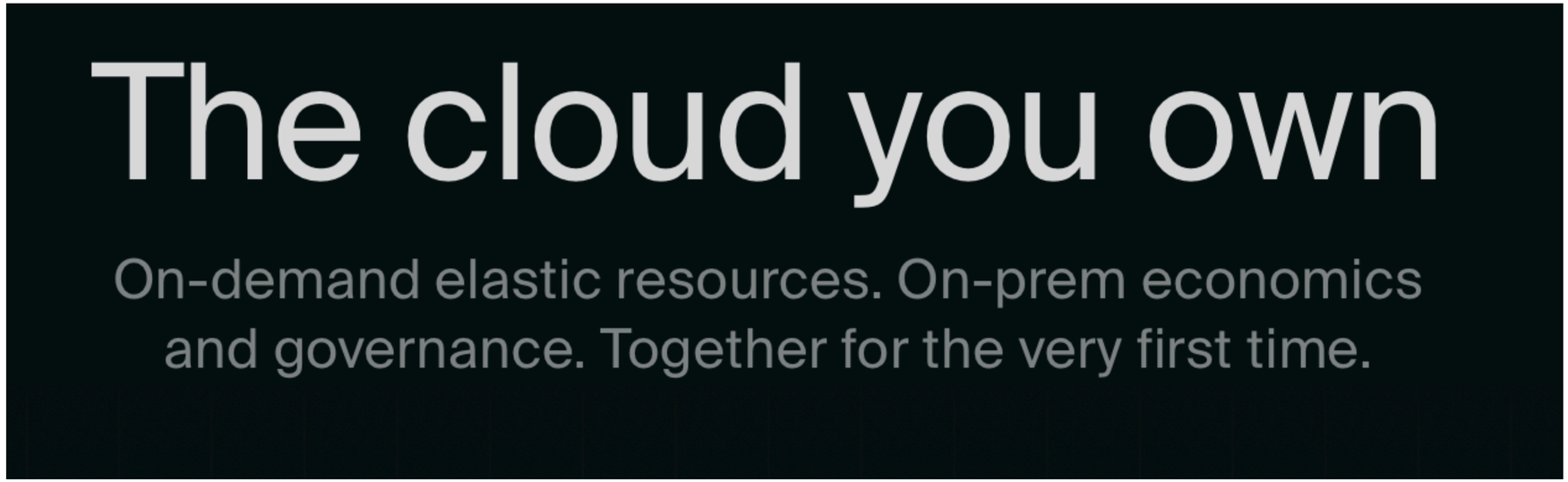
GET OUR NEWSLETTER



Max:

