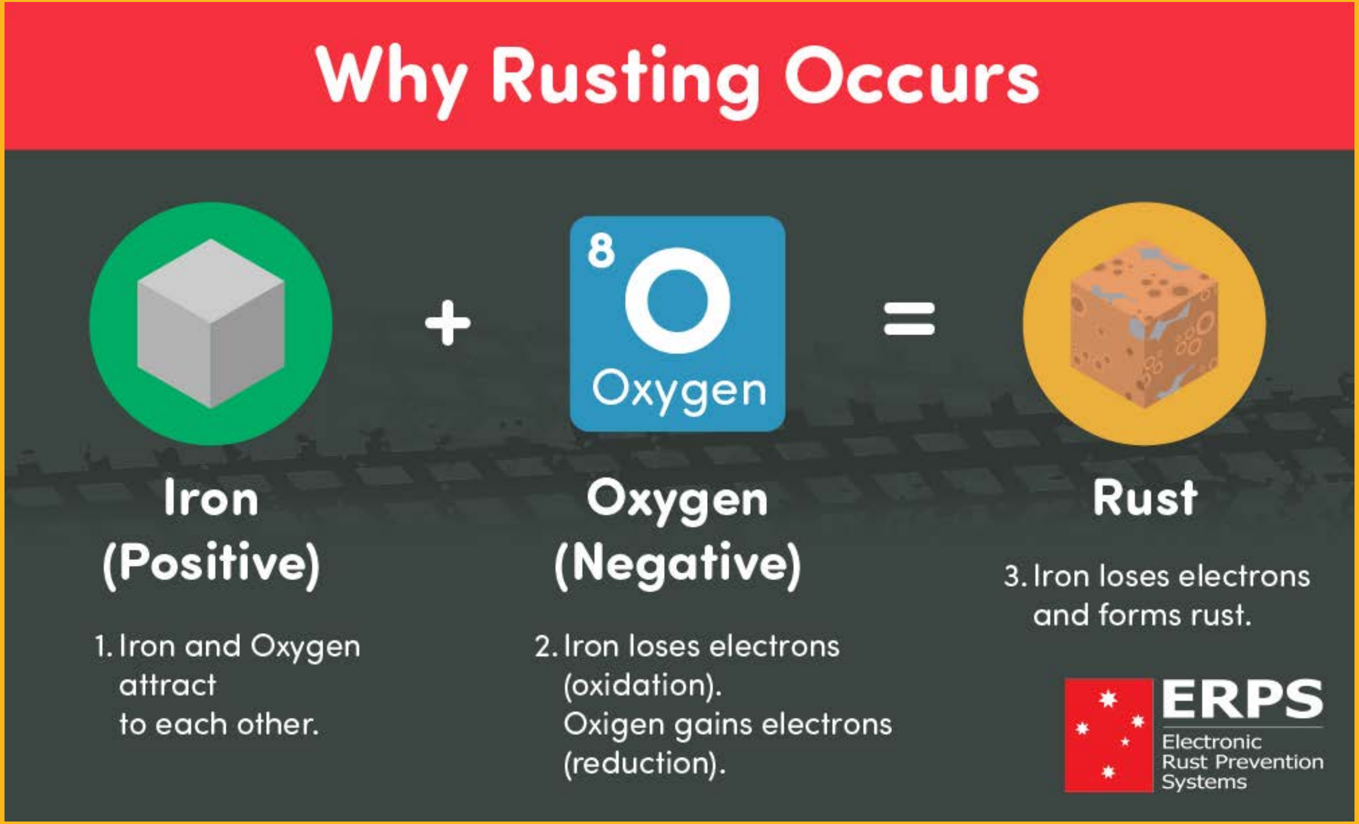
Matthias Endler

RUST IN PRODUCTION

corrode

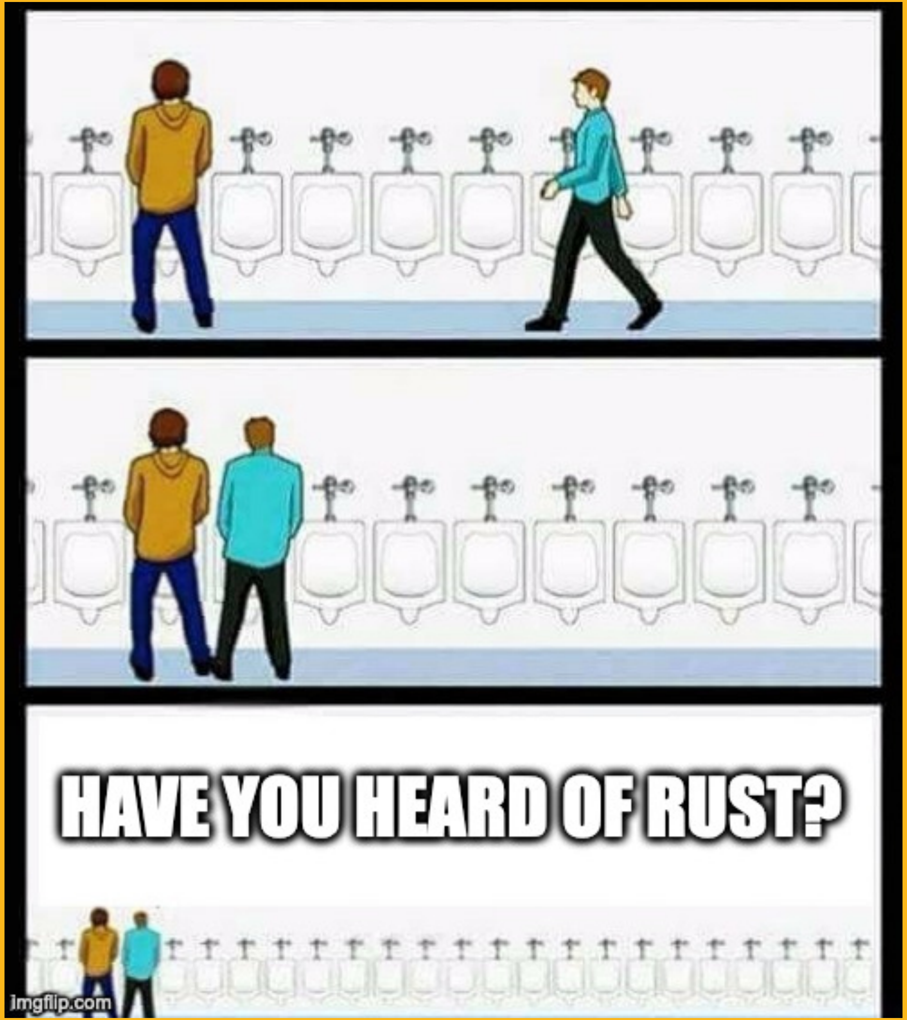
WHAT IS RUST?

corrode


Why Rusting Occurs
8
O
Oxygen
Iron
(Positive)
1. Iron and Oxygen attract to each other.
Oxygen
(Negative)
2. Iron loses electrons (oxidation). Oxigen gains electrons (reduction).
Rust
3. Iron loses electrons and forms rust.
ERPS
Electronic Rust Prevention Systems

https://www.erps.com.au/what-is-rust

corrode



corrode

```rust
use std::collections::HashMap;

fn main() {
    let mut dict = HashMap::new();
    "mississippi".chars().for_each(|char| {
        let item = dict.entry(char).or_insert(0);
        *item += 1;
    });

    println!("{dict:?}");
}
```

corrode

```rust
use std::collections::HashMap;

fn main() {
    let mut dict = HashMap::new();
    "mississippi".chars().for_each(|char| {
        let item = dict.entry(char).or_insert(0);
        *item += 1;
    });

    println!("{dict:?}");
}
```

OUTPUT:

```
{'p': 2, 's': 4, 'i': 4, 'm': 1}
```

corrode

Me explaining Rust's ownership & borrowing, lifetimes, generics and expression syntax

You

corrode

```rust
let v = String::from("Hello Rust");
my_function(v);
```

corrode

✓

```
let v = String::from("Hello Rust");
my_function(v);
```

corrode

```rust
let v = String::from("Hello Rust");
my_function(v);
my_function(v);
```

```rust
let v = String::from("Hello Rust");
my_function(v);
my_function(v);
```

```
error[E0382]: use of moved value: `v`
```

corrode

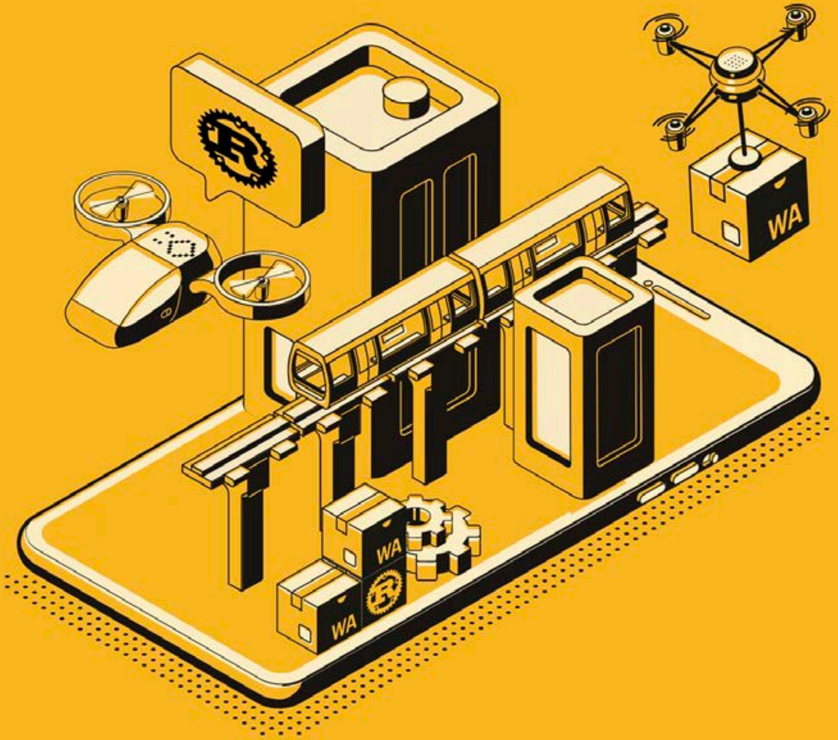
```rust
let v = String::from("Hello Rust");
delete(v);
reuse(v) // bug
```

```
error[E0382]: use of moved value: `v`
```

corrode

ABOUT RUST

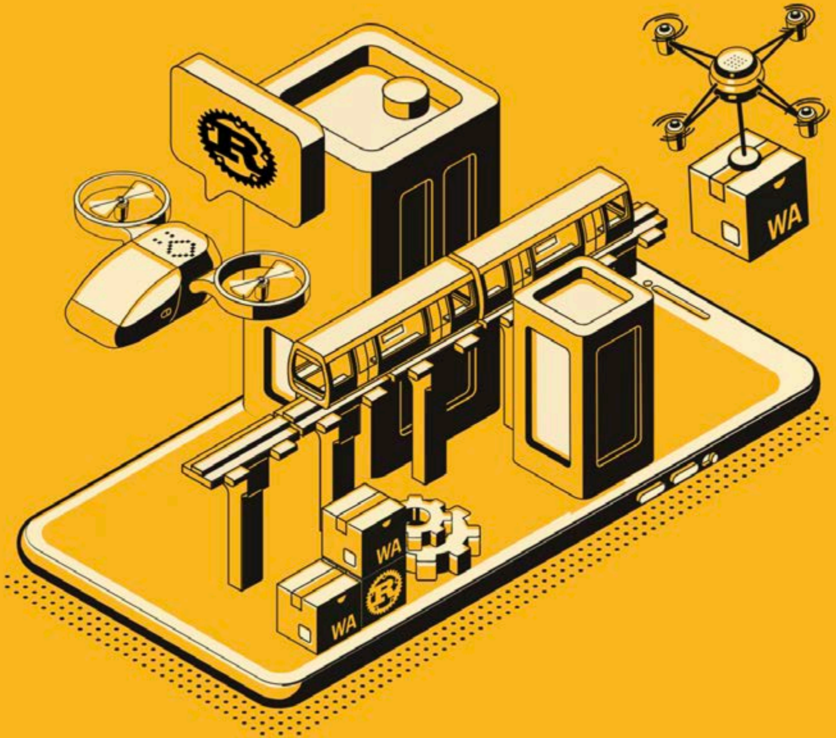
2006	Research project at Mozilla
2015	Version 1.0
2017	First Rust code in Firefox
2021	Rust Foundation
2024	3.7 million Rust users



corrode

MY RUST EXPERIENCE

- Using Rust since 2015
- Rust Cologne User Group
- *"Hello Rust"* YouTube channel
- Open Source work
- Rust consultancy
- Rust in production since 2020



corrode

WHY WAS RUST CREATED?

- C++ code in Firefox had many security issues.
- Multithreaded code is hard to write with C++ (data races).

corrode

RUST VS OTHER LANGUAGES

Source: Jon Gjengset - Considering Rust

corrode

VS PYTHON

Much faster.

Much lower memory use.

Multi-threading.

Algebraic data types.

Pattern matching.

Comprehensive static typing, so:

Many fewer runtime crashes.

VS C/C++

No segfaults.

No buffer overflows.

No null pointers.

No data races.

Powerful type system.

Unified build system.

Dependency management.

VS GO

No GC pauses; lower memory use.

No null pointers.

Nicer error handling.

Safe concurrency.

Stronger type system.

Zero-cost abstractions.

Dependency management.

VS JAVA

No JVM overhead or GC pauses.

Much lower memory use.

Zero-cost abstractions.

ConcurrentModificationException

Pattern matching.

Unified build system.

Dependency management.

VS JAVASCRIPT

Multi-threaded by design

Stronger typesystem

Static typing

No runtime overhead

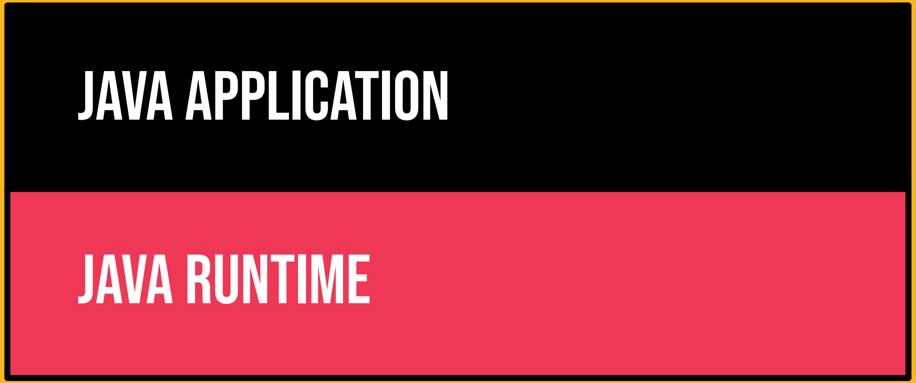
Unified build system

Zero-cost abstractions

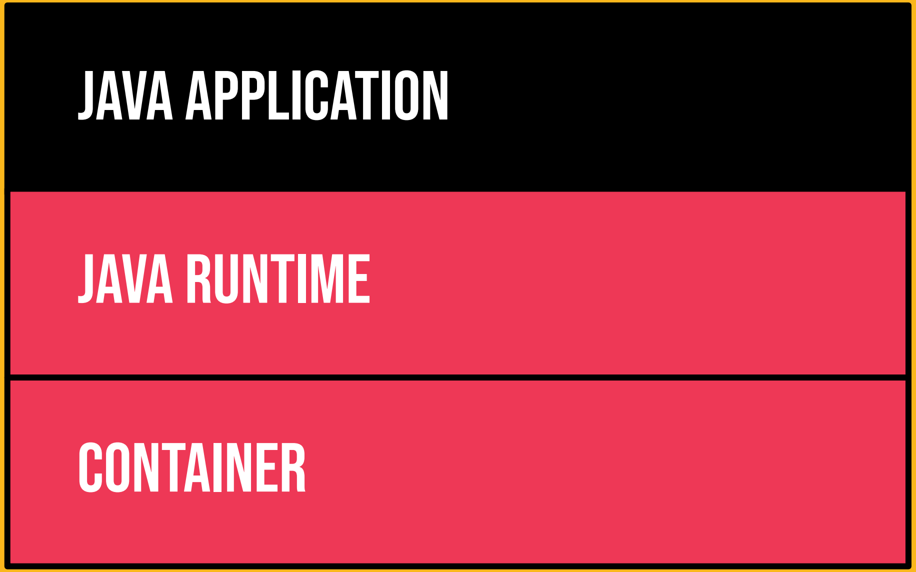
COMPARISON

	RUST	JAVA	GO	JS/TS
COMMUNITY SIZE	Small	**Large**	**Medium/Large**	**Large**
ECOSYSTEM	Small	**Large**	Medium	**Large**
OPERATIONAL COST	**Low**	High	Medium	Medium
SAFETY	**High**	Low/Medium	Low/Medium	Low/Medium
PERFORMANCE	**High**	Medium	Medium	Low
TOOLING	**Good**	Okay	Okay	Okay
COMPLEXITY	High	Medium	**Low**	**Low**

