State of Rust

Jakub Beránek

With input from:

- Vitaly Bragilevsky
- Hugo van de Pol
- Joran Dirk Greef

What?

What?
Why?

What?
Why?
Where?

What?
Why?
Where?
Who?

What?
Why?
Where?
Who?
When?

Jakub Beránek

github.com/kobzol

@ jakub@berankovi.net

Jakub Beránek

github.com/kobzol

@ jakub@berankovi.net

- PhD, teaching, research @ IT4Innovations (Czech Republic supercomputing center)

Jakub Beránek

github.com/kobzol

@ jakub@berankovi.net

- PhD, teaching, research @ IT4Innovations (Czech Republic supercomputing center)
- Rust Project open-source contributor

Jakub Beránek

github.com/kobzol

@ jakub@berankovi.net

- PhD, teaching, research @ IT4Innovations (Cz
- Rust Project open-source contributor
  - Rust Infrastructure team (member)

Infrastructure team

Managing the infrastructure supporting the Rust project itself, including CI, releases, bots, and metrics

Members

Jan David Nose

GitHub: jdno

Team leader

Jake Goulding

GitHub: shepmaster

Team leader

kennytm

GitHub: kennytm

Jakub Beránek

GitHub: Kobzol

Mark Rousskov

GitHub: Mark-Simulacrum

Pietro Albini

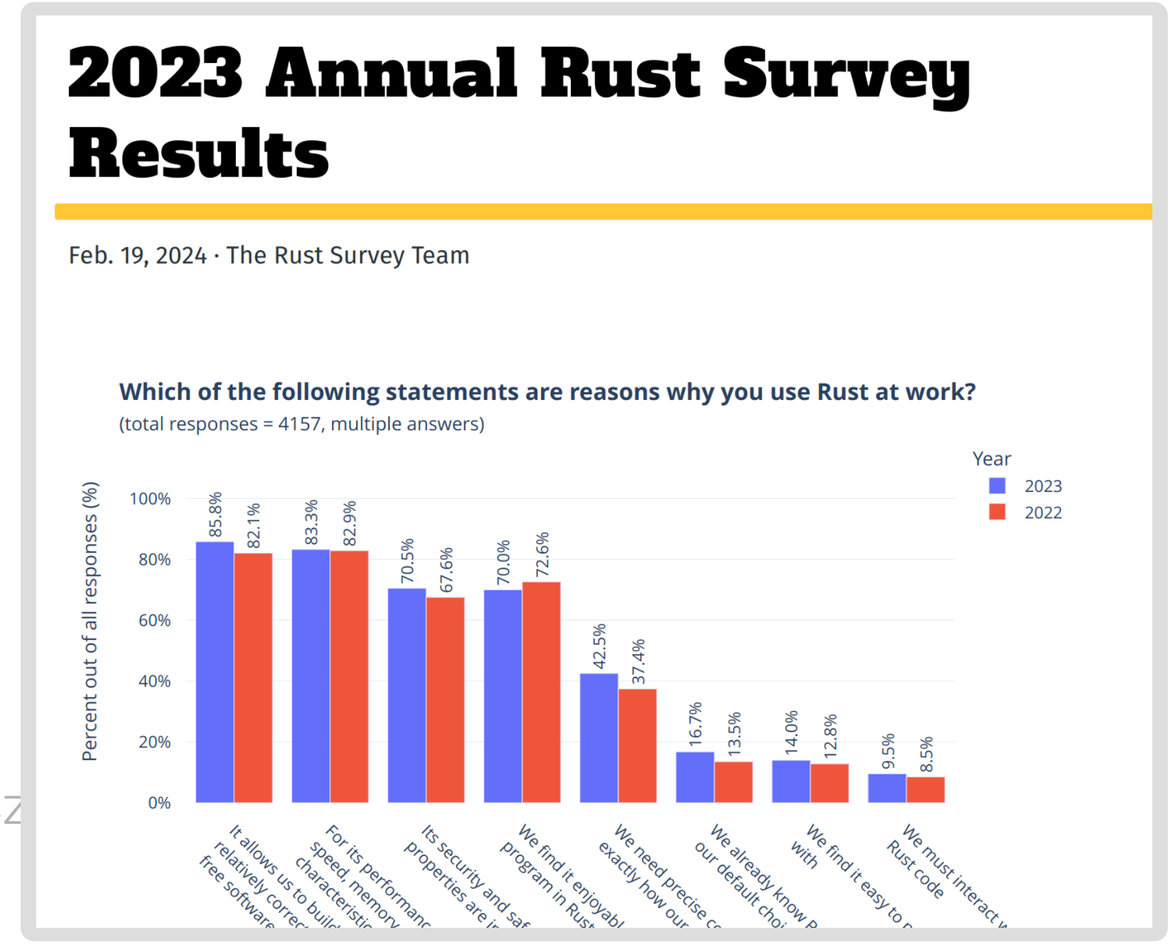
GitHub: pietroalbini

Jakub Beránek

github.com/kobzol

@ jakub@berankovi.net

- PhD, teaching, research @ IT4Innovations (Cz
- Rust Project open-source contributor
  - Rust Infrastructure team (member)
  - Rust Survey team (co-lead)



(disclaimer)

All opinions (and possible errors) are my own :-)

Rust

A language empowering everyone
to build reliable and efficient software.