~2000 AMD Cores

~32 TiB RAM



oxide.computer

Oxide Computer

Elastic compute capacity from a fluid resource pool

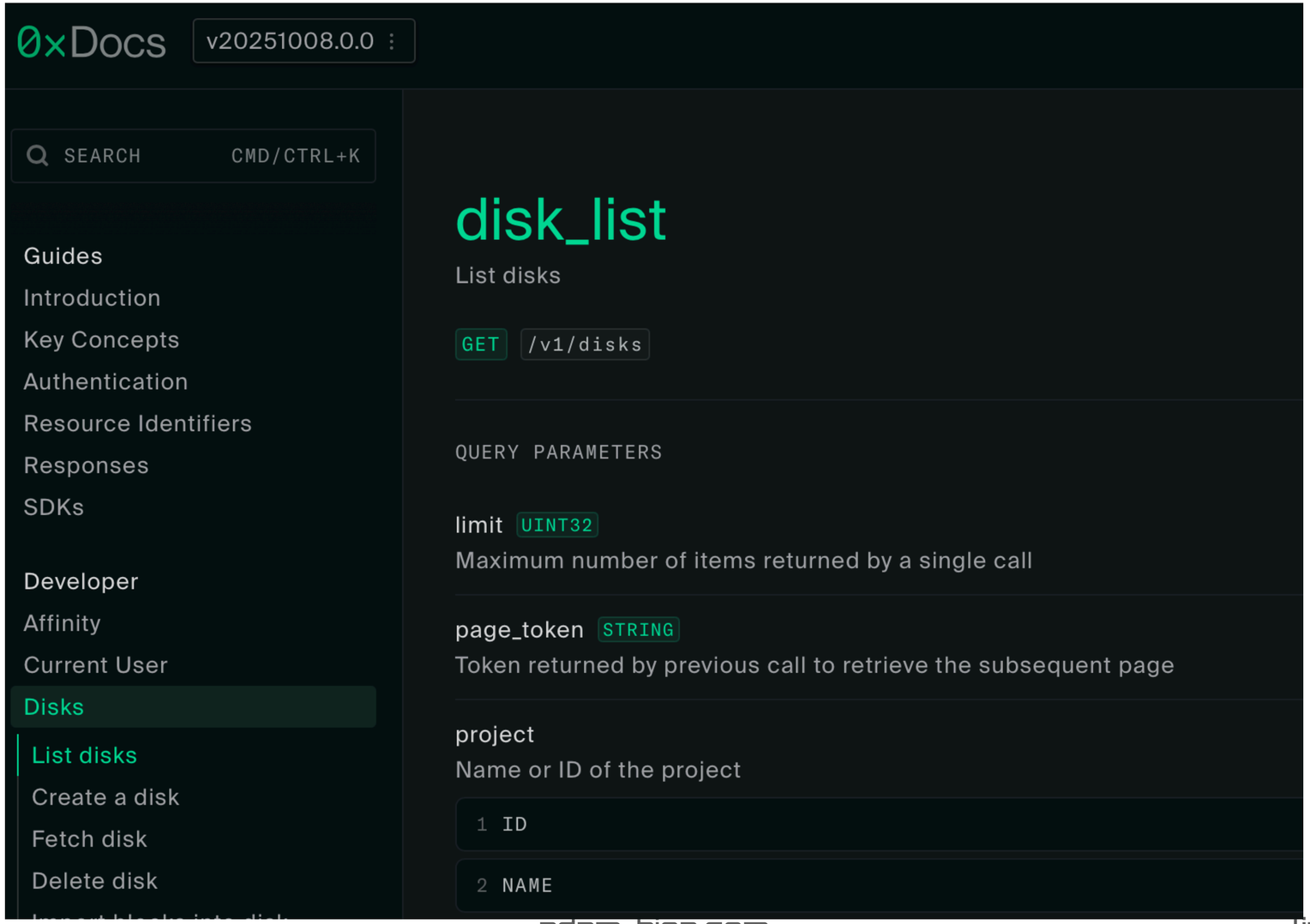
High performance, persistent block storage with configurable capacity and IOPS

VPC and network virtualization capabilities with per-tenant isolation

API-driven, self-service infrastructure

Virtual machines (custom hypervisor)

Oxide Computer



Service Meshes (e.g. RHOSM)

Envoy sidecar proxies handle security transparently

Service identities via Kubernetes service accounts

Mutual TLS (mTLS) between services, automatic certificate rotation

AuthorizationPolicy resources defining service-to-service communication rules

Distributed certificate authority

Best Infrastructure

“Cheap” RAM + CPU

JFR + Custom Events

Cryostat

Virtual Threads

Leyden

JLink / GraalVM

“Bare Metal”

Best Infrastructure

CraC

Azul Intelligence Cloud: Java vulnerabilities and dead/unused code detection

ReadyNow

(…)