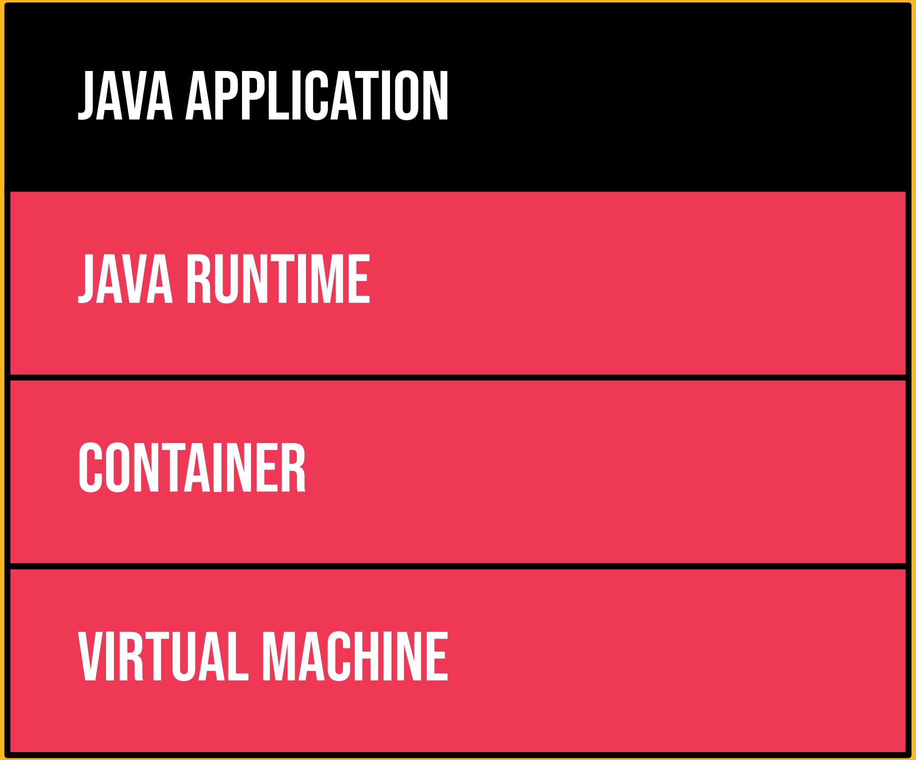
TANGENT: JAVA RUNTIME BEHAVIOR



TANGENT: JAVA RUNTIME BEHAVIOR

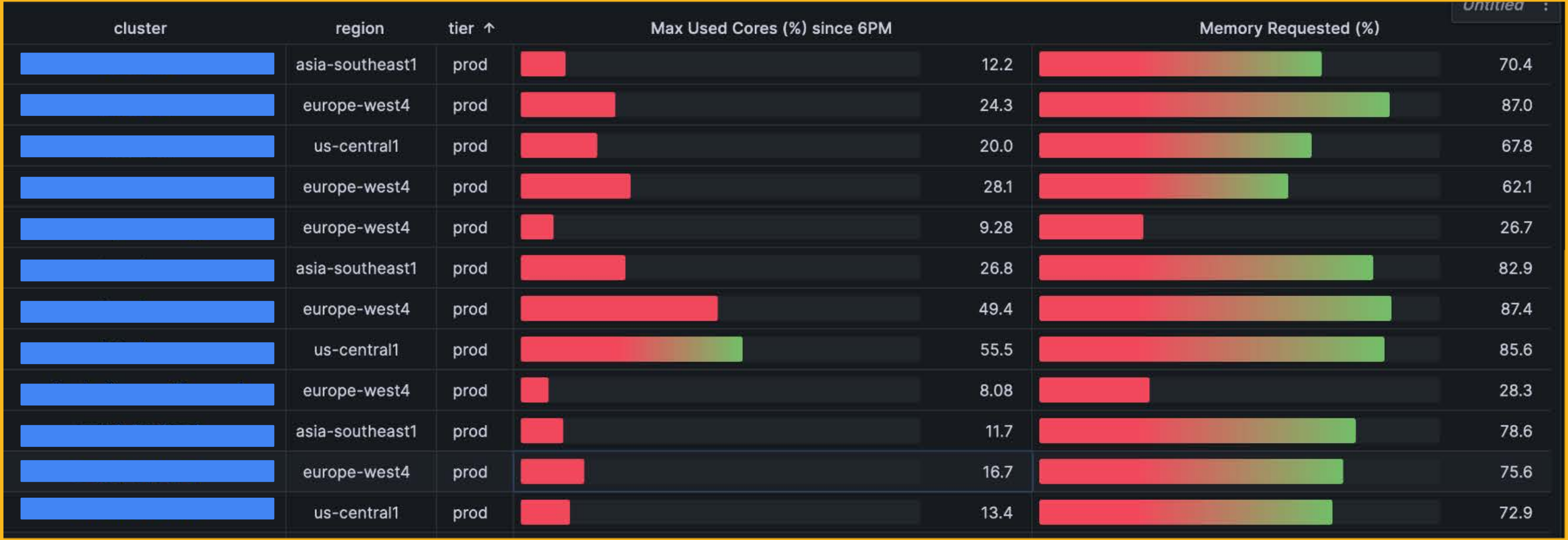


TANGENT: JAVA RUNTIME BEHAVIOR



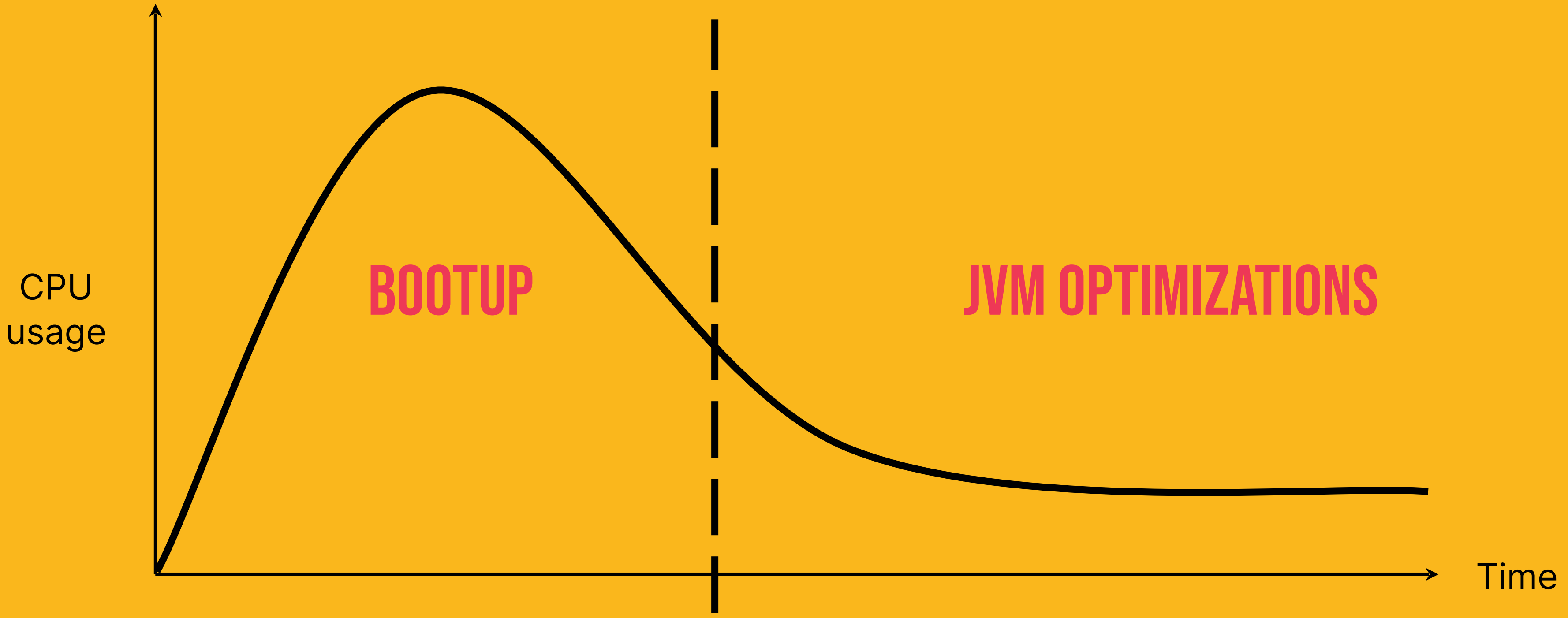
corrode

TANGENT: JAVA RUNTIME BEHAVIOR



cluster	region	tier ↑	Max Used Cores (%) since 6PM	Memory Requested (%)
[redacted]	asia-southeast1	prod	12.2	70.4
[redacted]	europe-west4	prod	24.3	87.0
[redacted]	us-central1	prod	20.0	67.8
[redacted]	europe-west4	prod	28.1	62.1
[redacted]	europe-west4	prod	9.28	26.7
[redacted]	asia-southeast1	prod	26.8	82.9
[redacted]	europe-west4	prod	49.4	87.4
[redacted]	us-central1	prod	55.5	85.6
[redacted]	europe-west4	prod	8.08	28.3
[redacted]	asia-southeast1	prod	11.7	78.6
[redacted]	europe-west4	prod	16.7	75.6
[redacted]	us-central1	prod	13.4	72.9

TANGENT: JAVA RUNTIME BEHAVIOR



corrode

RUST'S STRENGTHS

corrode

SAFETY
PERFORMANCE
RUST DEVELOPER
Did you just take both pills?
imgflip.com

corrode

“

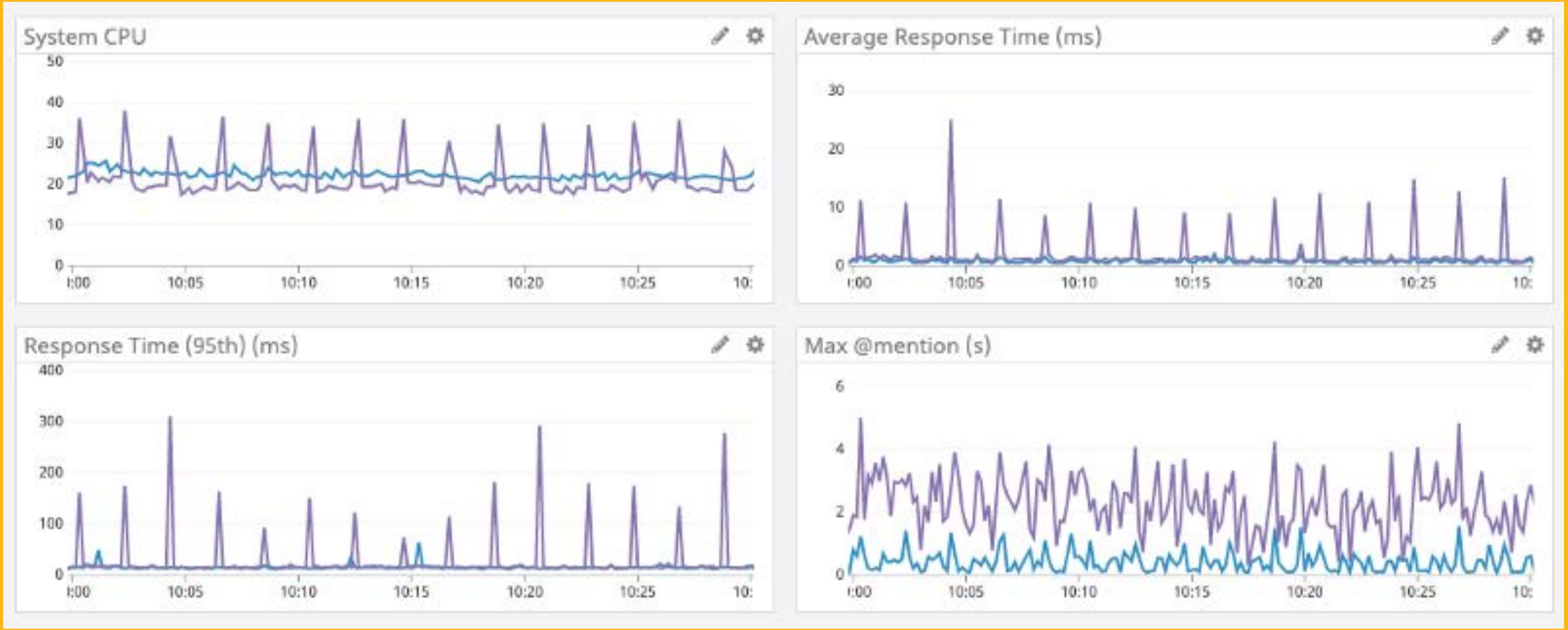
THE REASON THAT PEOPLE USE RUST IS BECAUSE ACTUALLY IT'S BETTER FOR BUILDING MORE RELIABLE SYSTEMS.

Niko Matsakis, Lead of Rust Language Design Team

corrode

DISCORD: SWITCHING FROM GO TO RUST

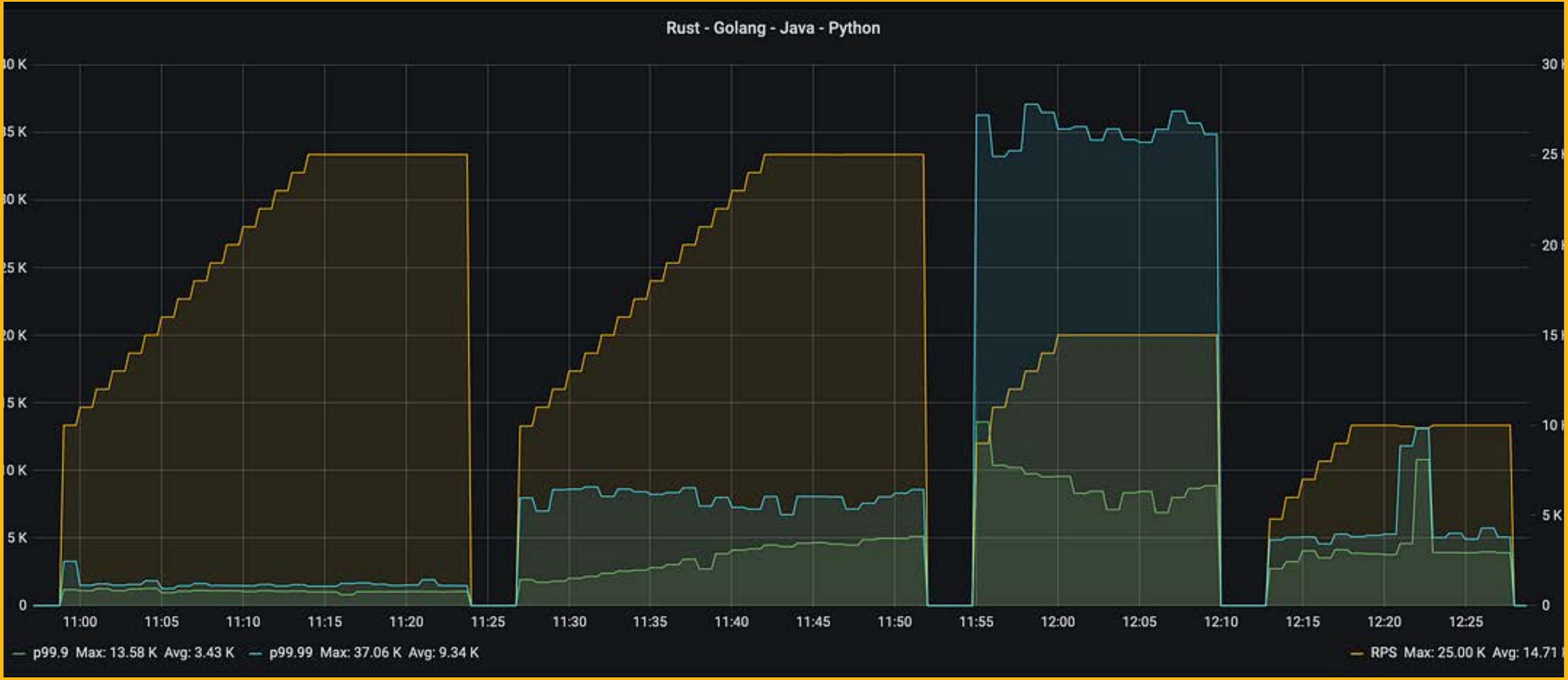
● Rust ● Go



https://discord.com/blog/why-discord-is-switching-from-go-to-rust

corrode

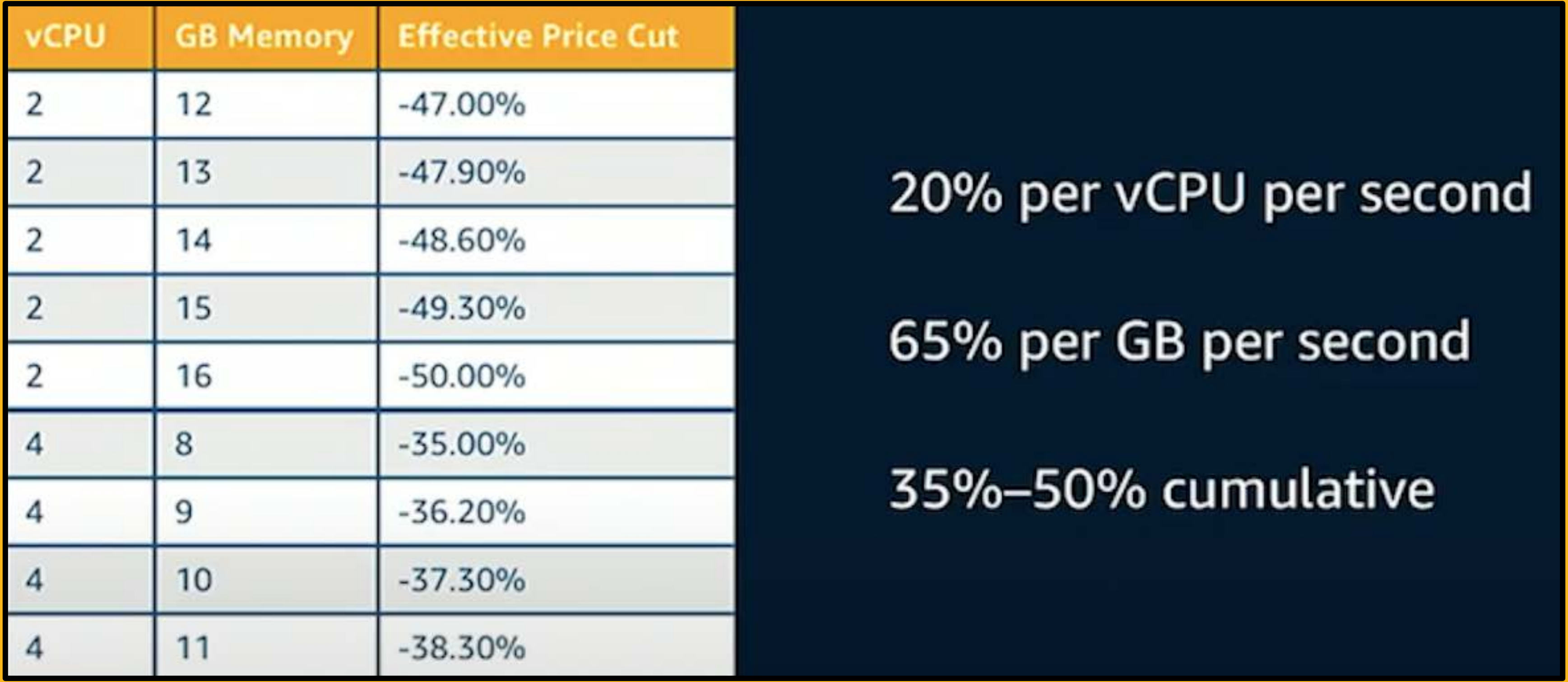
RUST VS GOLANG VS JAVA VS PYTHON



https://medium.com/star-gazers/benchmarking-low-level-i-o-c-c-rust-golang-java-python-9a0d505f85f7

corrode

AWS: RUST'S IMPACT ON SERVERLESS PRICING



vCPU	GB Memory	Effective Price Cut
2	12	-47.00%
2	13	-47.90%
2	14	-48.60%
2	15	-49.30%
2	16	-50.00%
4	8	-35.00%
4	9	-36.20%
4	10	-37.30%
4	11	-38.30%