Rust timeline

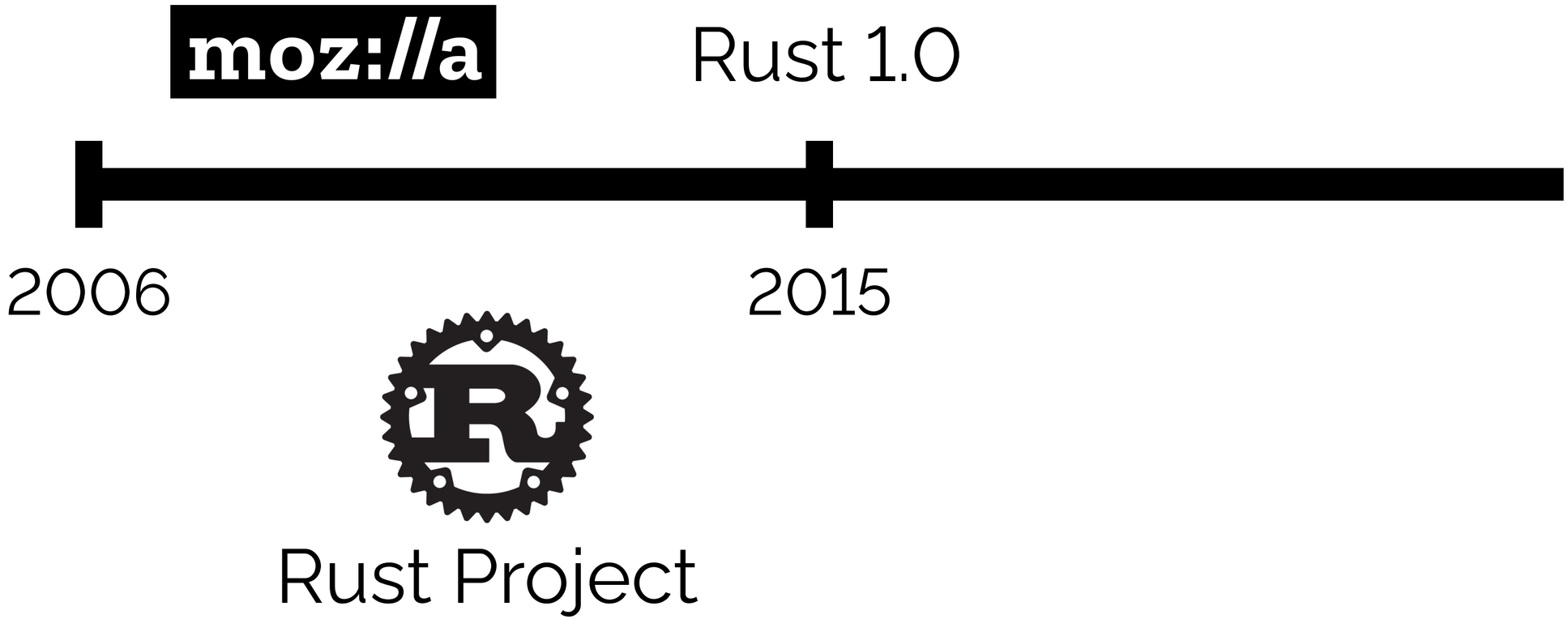
Rust timeline

Rust timeline

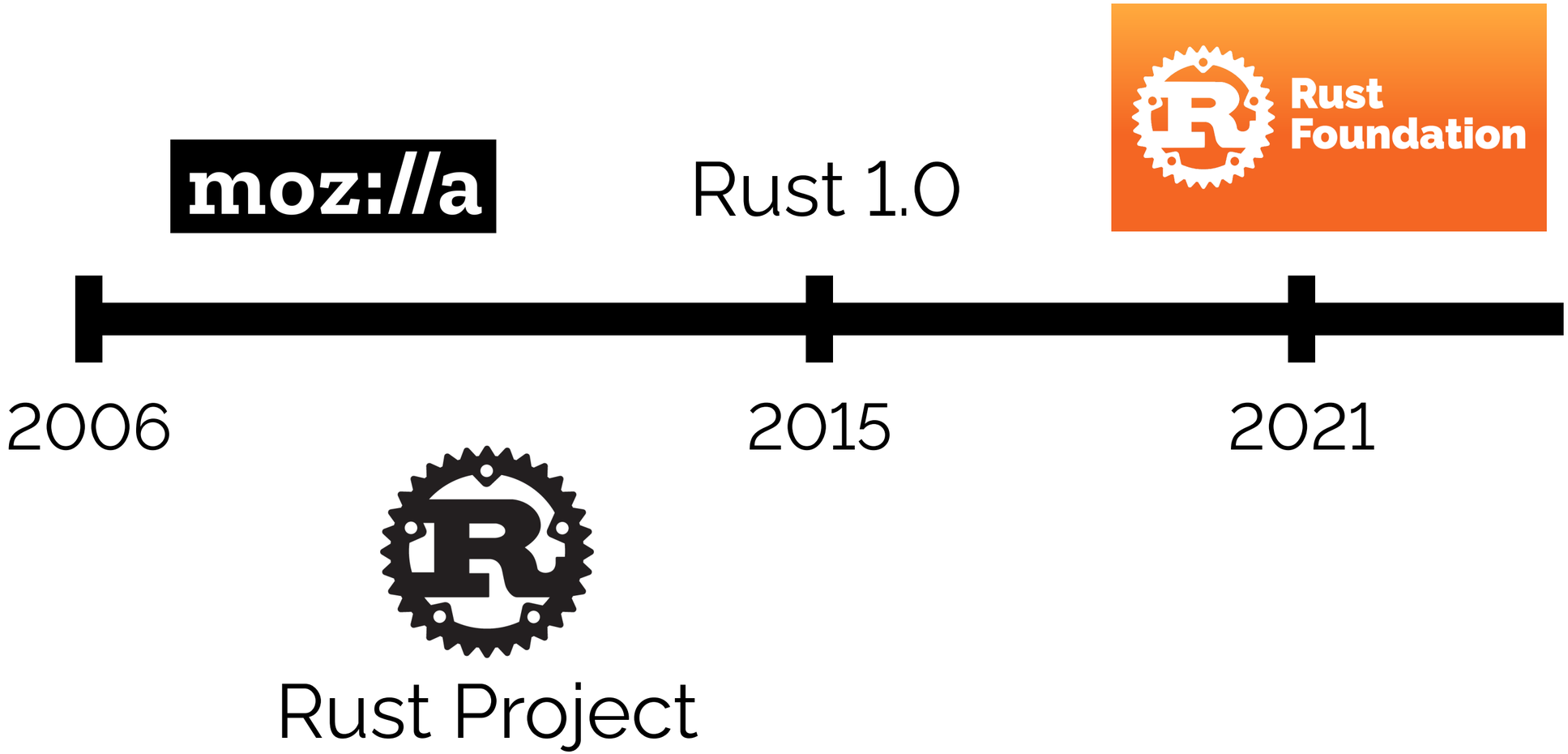
2006

Rust Project

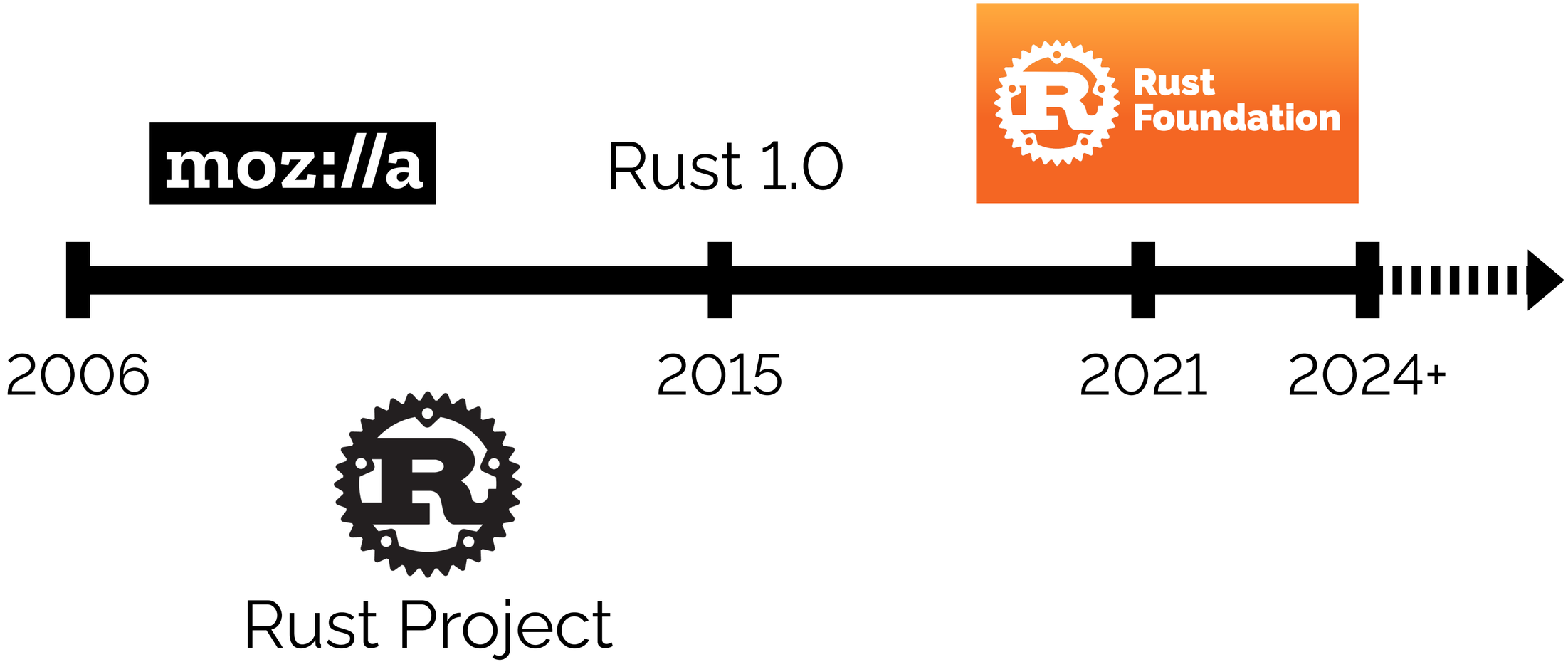
Rust timeline



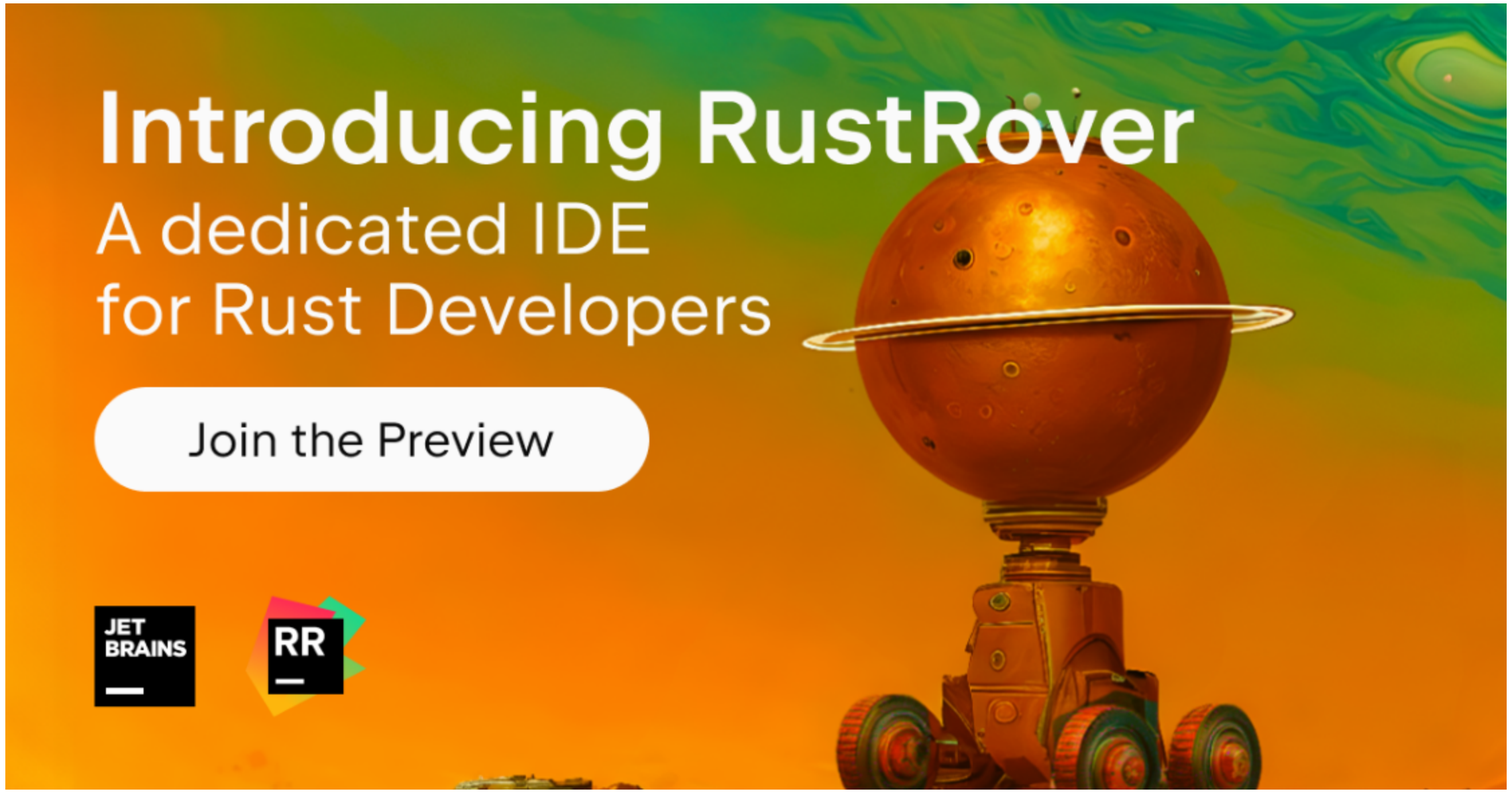
Rust timeline



Rust timeline



Rust Foundation Platinum members


Introducing RustRover
A dedicated IDE
for Rust Developers
Join the Preview
JET
BRAINS
RR

Why Rust?

Performance

Reliability

Productivity

Why Rust?

Performance

Reliability

Productivity

Performance

Rust programs are efficient

Performance

Rust programs are efficient **by default.**

Performance

Rust programs are efficient **by default.**

"Time-to-performance"

Performance

ENGINEERING & DEVELOPERS

WHY DISCORD IS SWITCHING FROM GO TO RUST

Jesse Howarth
February 4, 2020

Rust is becoming a first class language in a variety of domains. At Discord, we've seen success with Rust on the client side and server side. For example, we use it on the client side for our video encoding pipeline for Go Live and on the server side for Elixir NIFs.

Performance

> “
> Even with **just basic optimization**,
> Rust was able to **outperform the hyper hand-tuned** Go version...
>
> Discord ”

Performance

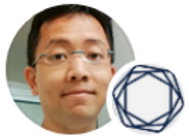
> “
> Even with **just basic optimization**,
> Rust was able to **outperform the hyper hand-tuned** Go version...
> ...we were able to beat Go on every single performance metric.
>
> Discord ”