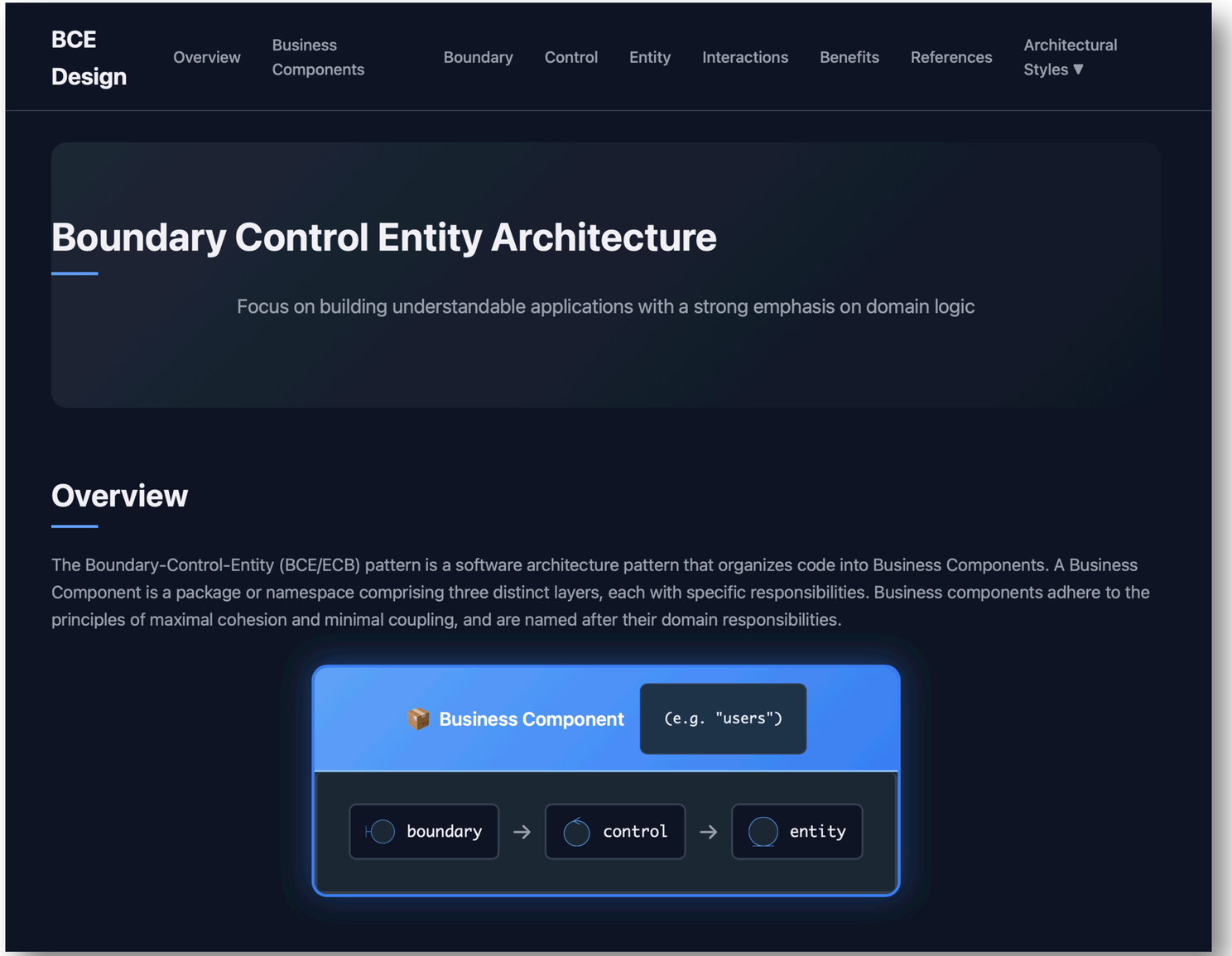
Jakarta EE + MicroProfile = The Cloud OS

on premise


JAKARTA EE 10 PLATFORM
JAKARTA EE 10 WEB PROFILE
JAKARTA EE 10 CORE PROFILE
Expression Language 5.0
Server Pages 3.1
Authorization 2.1
Authentication 3.0
CDI 4.0
CDI Lite 4.0
Activation 2.1
Concurrency 3.0
WebSocket 2.1
JSON Binding 3.0
Batch 2.1
Persistence 3.1
Bean Validation 3.0
Annotations 2.1
Connectors 2.1
Faces 4.0
Debugging Support 2.0
Interceptors 2.1
Mail 2.1
Security 3.0
Enterprise Beans Lite 4.0
RESTful Web Services 3.1
Messaging 3.1
Servlet 6.0
Managed Beans 2.0
JSON Processing 2.1
Enterprise Beans 4.0
Standard Tag Libraries 3.0
Transactions 2.0
Dependency Injection 2.0
Updated
Not Updated
New

in the clouds

https://bce.design



Why not:

Automate everything

Immutable infrastructure

Build, provision / install on push

A production environment per developer

System / e2e Test

Write Once, Run on Functions, Containers, VMs and Bare Metal

Use LLMs to write better code faster

references

https://bce.design

https://microprofile.io

https://github.com/AdamBien/quarkus-microprofile (used in the session)

https://github.com/AdamBien/ebank

https://github.com/AdamBien/bce-icons/

https://github.com/AdamBien/bce.design (web components)

https://constructions.cloud

https://youtube.com/bienadam/shorts

https://airhacks.industries

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with Adam Bien

Live, Virtual Online Workshops, Winter 2025:

FULLY BOOKED The 50x Java Dev: Crafting Elegant Code with LLMs, 16 December 2025

...or how to write maintainable code with LLMs fast

FULLY BOOKED Architecting Applications with MCP, A2A and AI Agents, 18 December 2025

...or how to write agentic applications

Live, Virtual Online Workshops, Winter 2026:

Effective Java Evolution: Incremental Development and Continuous Improvement with LLMs, 26 February 2026

...or how to incrementally write maintainable code with LLMs

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

by Adam Bien

Thank YOU!

airhacks.industries

airhacks.live