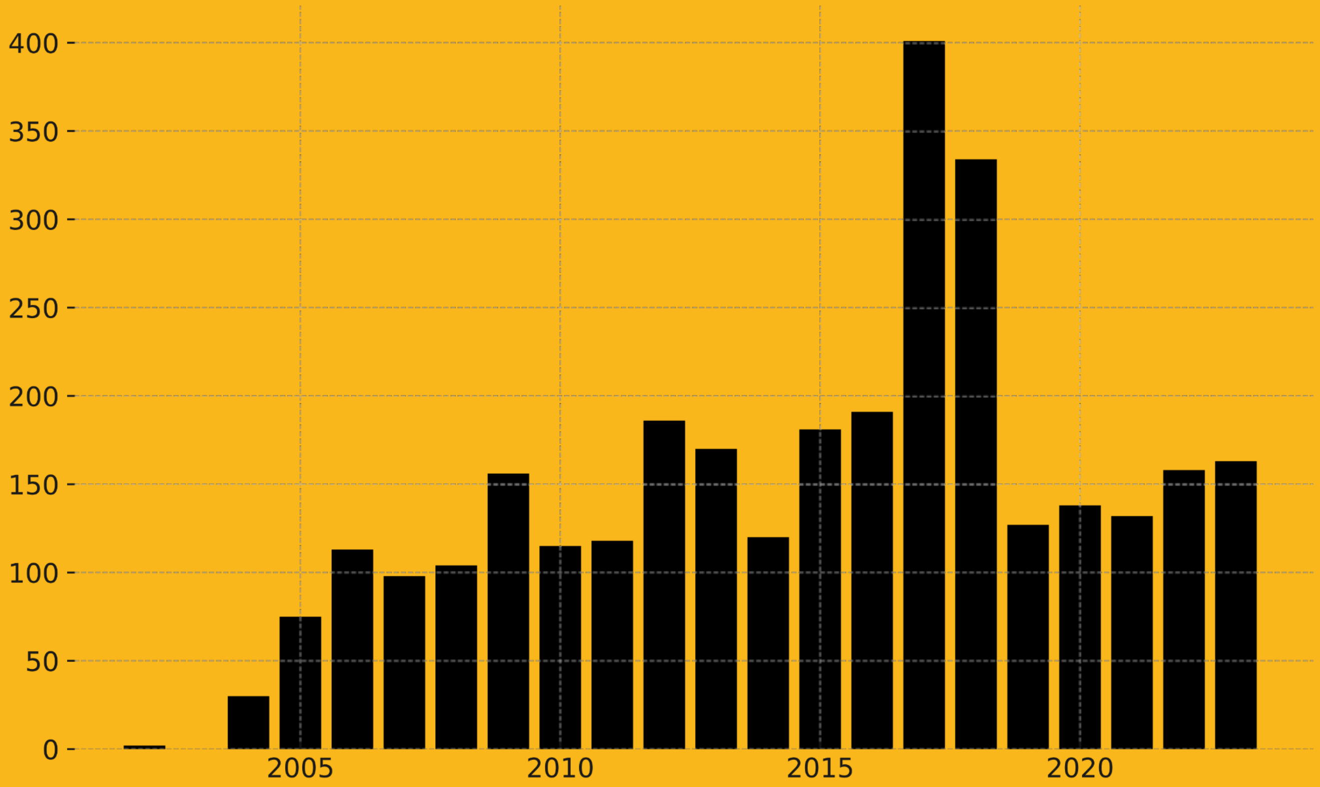
20% per vCPU per second

65% per GB per second

35%–50% cumulative

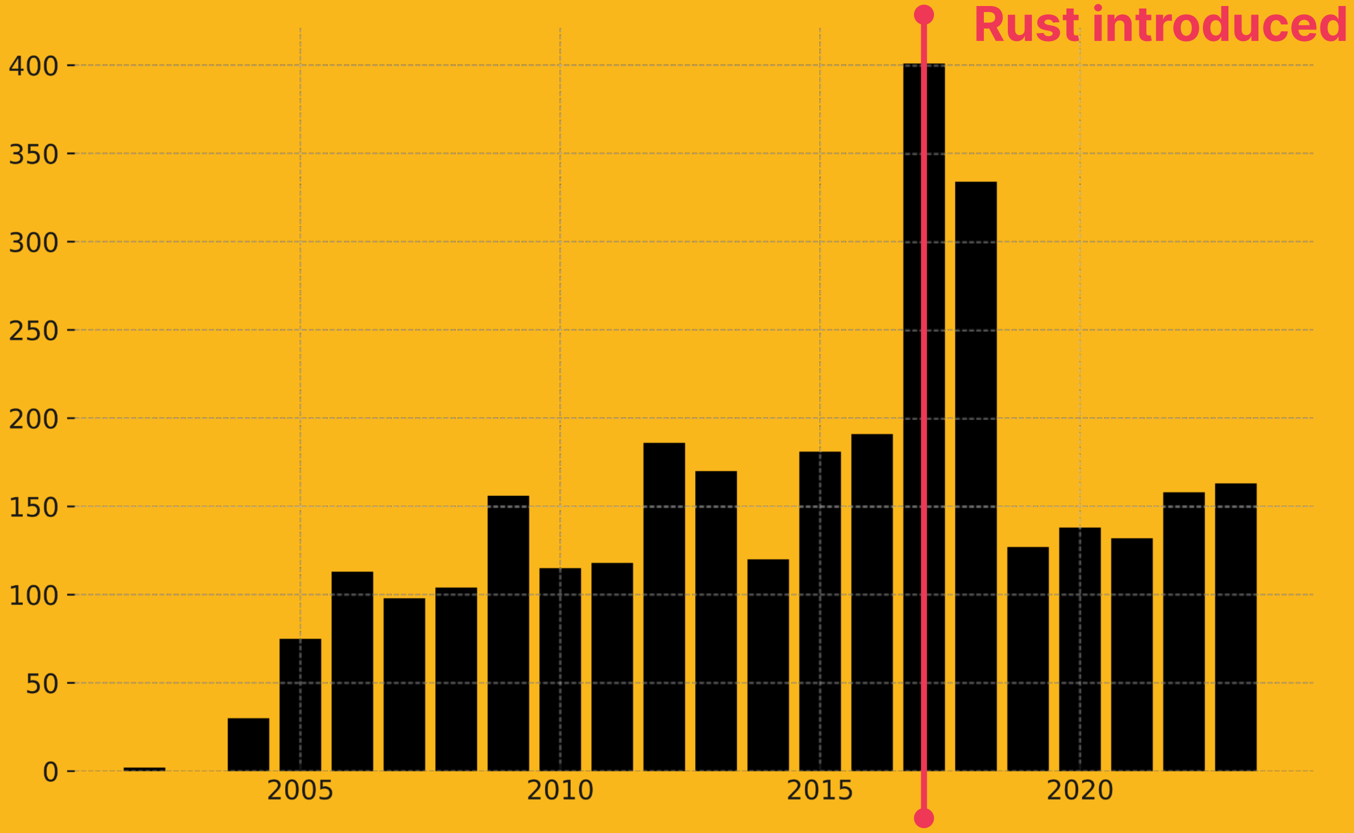
corrode

VULNERABILITIES IN FIREFOX OVER TIME

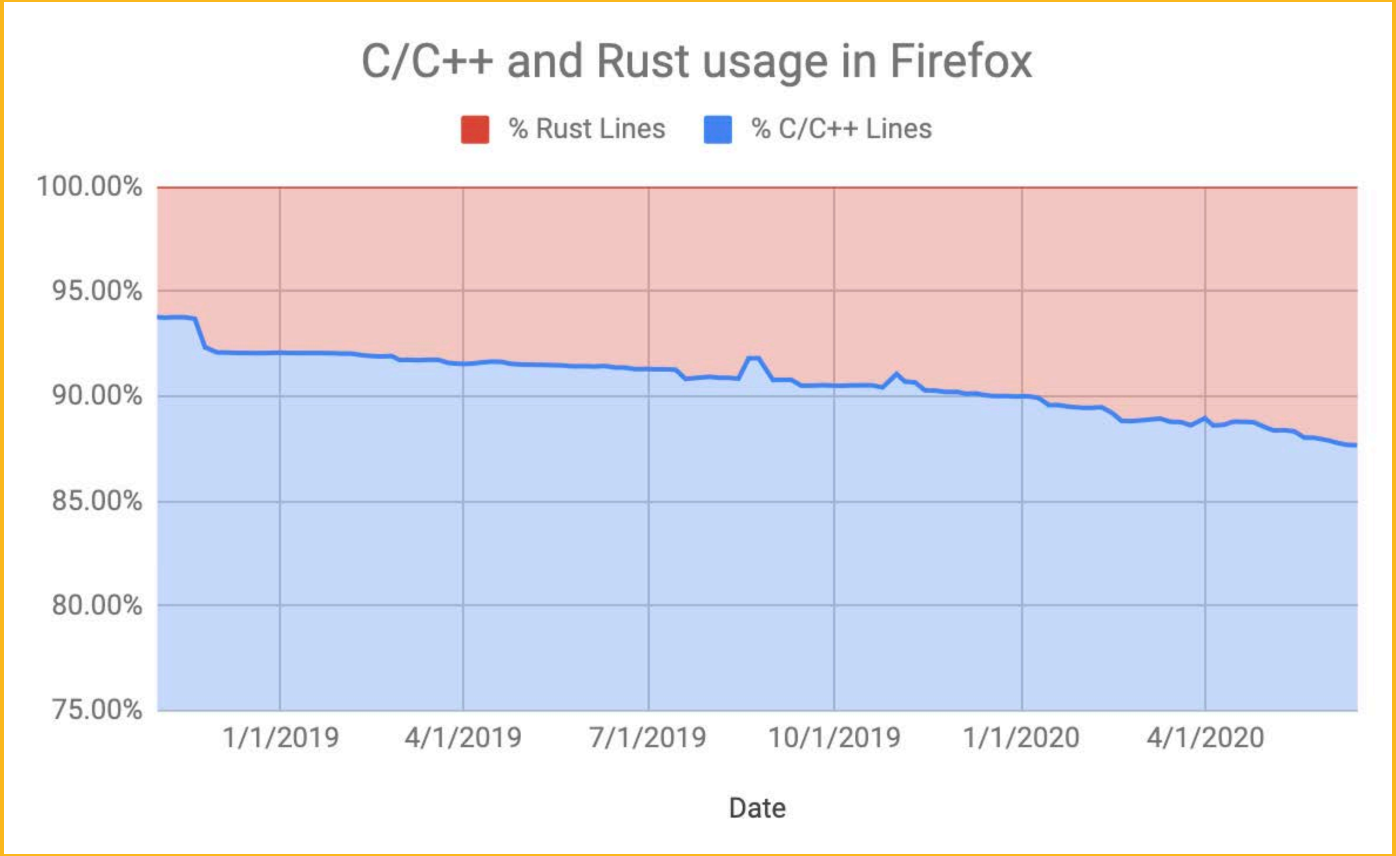


corrode

VULNERABILITIES IN FIREFOX OVER TIME



corrode



https://docs.google.com/spreadsheets/d/1flUGg6Ut4bjtyWdyH_9emD9EAN01ljTAVft2S4Dq620

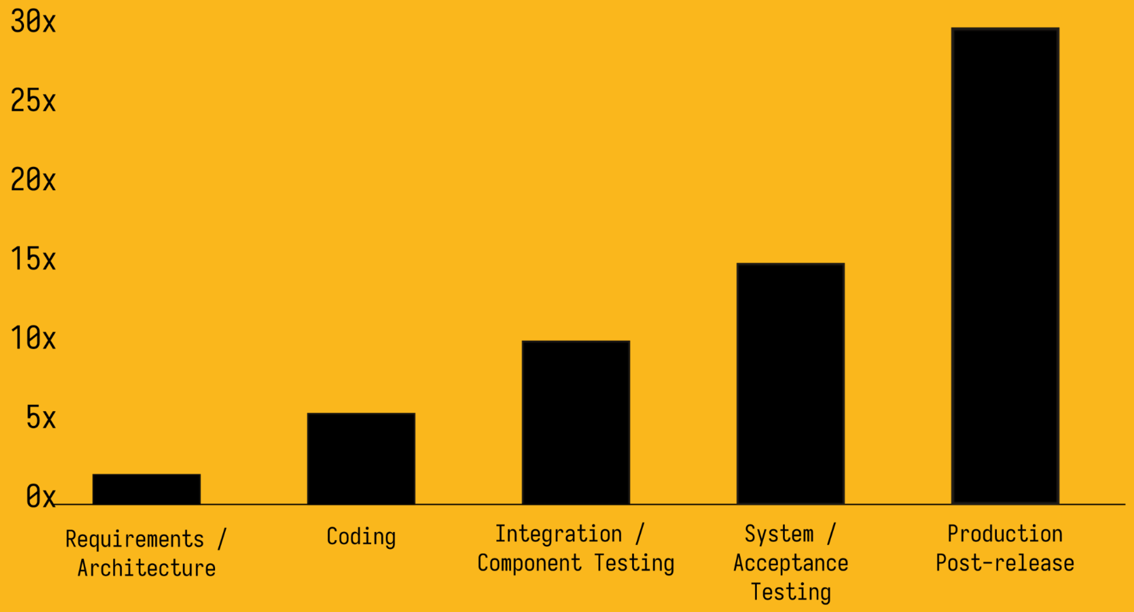
corrode

SECURITY ISSUES ARE A VERY REAL PROBLEM

- Google
  - Chromium project finds that around 70% of serious security bugs are memory safety problems
- Microsoft
  - 70% of bugs are memory safety issues
  - Each bug costs $150,000 to fix
  - >70 Million Dollars for fixing those bugs (in 2018)