Performance

More efficient => **cheaper** to run

Performance

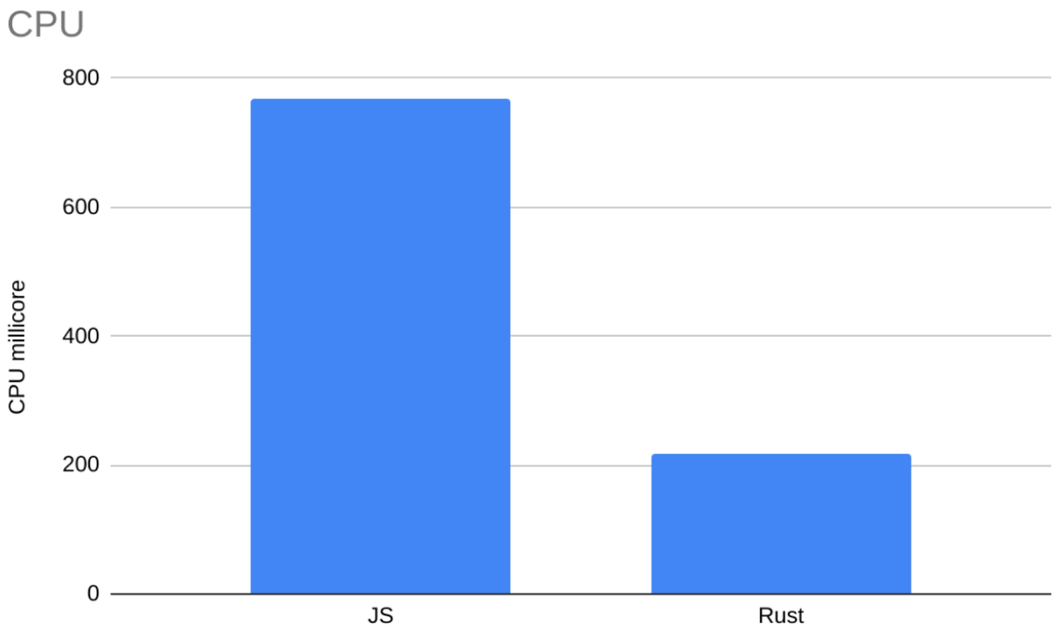
Optimizing 700 CPUs Away With Rust



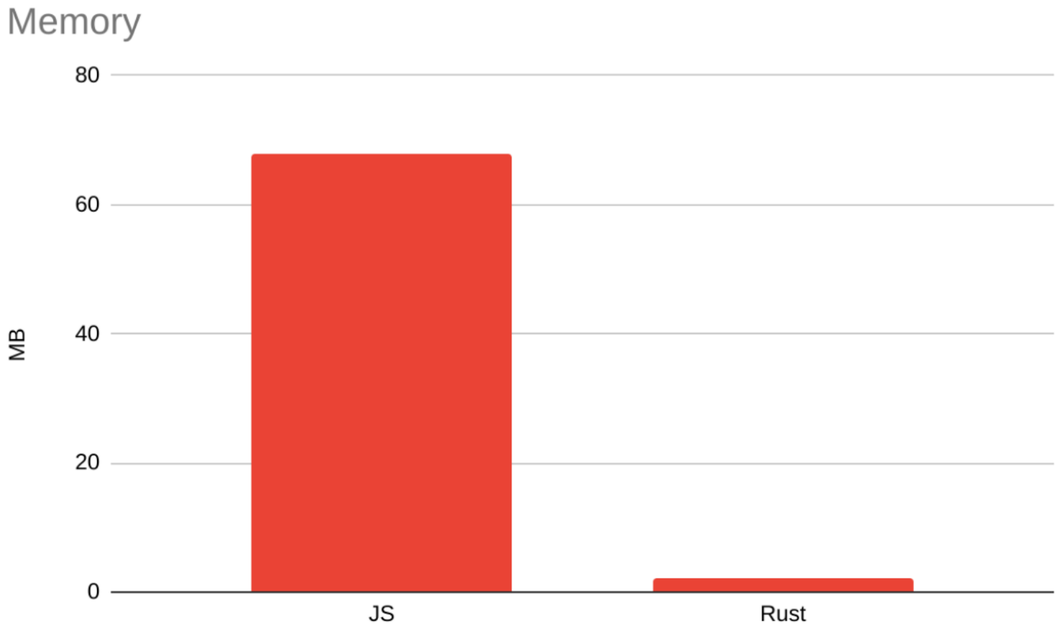
Alan Ning · Follow

Published in Tenable TechBlog · 3 min read · May 6, 2021

Performance



Per pod, average CPU reduced from 800m to 200m core



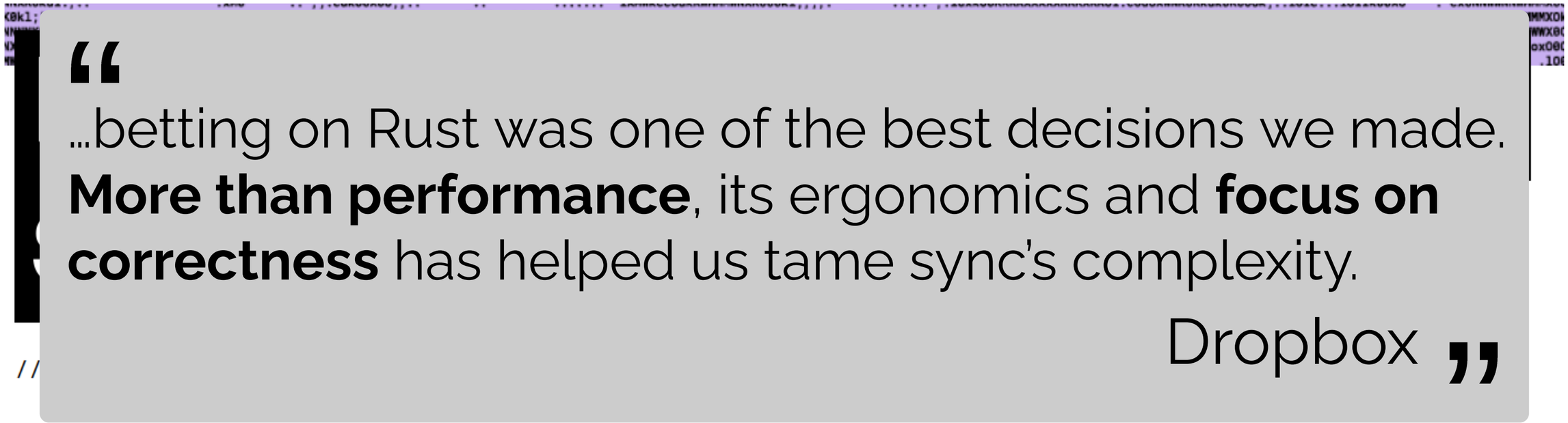
Per pod, average memory reduced from 70MB to 5MB.

Performance

Rewriting the heart of our sync engine

// By Sujay Jayakar • Mar 09, 2020

Performance



> “
> …betting on Rust was one of the best decisions we made. **More than performance**, its ergonomics and **focus on correctness** has helped us tame sync’s complexity.
>
> Dropbox ”

Why Rust?