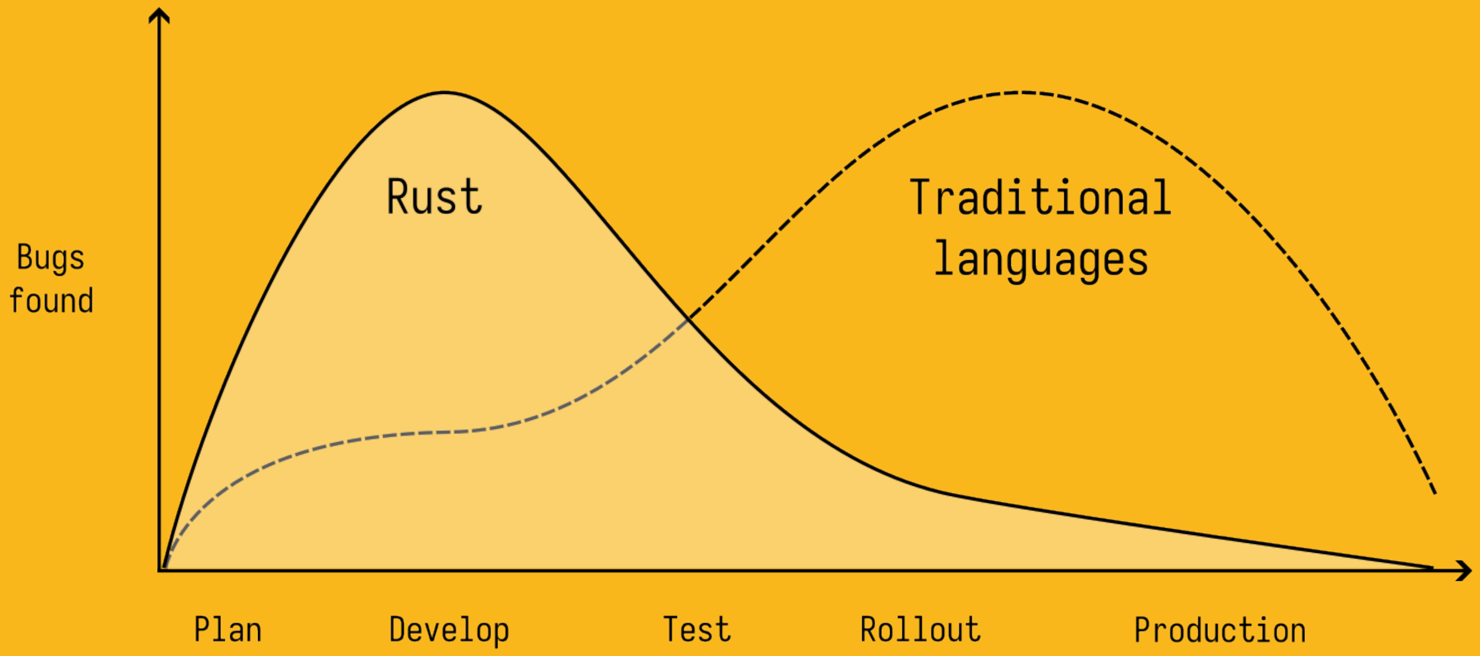
corrode

RELATIVE COST TO FIX BUGS BASED ON TIME OF DETECTION



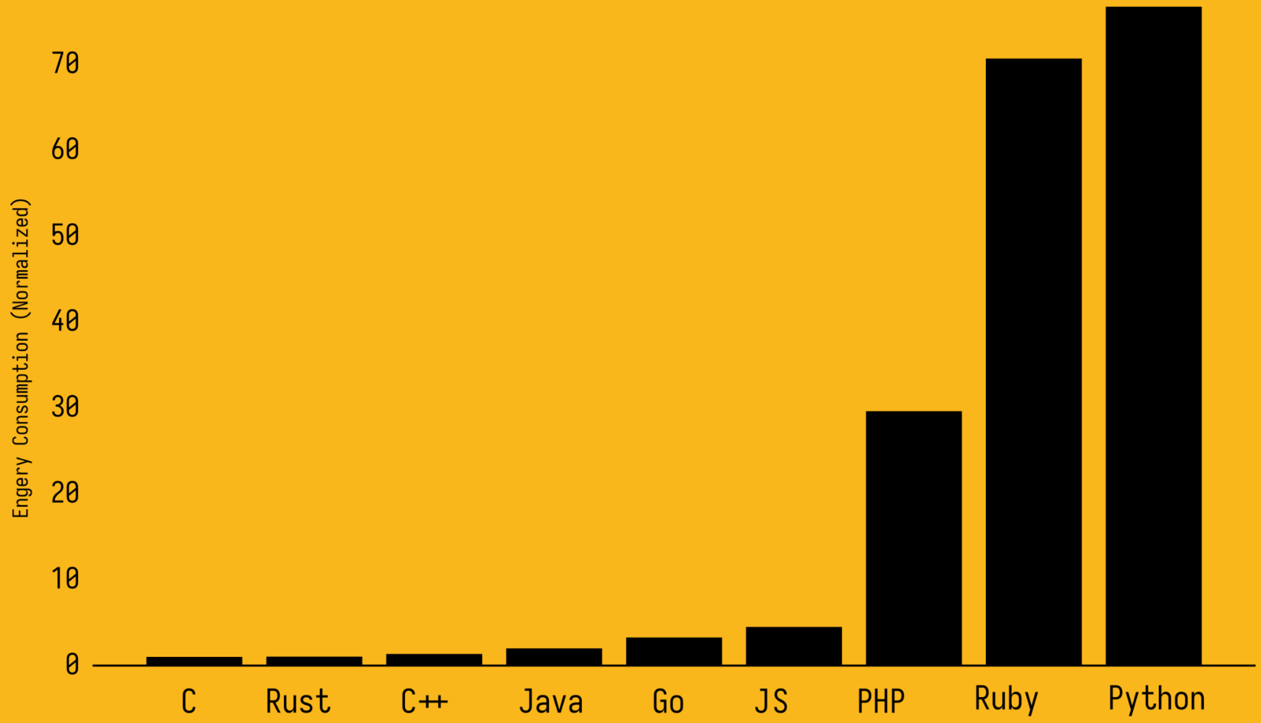
corrode

BUGS DETECTED DURING DEVELOPMENT CYCLE



corrode

ENERGY CONSUMPTION



corrode

RUST'S WEAKNESSES

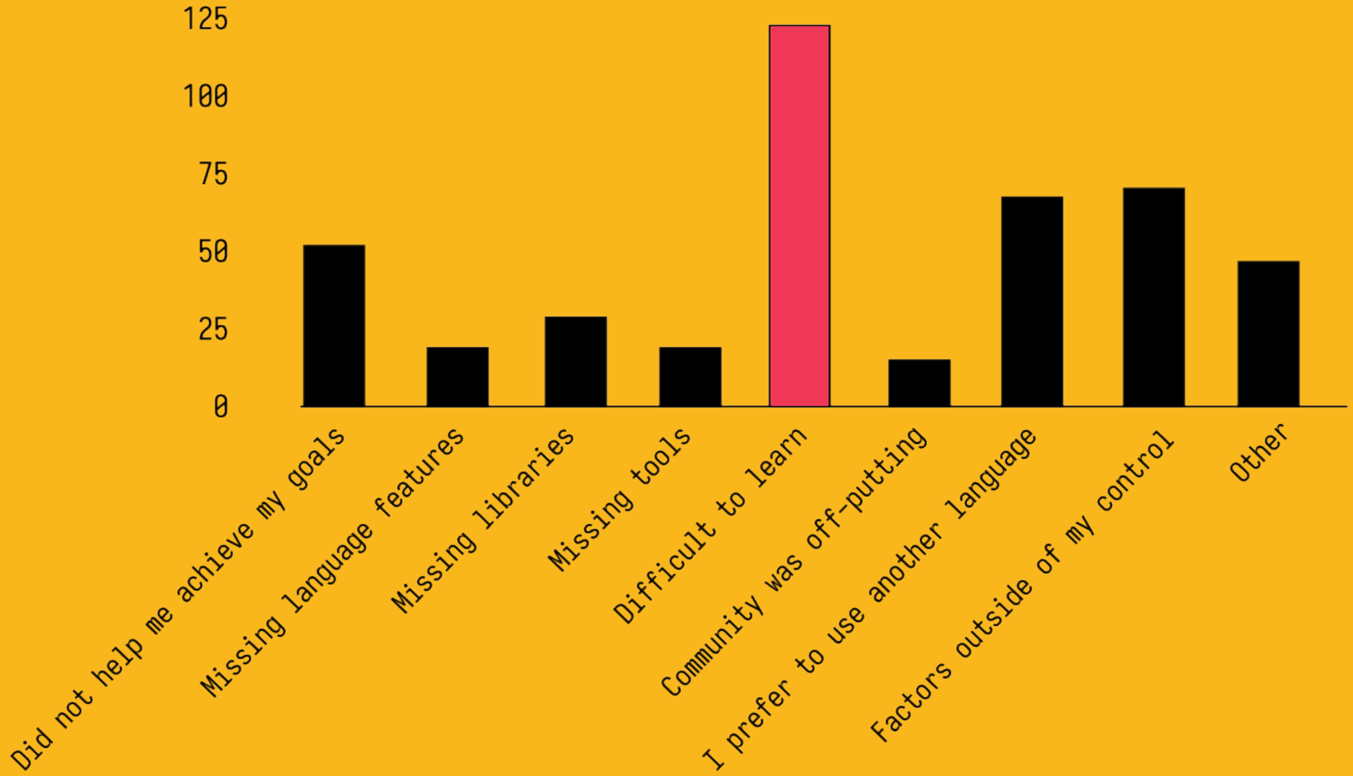
corrode

RUST WEAKNESSES

- Immature ecosystem
- Async/await support still very basic
- Lack of developers
- Learning curve
- Compile times

corrode

LEARNING CURVE



corrode

“

50% OF DEVELOPERS WERE PRODUCTIVE IN RUST AFTER 4 MONTHS

Google

corrode

“

IT TAKES
SEVERAL WEEKS
OF HARD EFFORT

Microsoft

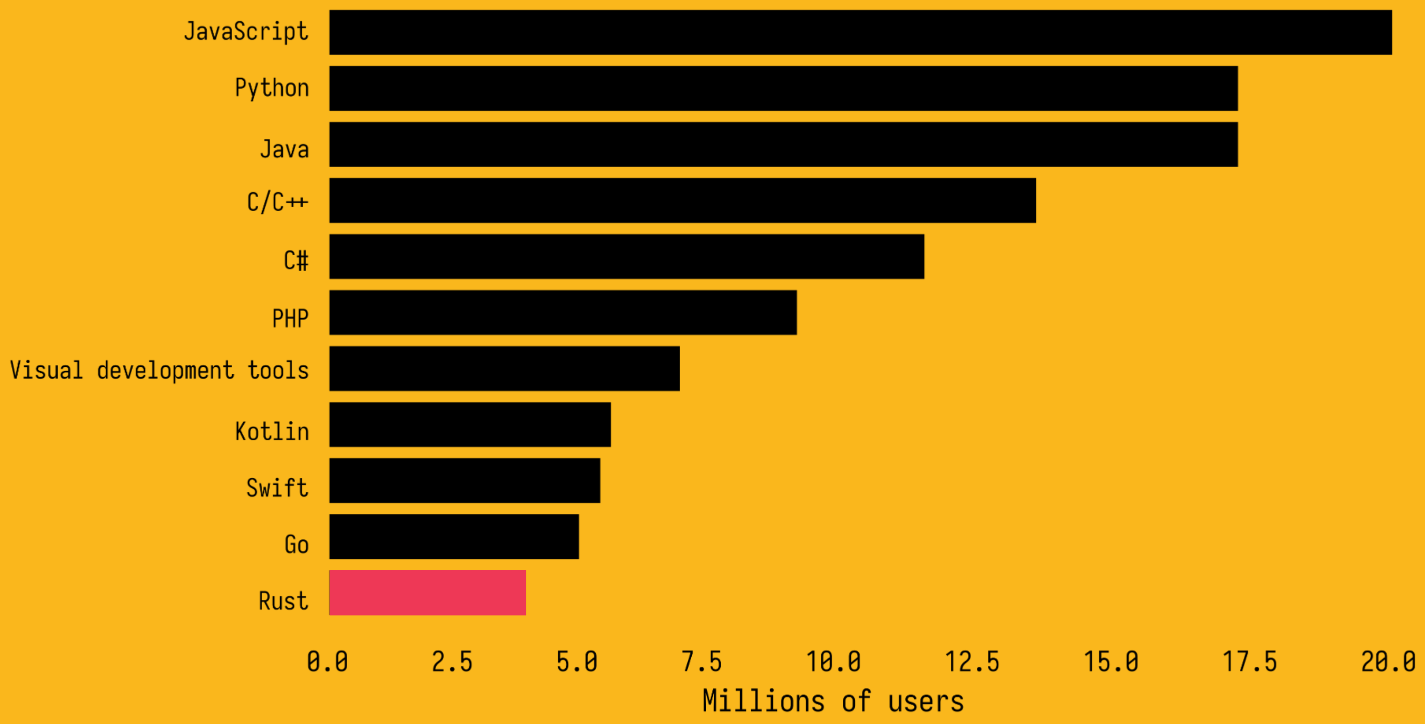
corrode

corrode

When your Rust program runs on the first try

corrode

SIZE OF PROGRAMMING COMMUNITIES 2023



corrode

RUST USERS

corrode

MAJOR RUST USAGE

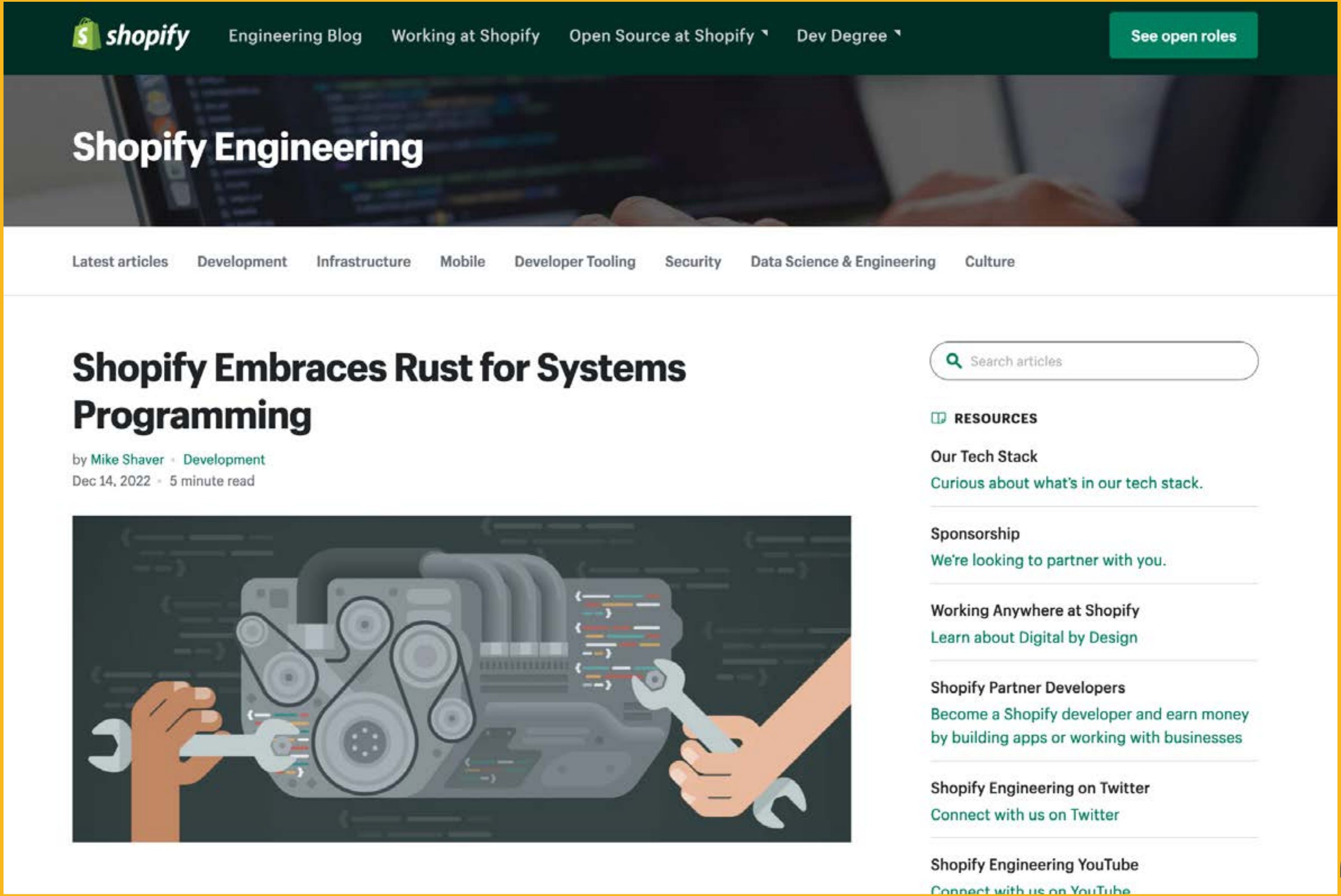
- Linux Kernel
- Windows Kernel
- AWS Firecracker
- Dropbox storage layer
- Deno
- Turbopack (Webpack)
- Figma
- Cloudflare

corrode

MAJOR RUST USAGE

- Linux Kernel
- Windows Kernel
- AWS Firecracker
- Dropbox storage layer
- Deno
- Turbopack (Webpack)
- Figma
- Cloudflare
- Yes, and Crypto

corrode



shopify Engineering Blog Working at Shopify Open Source at Shopify Dev Degree See open roles

Shopify Engineering

Latest articles Development Infrastructure Mobile Developer Tooling Security Data Science & Engineering Culture

Shopify Embraces Rust for Systems Programming

by Mike Shaver · Development
Dec 14, 2022 · 5 minute read

Search articles

RESOURCES

Our Tech Stack
Curious about what's in our tech stack.

Sponsorship
We're looking to partner with you.

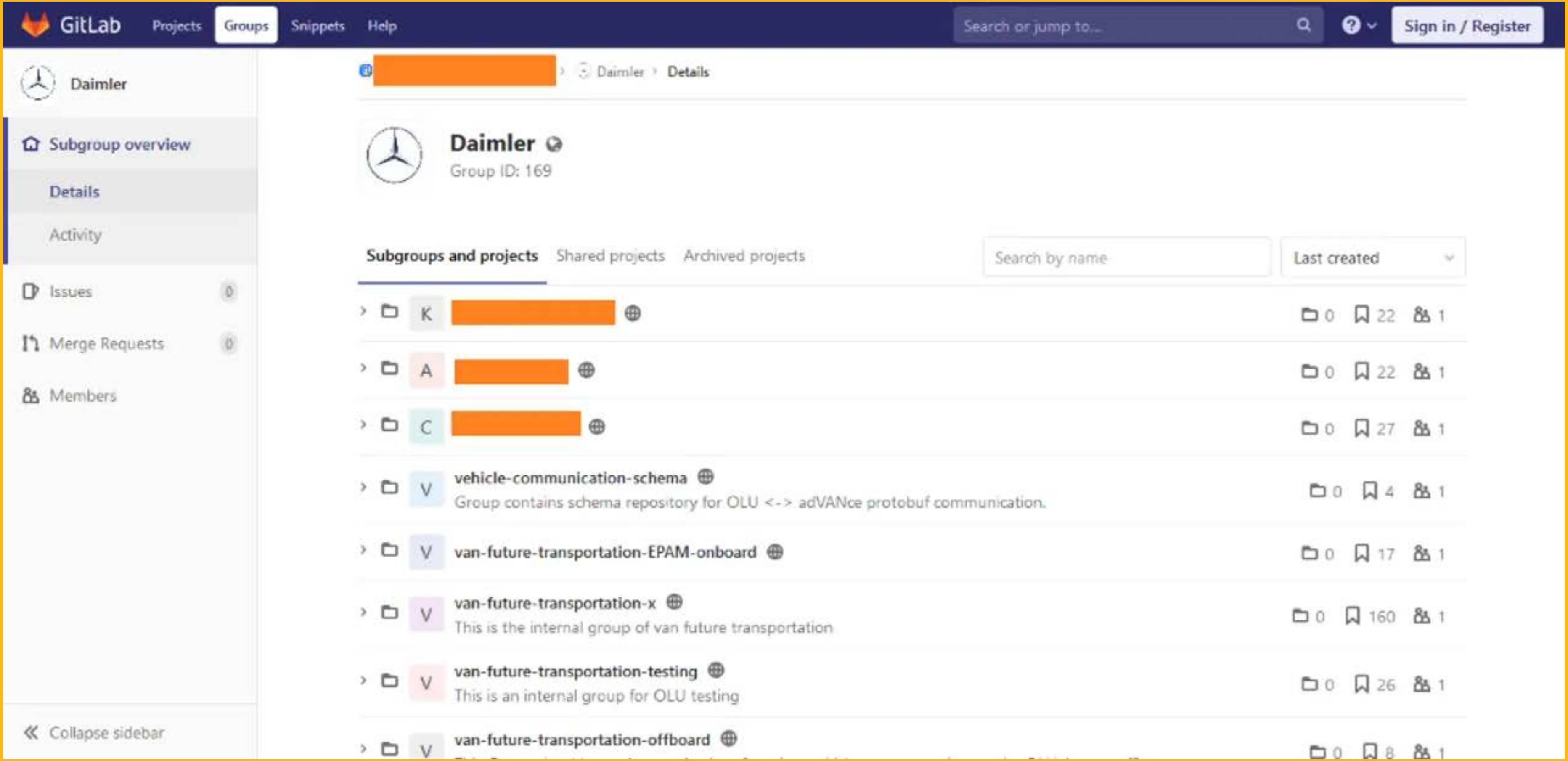
Working Anywhere at Shopify
Learn about Digital by Design

Shopify Partner Developers
Become a Shopify developer and earn money by building apps or working with businesses

Shopify Engineering on Twitter
Connect with us on Twitter

Shopify Engineering YouTube
Connect with us on YouTube

rrode



[redacted] › Daimler › Details

Daimler
Group ID: 169

Subgroups and projects | Shared projects | Archived projects

K [redacted] 0 22 1

A [redacted] 0 22 1

C [redacted] 0 27 1

V vehicle-communication-schema 0 4 1
Group contains schema repository for OLU <-> adVANce protobuf communication.

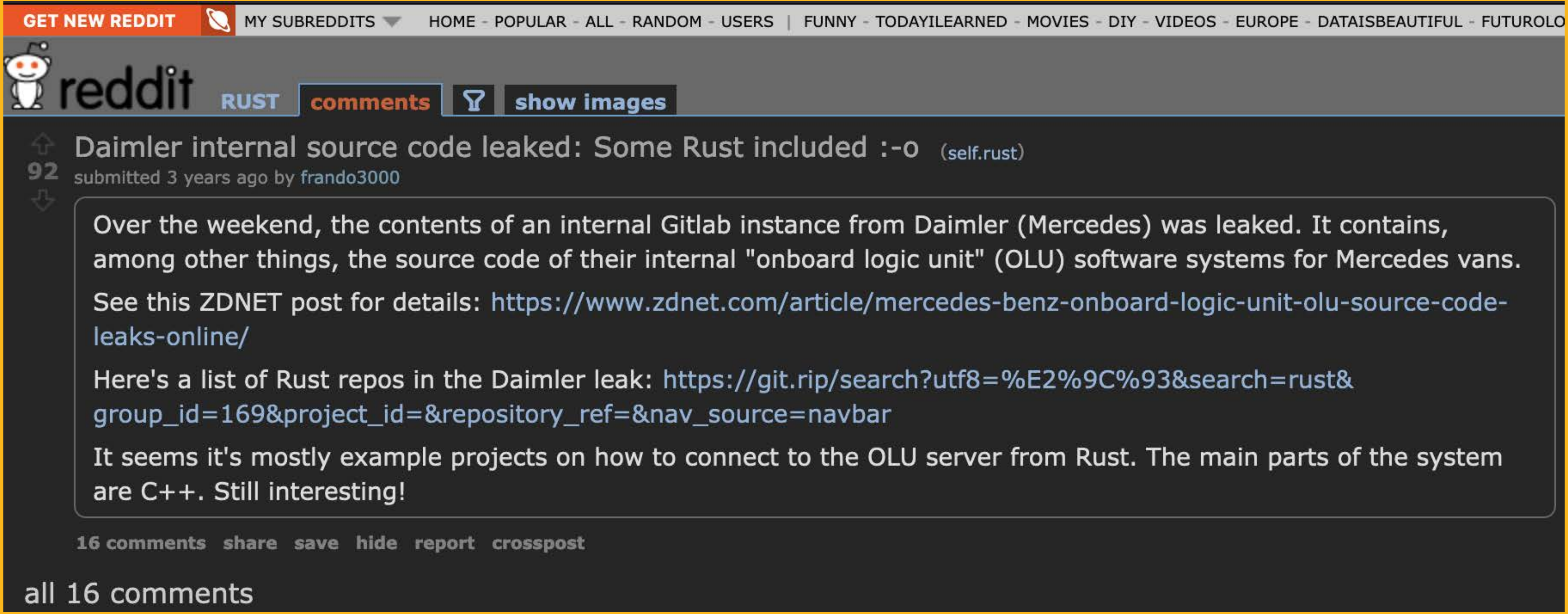
V van-future-transportation-EPAM-onboard 0 17 1

V van-future-transportation-x 0 160 1
This is the internal group of van future transportation

V van-future-transportation-testing 0 26 1
This is an internal group for OLU testing

V van-future-transportation-offboard 0 8 1

corrode


GET NEW REDDIT
MY SUBREDDITS
HOME - POPULAR - ALL - RANDOM - USERS | FUNNY - TODAYILEARNED - MOVIES - DIY - VIDEOS - EUROPE - DATAISBEAUTIFUL - FUTUROLO
reddit
RUST
comments
show images

92

Daimler internal source code leaked: Some Rust included :-o (self.rust)

submitted 3 years ago by frando3000

Over the weekend, the contents of an internal Gitlab instance from Daimler (Mercedes) was leaked. It contains, among other things, the source code of their internal "onboard logic unit" (OLU) software systems for Mercedes vans.

See this ZDNET post for details: https://www.zdnet.com/article/mercedes-benz-onboard-logic-unit-olu-source-code-leaks-online/

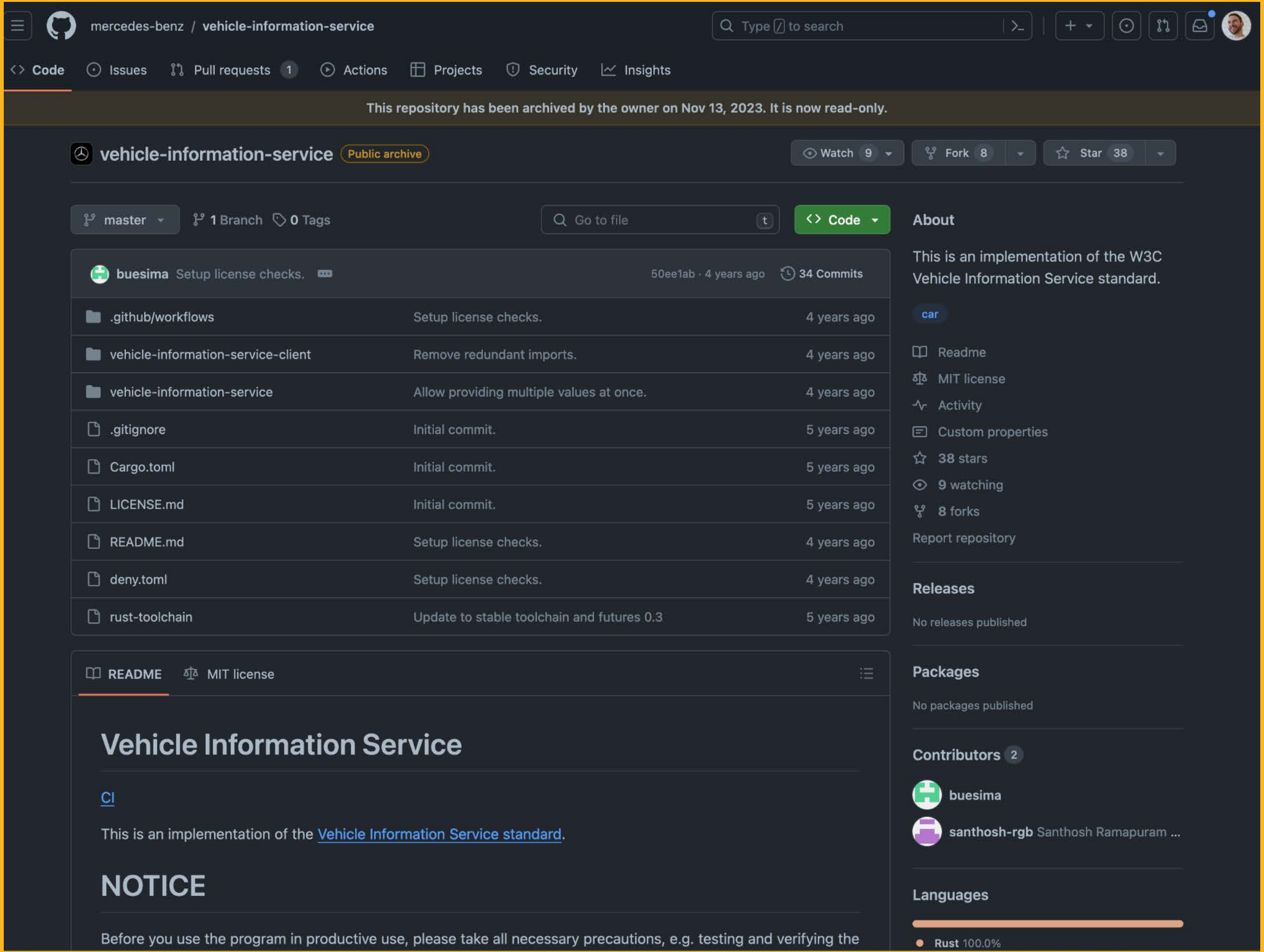
Here's a list of Rust repos in the Daimler leak: https://git.rip/search?utf8=%E2%9C%93&search=rust&group_id=169&project_id=&repository_ref=&nav_source=navbar

It seems it's mostly example projects on how to connect to the OLU server from Rust. The main parts of the system are C++. Still interesting!

16 comments share save hide report crosspost

all 16 comments

corrode


mercedes-benz / vehicle-information-service

This repository has been archived by the owner on Nov 13, 2023. It is now read-only.

vehicle-information-service Public archive

master · 1 Branch · 0 Tags

buesima Setup license checks. 50ee1ab · 4 years ago · 34 Commits

.github/workflows	Setup license checks.	4 years ago
vehicle-information-service-client	Remove redundant imports.	4 years ago
vehicle-information-service	Allow providing multiple values at once.	4 years ago
.gitignore	Initial commit.	5 years ago
Cargo.toml	Initial commit.	5 years ago
LICENSE.md	Initial commit.	5 years ago
README.md	Setup license checks.	4 years ago
deny.toml	Setup license checks.	4 years ago
rust-toolchain	Update to stable toolchain and futures 0.3	5 years ago

About

This is an implementation of the W3C Vehicle Information Service standard.

car

Readme · MIT license · Activity · Custom properties · 38 stars · 9 watching · 8 forks

Releases: No releases published

Packages: No packages published

Contributors 2: buesima, santhosh-rgb Santhosh Ramapuram ...

Languages: Rust 100.0%

Vehicle Information Service

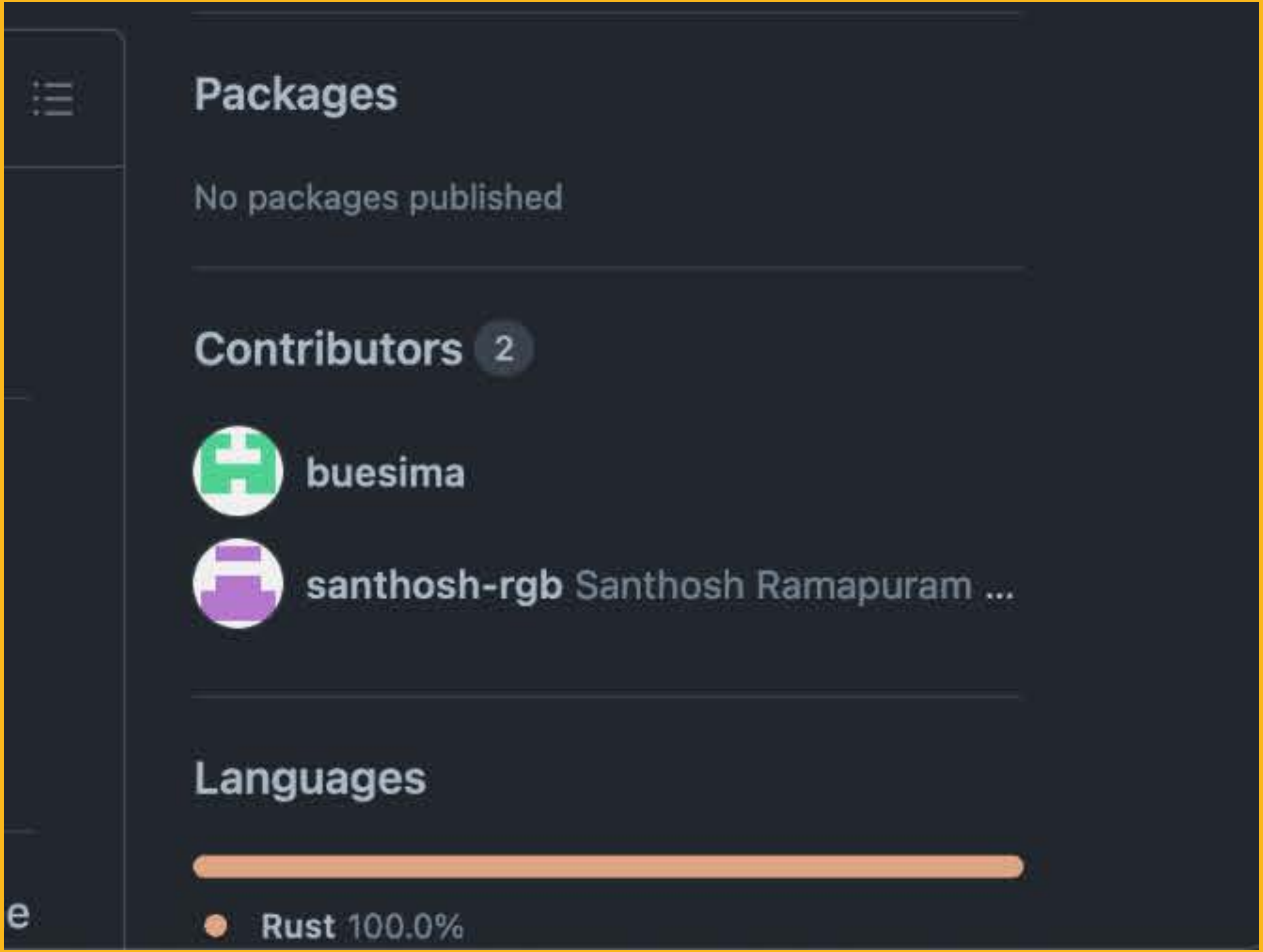
CI

This is an implementation of the Vehicle Information Service standard.