Performance

Reliability

Productivity

Reliability

Rust makes it easier to write **correct software**

Reliability

Rust makes it easier to write **correct software**
by adding **guardrails**

Confidence

Reliability

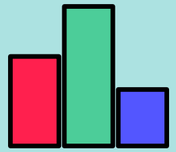
> “
> The language's expressiveness allows our developers to encode constraints that **catch errors at compile time rather than in GitHub issues**.
>
> Vercel ”

Reliability

Which of the following statements about Rust do you feel are true?

"Rust is risky to use in production"

1.6%



Source: Rust Annual 2023 survey

Reliability

Rust is memory safe by default

Chrome: 70% of all security bugs are memory safety issues

Google software engineers are looking into ways of eliminating memory management-related bugs from Chrome.

Reliability

Chrome: 70% of all security bugs are memory safety issues

Google software engineers are looking into ways of eliminating memory management-related bugs from Chrome.

Microsoft: 70 percent of all security bugs are memory safety issues

Percentage of memory safety issues has been hovering at 70 percent for the past 12 years.

Reliability

Chrome: 70% of all security bugs are memory safety issues

Google software engineers are looking into ways of eliminating memory management-related bugs from Chrome.

Microsoft: 70 percent of all security bugs are memory safety issues

Percentage of memory safety issues has been hovering at 70 percent for the past 12 years.