NOTICE

Before you use the program in productive use, please take all necessary precautions, e.g. testing and verifying the

corrode


Packages

No packages published

Contributors 2

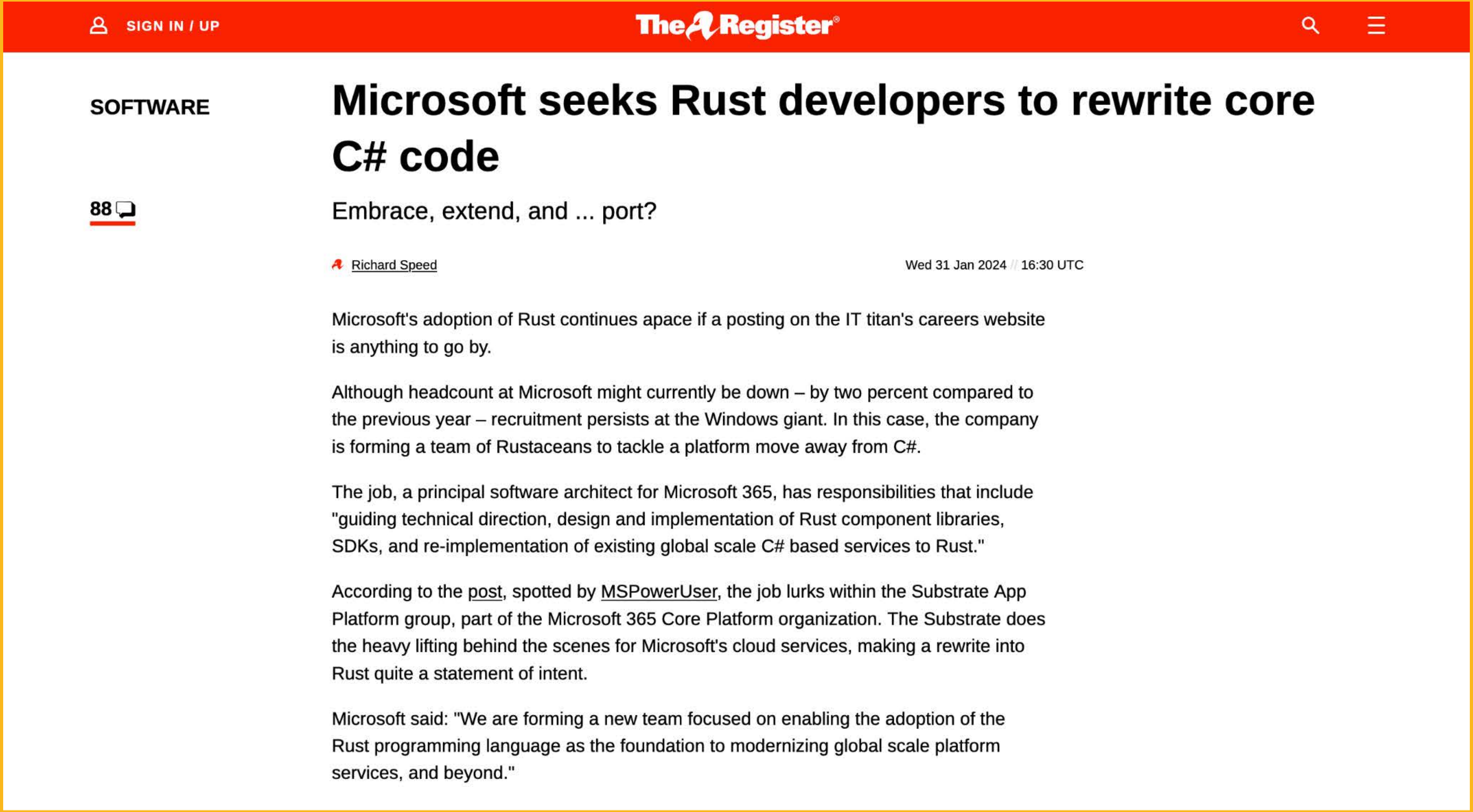
buesima

santhosh-rgb Santhosh Ramapuram ...

Languages

● Rust 100.0%

corrode



SOFTWARE

88

Microsoft seeks Rust developers to rewrite core C# code

Embrace, extend, and ... port?

Richard Speed Wed 31 Jan 2024 // 16:30 UTC

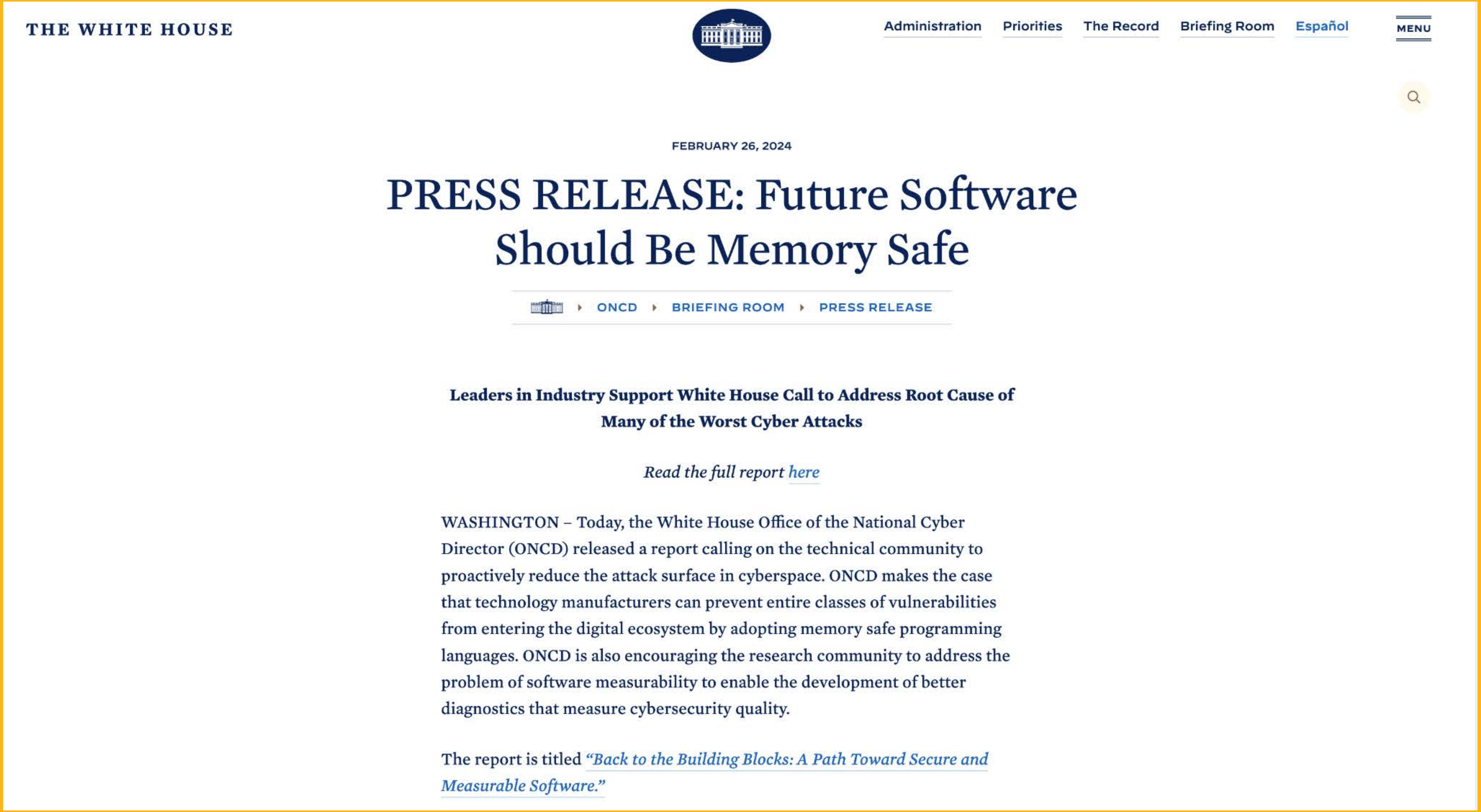
Microsoft's adoption of Rust continues apace if a posting on the IT titan's careers website is anything to go by.

Although headcount at Microsoft might currently be down – by two percent compared to the previous year – recruitment persists at the Windows giant. In this case, the company is forming a team of Rustaceans to tackle a platform move away from C#.

The job, a principal software architect for Microsoft 365, has responsibilities that include "guiding technical direction, design and implementation of Rust component libraries, SDKs, and re-implementation of existing global scale C# based services to Rust."

According to the post, spotted by MSPowerUser, the job lurks within the Substrate App Platform group, part of the Microsoft 365 Core Platform organization. The Substrate does the heavy lifting behind the scenes for Microsoft's cloud services, making a rewrite into Rust quite a statement of intent.

Microsoft said: "We are forming a new team focused on enabling the adoption of the Rust programming language as the foundation to modernizing global scale platform services, and beyond."



FEBRUARY 26, 2024

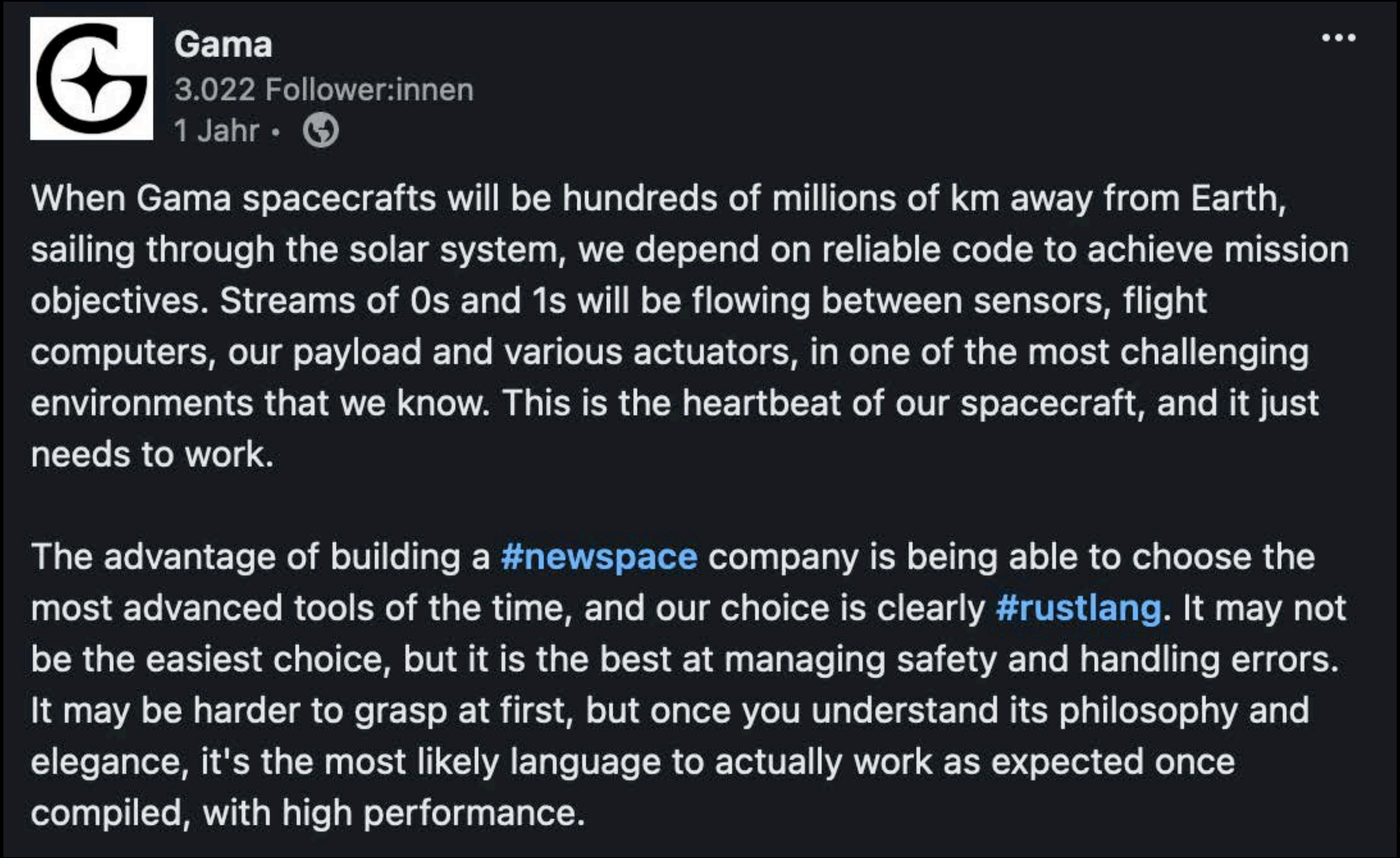
PRESS RELEASE: Future Software Should Be Memory Safe

Leaders in Industry Support White House Call to Address Root Cause of Many of the Worst Cyber Attacks

Read the full report here

WASHINGTON – Today, the White House Office of the National Cyber Director (ONCD) released a report calling on the technical community to proactively reduce the attack surface in cyberspace. ONCD makes the case that technology manufacturers can prevent entire classes of vulnerabilities from entering the digital ecosystem by adopting memory safe programming languages. ONCD is also encouraging the research community to address the problem of software measurability to enable the development of better diagnostics that measure cybersecurity quality.

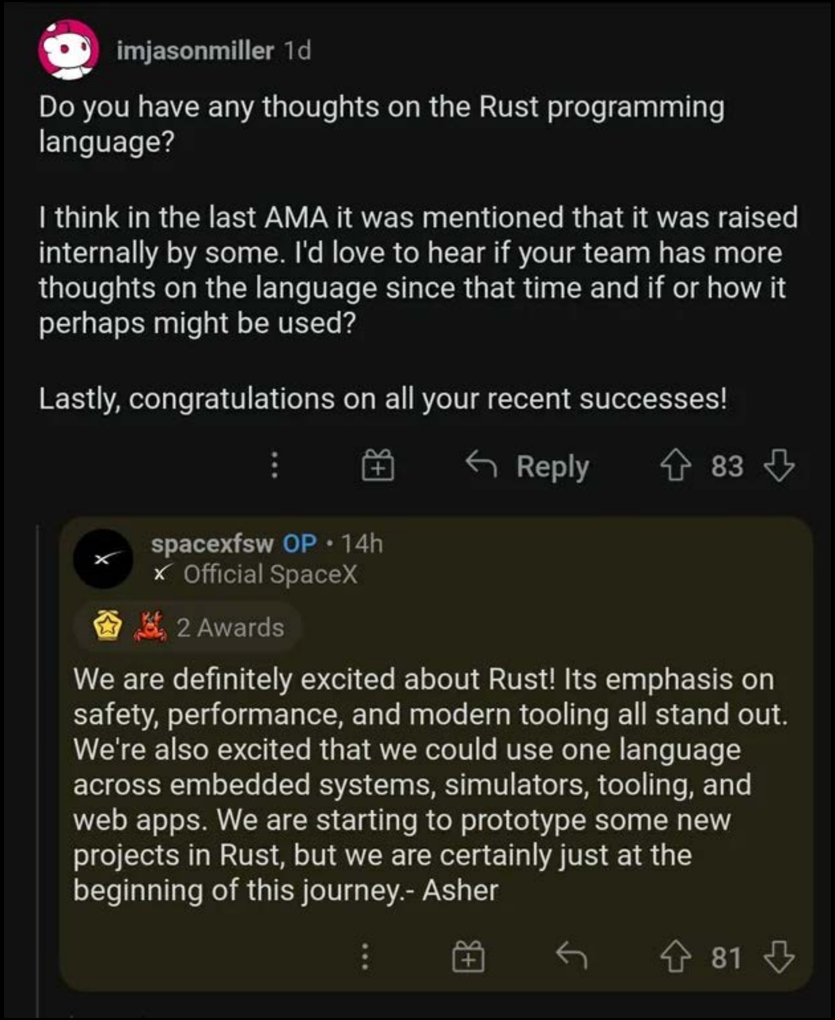
The report is titled *"Back to the Building Blocks: A Path Toward Secure and Measurable Software."*


Gama
3.022 Follower:innen
1 Jahr •

When Gama spacecrafts will be hundreds of millions of km away from Earth, sailing through the solar system, we depend on reliable code to achieve mission objectives. Streams of 0s and 1s will be flowing between sensors, flight computers, our payload and various actuators, in one of the most challenging environments that we know. This is the heartbeat of our spacecraft, and it just needs to work.

The advantage of building a #newspace company is being able to choose the most advanced tools of the time, and our choice is clearly #rustlang. It may not be the easiest choice, but it is the best at managing safety and handling errors. It may be harder to grasp at first, but once you understand its philosophy and elegance, it's the most likely language to actually work as expected once compiled, with high performance.

https://twitter.com/gamaspace



imjasonmiller 1d

Do you have any thoughts on the Rust programming language?

I think in the last AMA it was mentioned that it was raised internally by some. I'd love to hear if your team has more thoughts on the language since that time and if or how it perhaps might be used?

Lastly, congratulations on all your recent successes!

spacexfsw OP • 14h
Official SpaceX

2 Awards

We are definitely excited about Rust! Its emphasis on safety, performance, and modern tooling all stand out. We're also excited that we could use one language across embedded systems, simulators, tooling, and web apps. We are starting to prototype some new projects in Rust, but we are certainly just at the beginning of this journey.- Asher

You RN

A CASE FOR YOUR COMPANY?

corrode

STRATEGIES FOR RUST ADOPTION

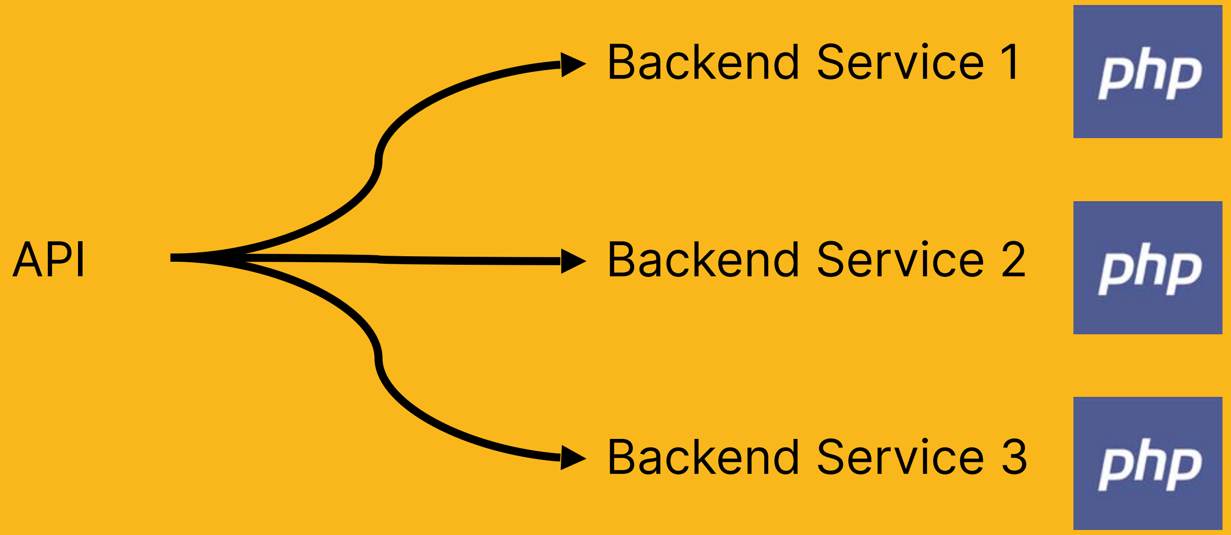
corrode

POPULAR WAYS OF RUST INTEGRATION

- Network APIs
  - Microservices
  - GraphQL
- Foreign-Function Interface (FFI)
  - Java
  - Python
  - C++
- WebAssembly
  - Frontend
  - Plugin-systems

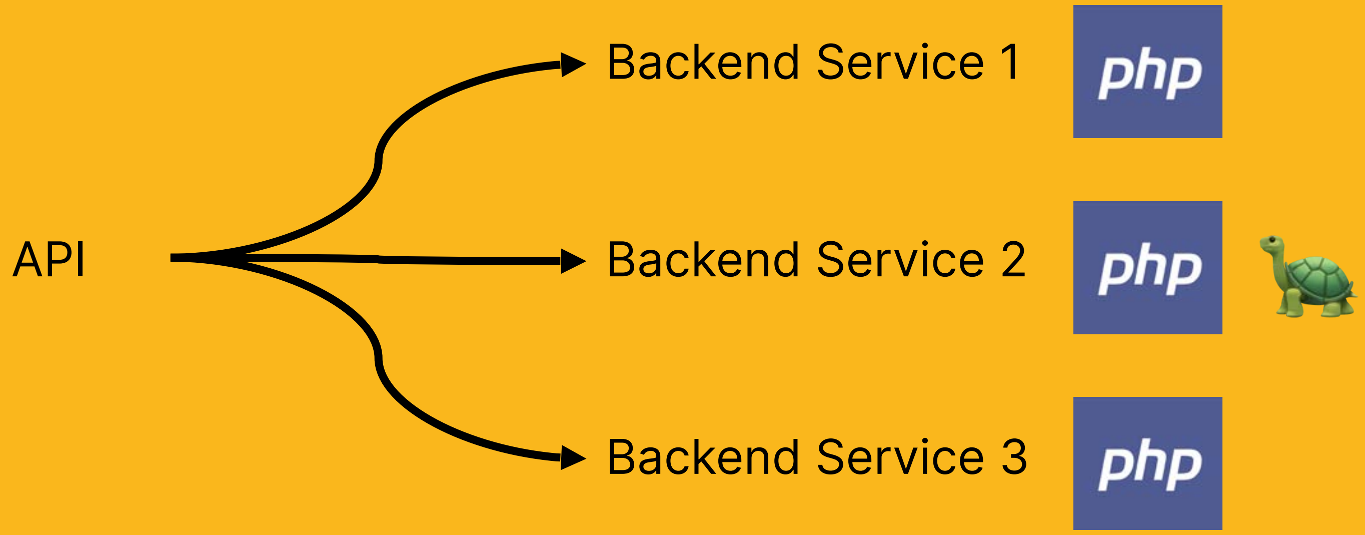
corrode

RUST ADOPTION IN NETWORK APIS



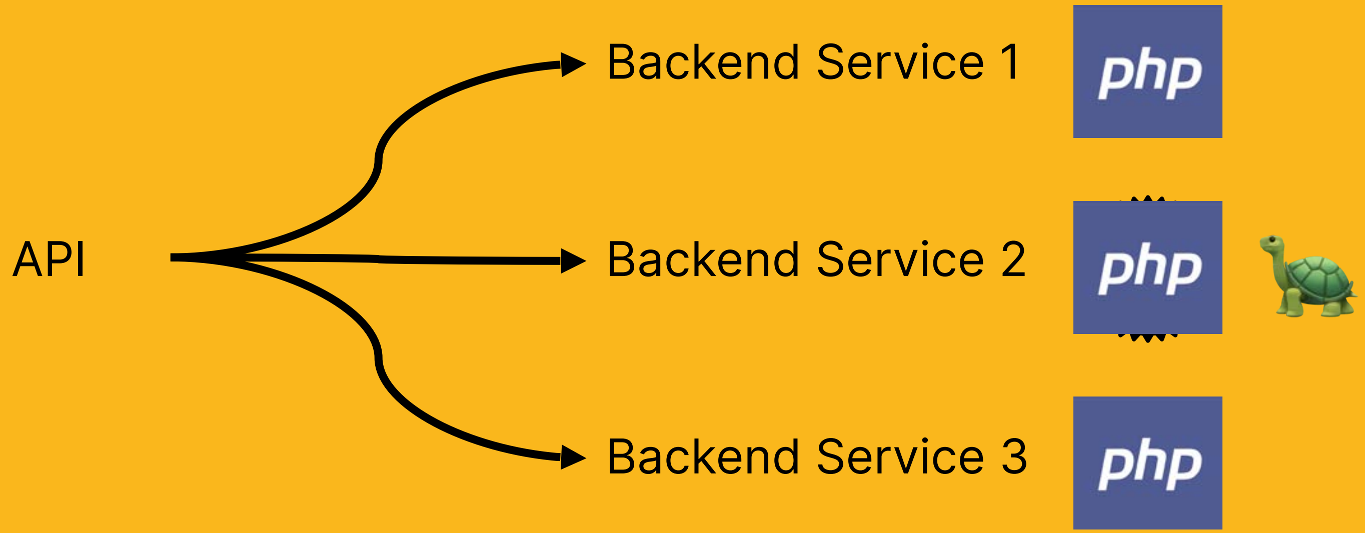
corrode

RUST ADOPTION IN NETWORK APIS



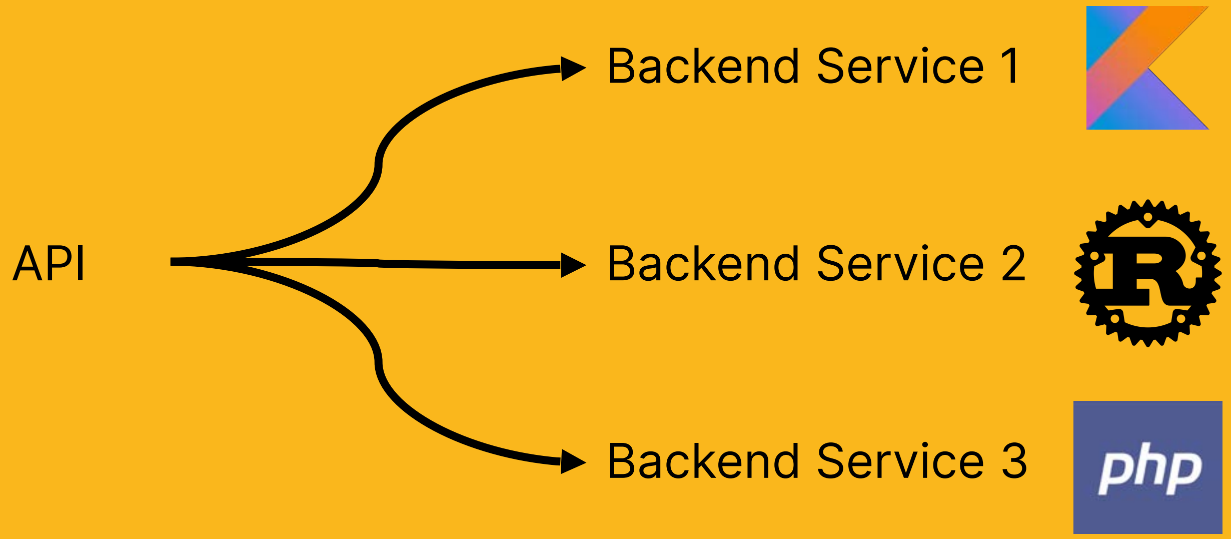
corrode

RUST ADOPTION IN NETWORK APIS



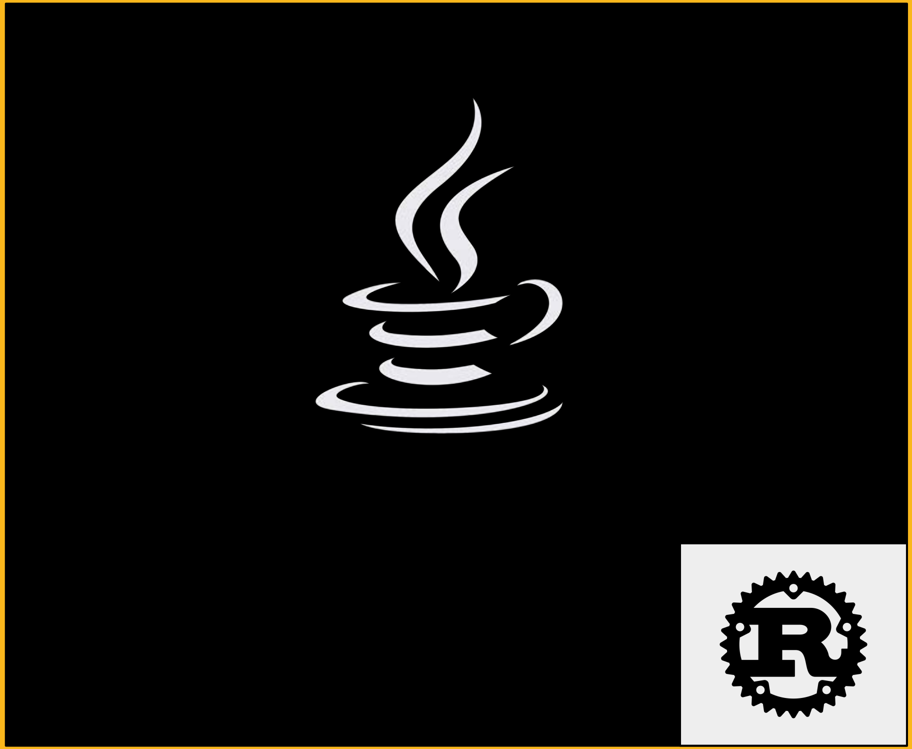
corrode

RUST ADOPTION IN NETWORK APIS



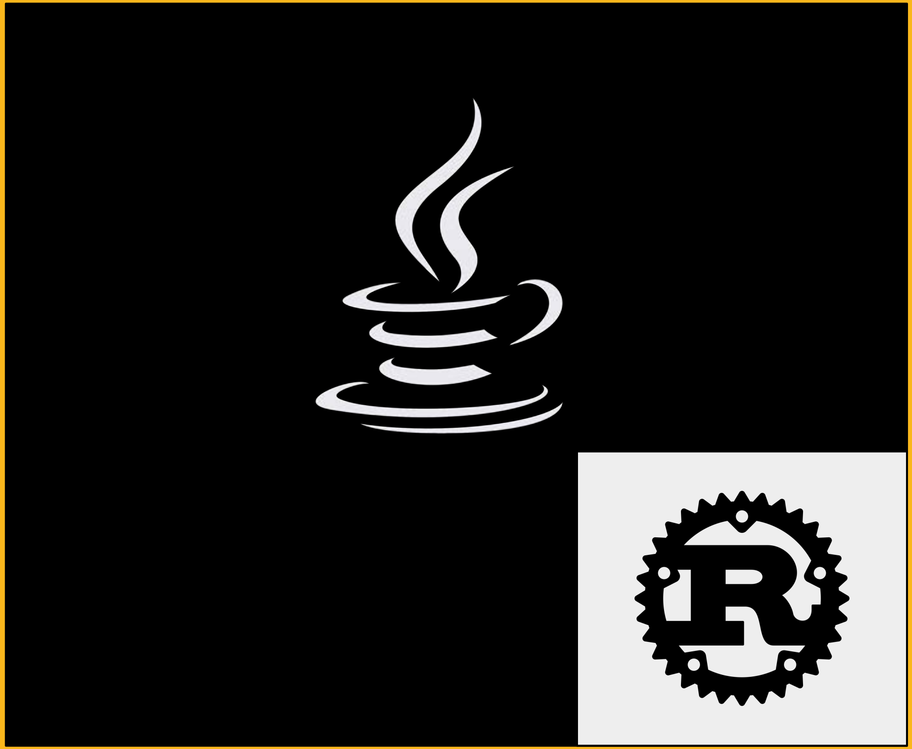
RUST ADOPTION IN MONOLITHS

RUST ADOPTION IN MONOLITHS

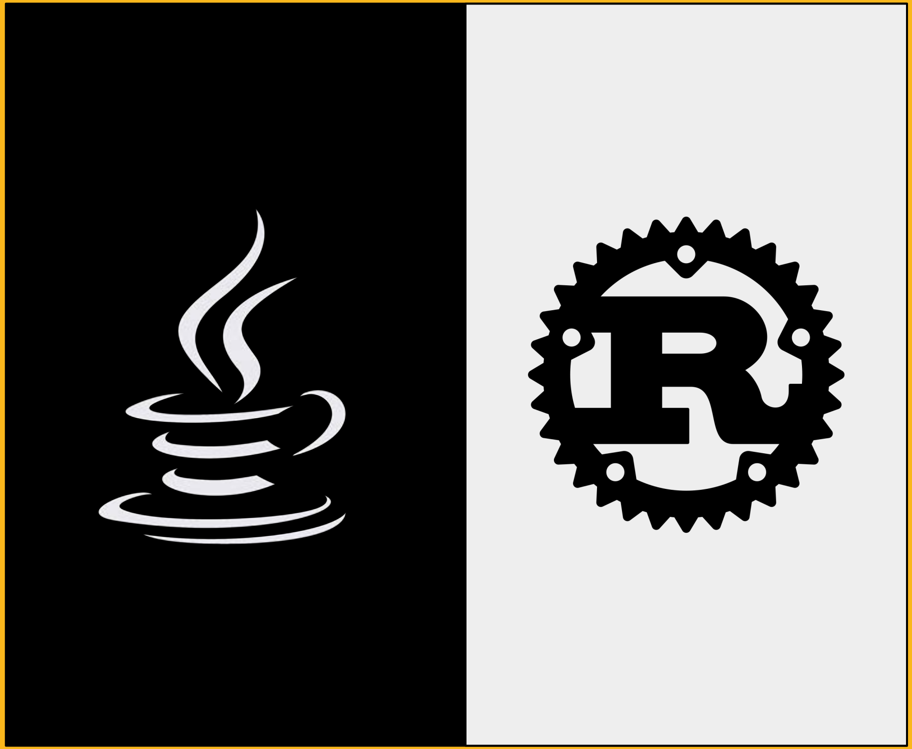


corrode

RUST ADOPTION IN MONOLITHS



RUST ADOPTION IN MONOLITHS



corrode

Vortexa & Rust, our Journey

JAVA VS RUST

Source corrode

JAVA

⏱️ 7 days

🤑 $1100/run

VS

RUST

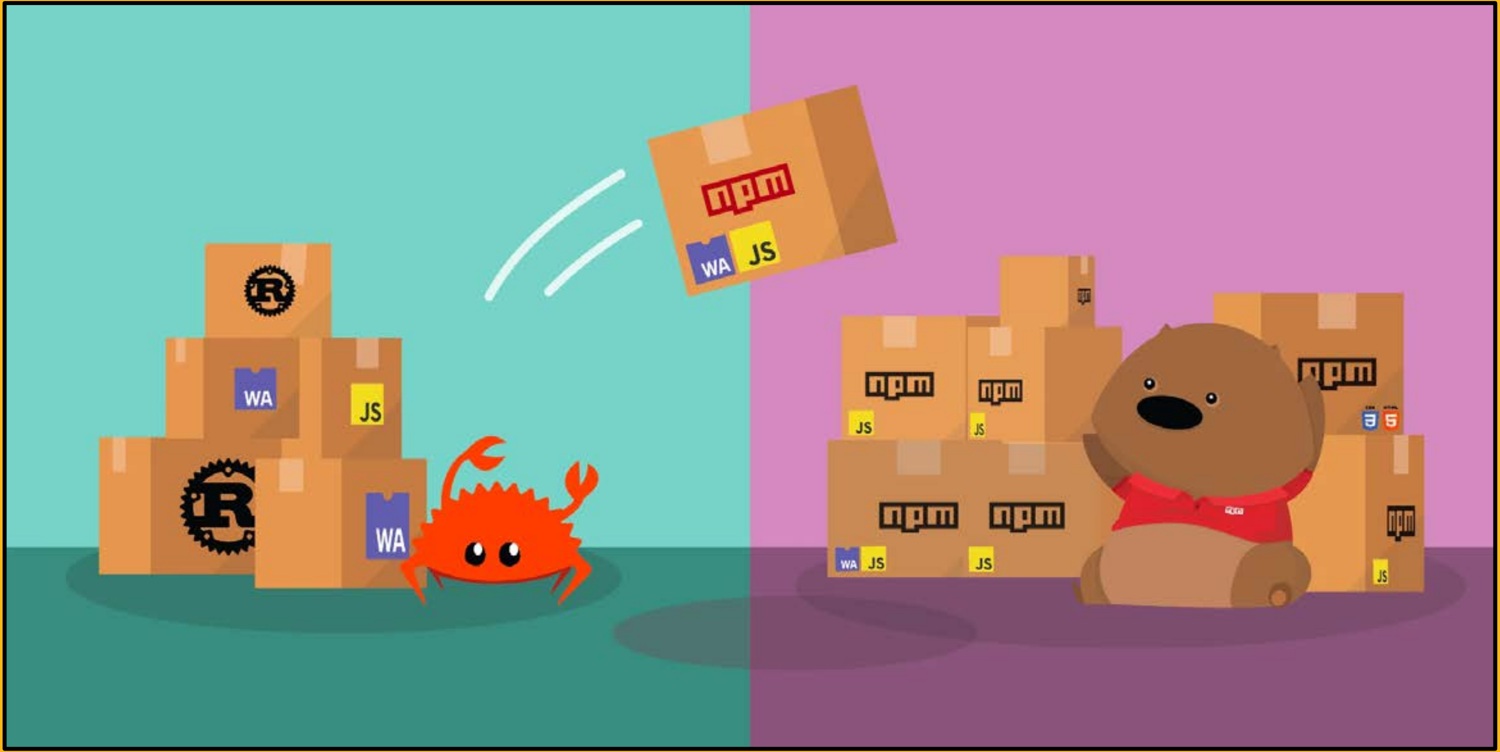
⏱️ 1 day

🤑 $80/run

Source corrode

RUST ADOPTION IN HS-WEB-APP

RUST INTEGRATION IN FRONTENDS / JAVASCRIPT



corrode

REQUIREMENTS FOR ADOPTION

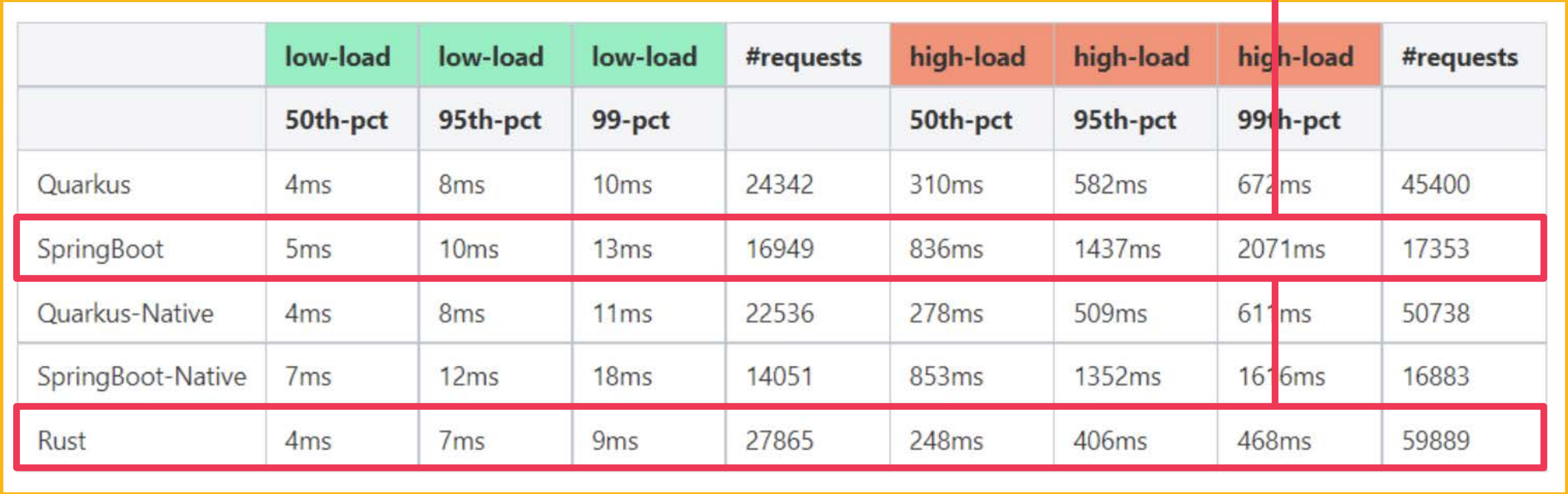
corrode

RUST ADOPTION NEEDS A CATALYST FOR SUCCESS

corrode

PERFORMANCE IS A WEAK CATALYST

4X PERFORMANCE BOOST



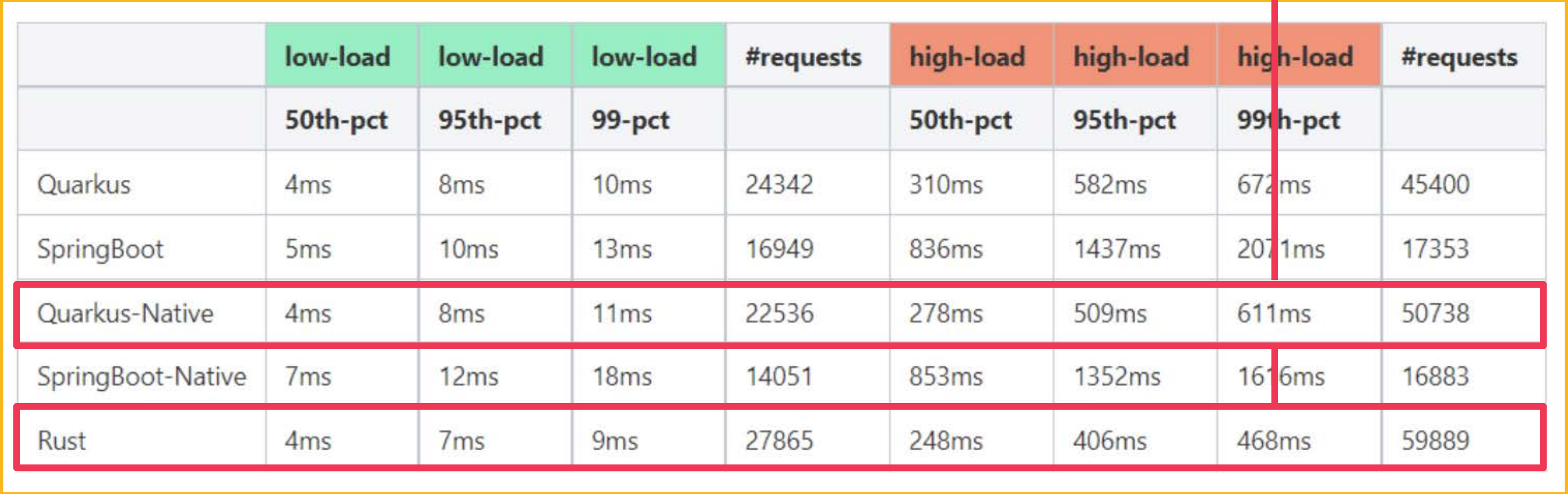
	low-load	low-load	low-load	#requests	high-load	high-load	high-load	#requests
	50th-pct	95th-pct	99-pct		50th-pct	95th-pct	99th-pct	
Quarkus	4ms	8ms	10ms	24342	310ms	582ms	672ms	45400
SpringBoot	5ms	10ms	13ms	16949	836ms	1437ms	2071ms	17353
Quarkus-Native	4ms	8ms	11ms	22536	278ms	509ms	611ms	50738
SpringBoot-Native	7ms	12ms	18ms	14051	853ms	1352ms	1616ms	16883
Rust	4ms	7ms	9ms	27865	248ms	406ms	468ms	59889

https://blog.consol.de/software-engineering/web-application-development/rust-vs-quarkus-native-vs-spring-native/

corrode

PERFORMANCE IS A WEAK CATALYST

1.5X PERFORMANCE BOOST

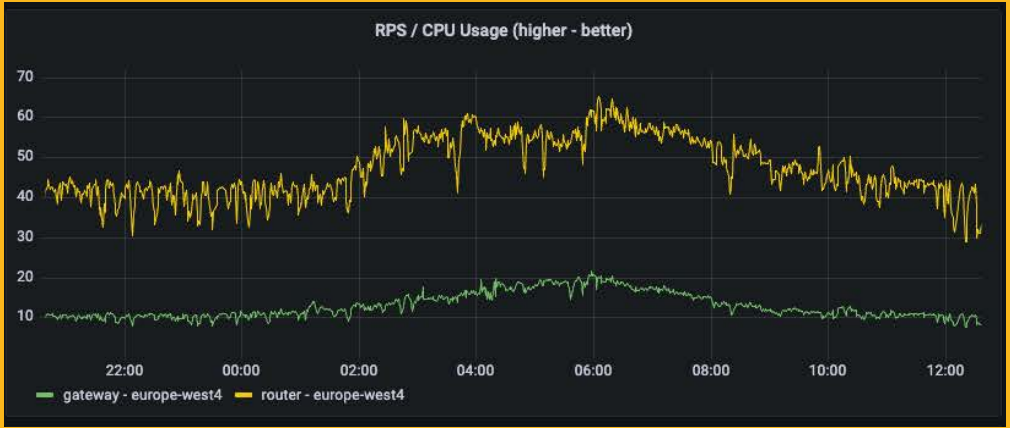


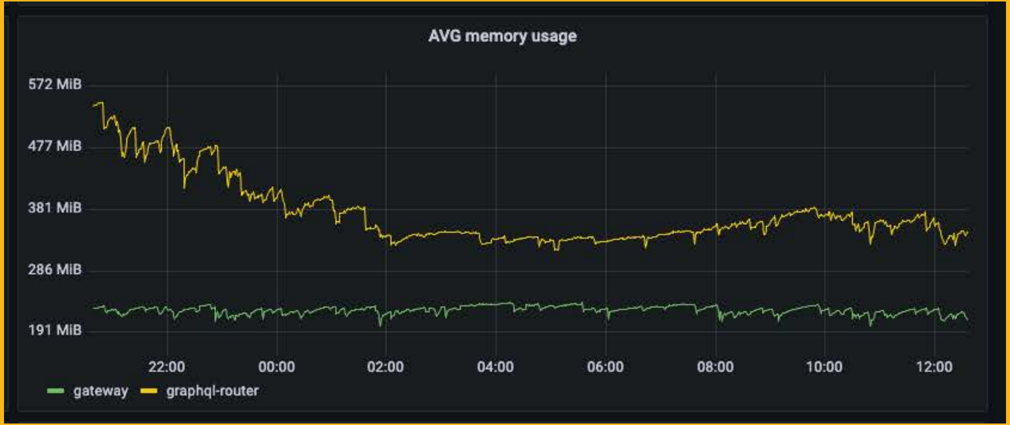
	low-load	low-load	low-load	#requests	high-load	high-load	high-load	#requests
	50th-pct	95th-pct	99-pct		50th-pct	95th-pct	99th-pct	
Quarkus	4ms	8ms	10ms	24342	310ms	582ms	672ms	45400
SpringBoot	5ms	10ms	13ms	16949	836ms	1437ms	2071ms	17353
Quarkus-Native	4ms	8ms	11ms	22536	278ms	509ms	611ms	50738
SpringBoot-Native	7ms	12ms	18ms	14051	853ms	1352ms	1616ms	16883
Rust	4ms	7ms	9ms	27865	248ms	406ms	468ms	59889

https://blog.consol.de/software-engineering/web-application-development/rust-vs-quarkus-native-vs-spring-native/