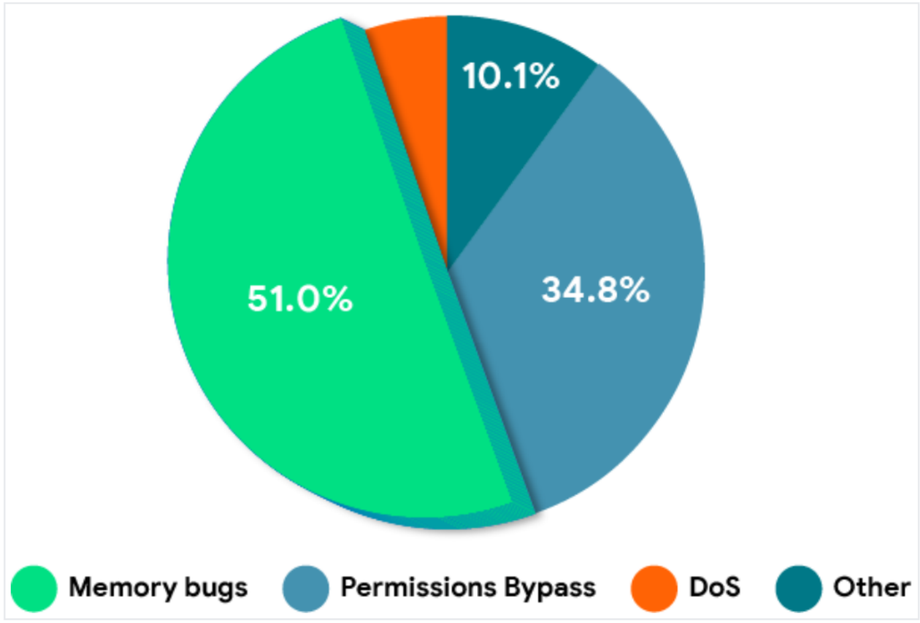


Figure 2: Memory safety bugs contribution to Android vulnerabilities

The White House

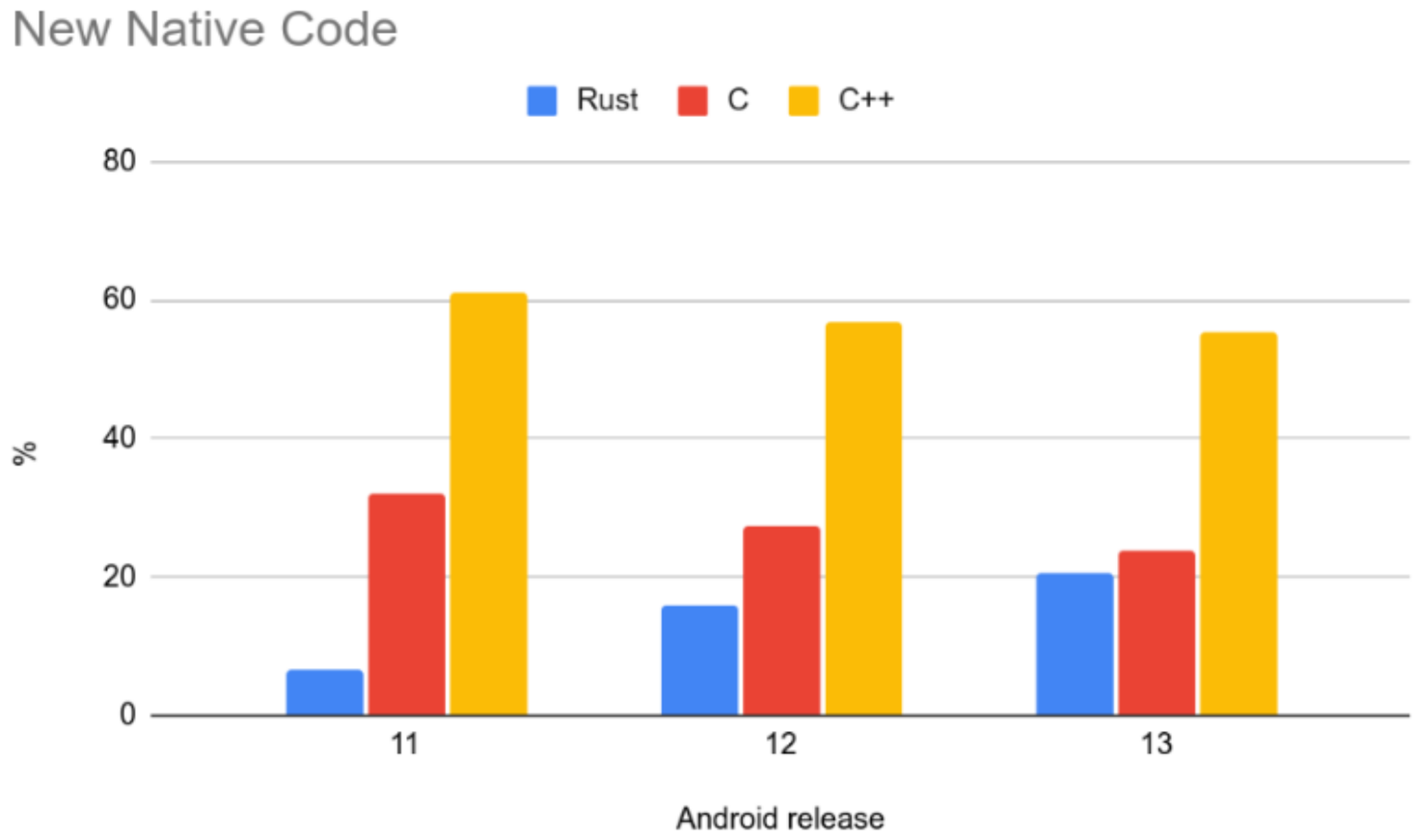
FEBRUARY 26, 2024

Press Release: Future Software Should Be Memory Safe

Reliability

Memory Safe Languages in Android 13

December 1, 2022



Reliability

A Memory Safe Implementation of the Network Time Protocol

Folkert de Vries
Oct 11, 2022

Bringing Memory Safety to sudo and su

Josh Aas
Apr 26, 2023

Stability

Stability

Rust 1.0 released in 2015 (9 years ago)

Stability

Rust 1.0 released in 2015 (9 years ago)

...without stagnation

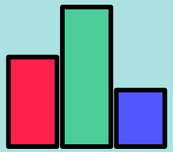
Rust 1.78 released last week

Reliability

Do you agree with the following statements on Rust stability?

"**Upgrading** to a new stable compiler version **requires** either **no changes** or extremely small & easy changes to my code"