corrode

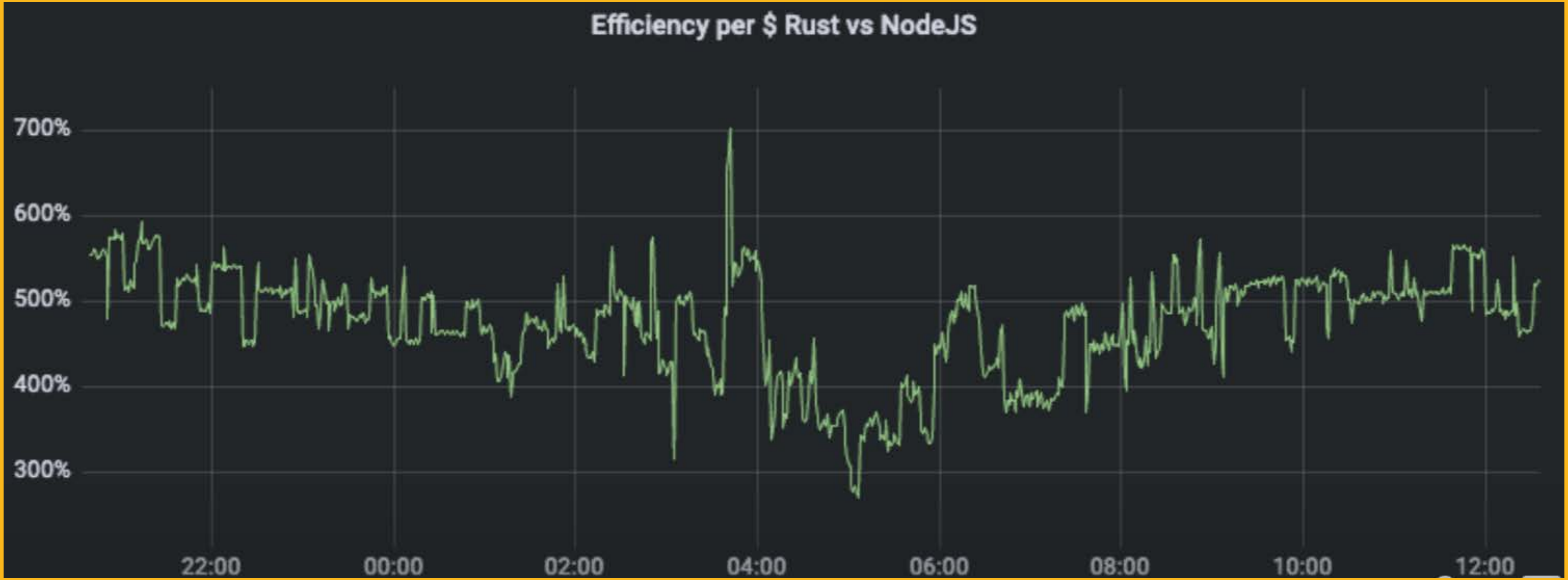
GRAPHQL CASE-STUDY





corrode

GRAPHQL CASE-STUDY



corrode

1. THE CHOSEN PROJECT DETERMINES THE ODDS OF SUCCESSFUL RUST ADOPTION.

(Choose wisely)

corrode

FINDING YOUR FIRST PROJECT FOR RUST

1. **Fix Pain Points**
   Ideal for performance or concurrency issues.

2. **Limit Scope**
   Choose an impactful yet medium-sized projects.

3. **Play Rust's Strengths**
   Find projects benefiting the most from lower operational costs and stability.

corrode

RECRUITING IS HARD AND EXPENSIVE.

TRAIN YOUR OWN PEOPLE.

(Or hire me to do it)

corrode

ADOPTING RUST

1. **Identify Project**
   Select a meaningful project for Rust implementation.

2. **Team Assessment**
   - Do not hire new staff specifically for Rust.
   - Evaluate the current team's readiness:
     - Check for hidden Rust experts.
     - Consider experience in languages similar to Rust. (Java, Kotlin, C++)
     - Gauge the team's willingness to learn Rust.

3. **Upskilling the Team**
   - Self-guided learning using books and hands-on exercises.
   - Organize training workshops.
   - Team augmentation for asking harder questions.
   - Code reviews to improve the codebase.

corrode

YOU NEED A LONG-TERM MINDSET

(Think: years)

corrode

RISKS

- No quick wins
- Steep learning curve
- Long build times (locally and in CI)
- Custom libraries required
- Need to write integrations with existing code

BENEFITS

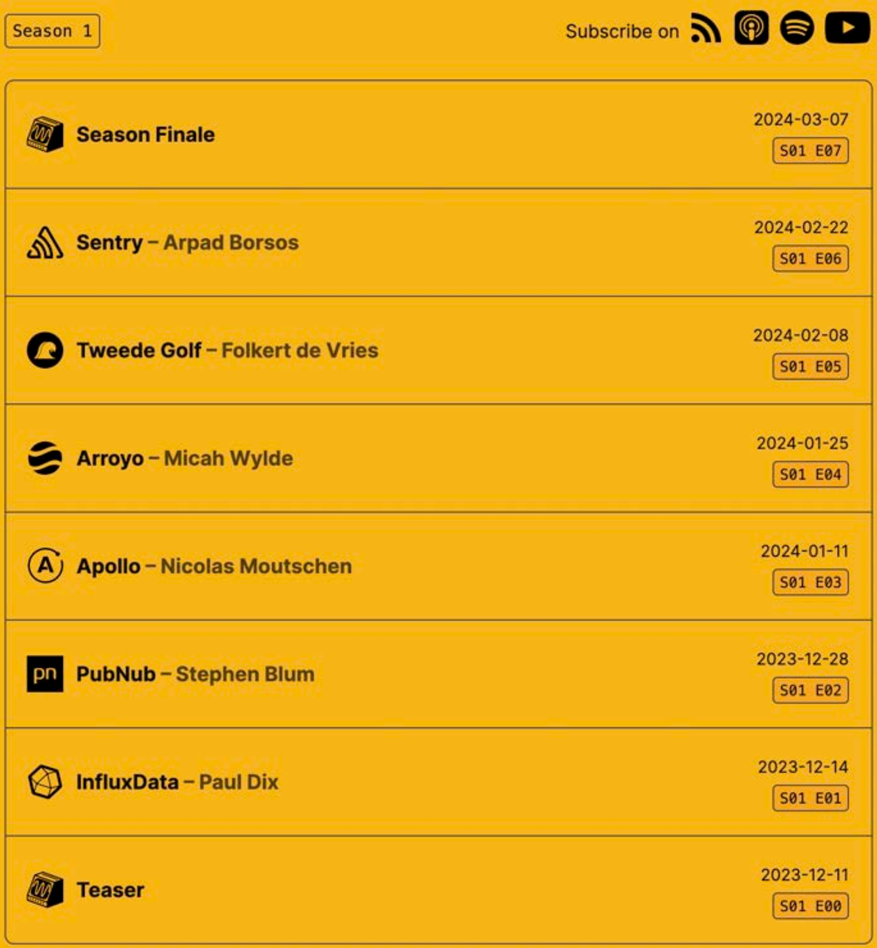
- Reduce operational costs
- Predictable performance
- Enable faster development cycles
- Less friction between dev and ops
- Developer happiness (most loved language for 7 years in a row)
- Gradual adoption possible

corrode

Rust
in
Production
Podcast

corrode.dev/podcast

corrode



Season 1

Subscribe on

Season Finale	2024-03-07	S01 E07
Sentry – Arpad Borsos	2024-02-22	S01 E06
Tweede Golf – Folkert de Vries	2024-02-08	S01 E05
Arroyo – Micah Wylde	2024-01-25	S01 E04
Apollo – Nicolas Moutschen	2024-01-11	S01 E03
PubNub – Stephen Blum	2023-12-28	S01 E02
InfluxData – Paul Dix	2023-12-14	S01 E01
Teaser	2023-12-11	S01 E00

corrode

corrode.dev/podcast

corrode