97.72%



Source: Rust Annual 2023 survey

Why Rust?

Performance

Reliability

Productivity

Productivity

Unified tooling

Productivity

Unified tooling

- Building

Productivity

Unified tooling

- Building
- Testing

Productivity

Unified tooling

- Building
- Testing
- Dependency management

Productivity

Unified tooling

- Building
- Testing
- Dependency management
- Documentation

Productivity

Unified tooling

- Building
- Testing
- Dependency management
- Documentation
- Deployment

Productivity

Unified tooling

- Building
- Testing
- Dependency management
- Documentation
- Deployment
- …

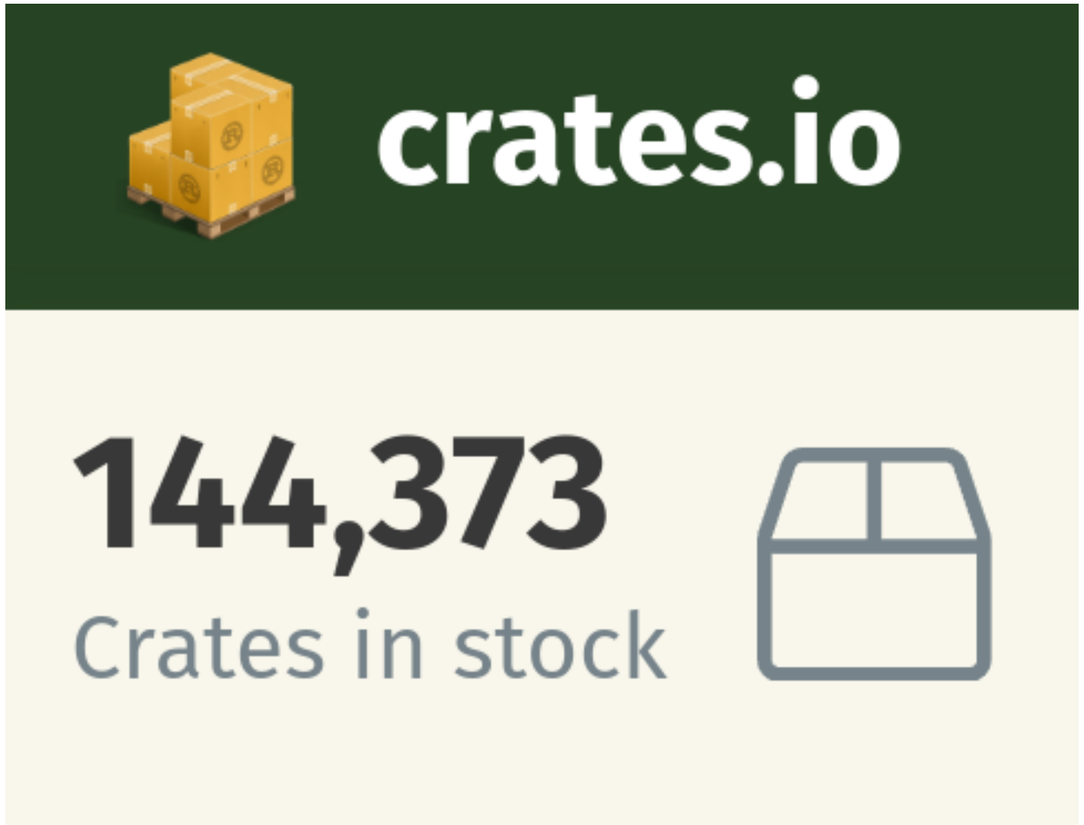
Unified tooling

- Building
- Testing
- Dependency management
- Documentation
- Deployment
- …

All with a **single tool** (Cargo)

Productivity

Package ecosystem



Productivity

> “
> Rust has absolutely **stunning dependency management**.
> NPM ”

Trade-offs

Rust is great for
efficient, production-ready software

Trade-offs

Rust is great for
efficient, production-ready software

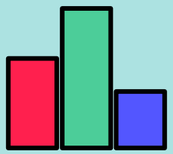
Not all software is like that

Trade-offs

Which of the following statements are reasons why you use Rust at work?

"We find it easy to prototype with"

14%



Source: Rust Annual 2023 survey

Trade-offs

Longer initial ramp-up time

Trade-offs

Longer initial ramp-up time

🤝

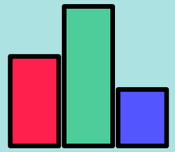
Less bugs in production

Trade-offs

Which of the following statements apply to your experience using Rust at work?

"Overall, adopting Rust has slowed down our team"

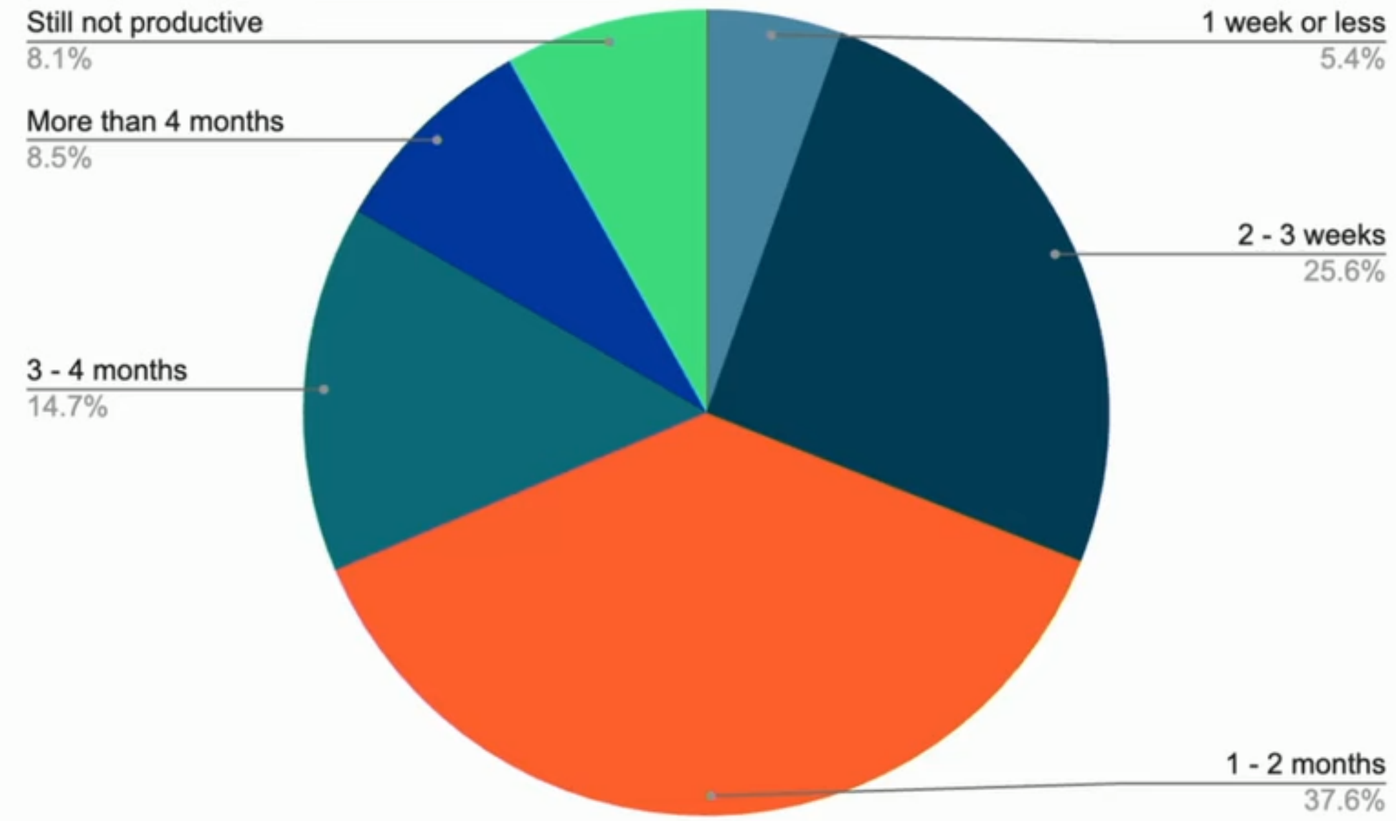
8.2%



Source: Rust Annual 2023 survey

Onboarding

Trade-offs



Confidence Contributing to Rust Codebase

Q14 - While learning Rust, how long did it take you to become sufficiently productive to contribute to your Rust codebase without worrying about the language? (choose one)

More than 2/3 of respondents are confident contributing to Rust codebase within 2 months or less when learning Rust.

Source: Lars Bergstrom - Beyond Safety and Speed: How Rust Fuels Team Productivity

Ecosystem maturity

Ecosystem maturity

Many libraries, less frameworks

Trade-offs

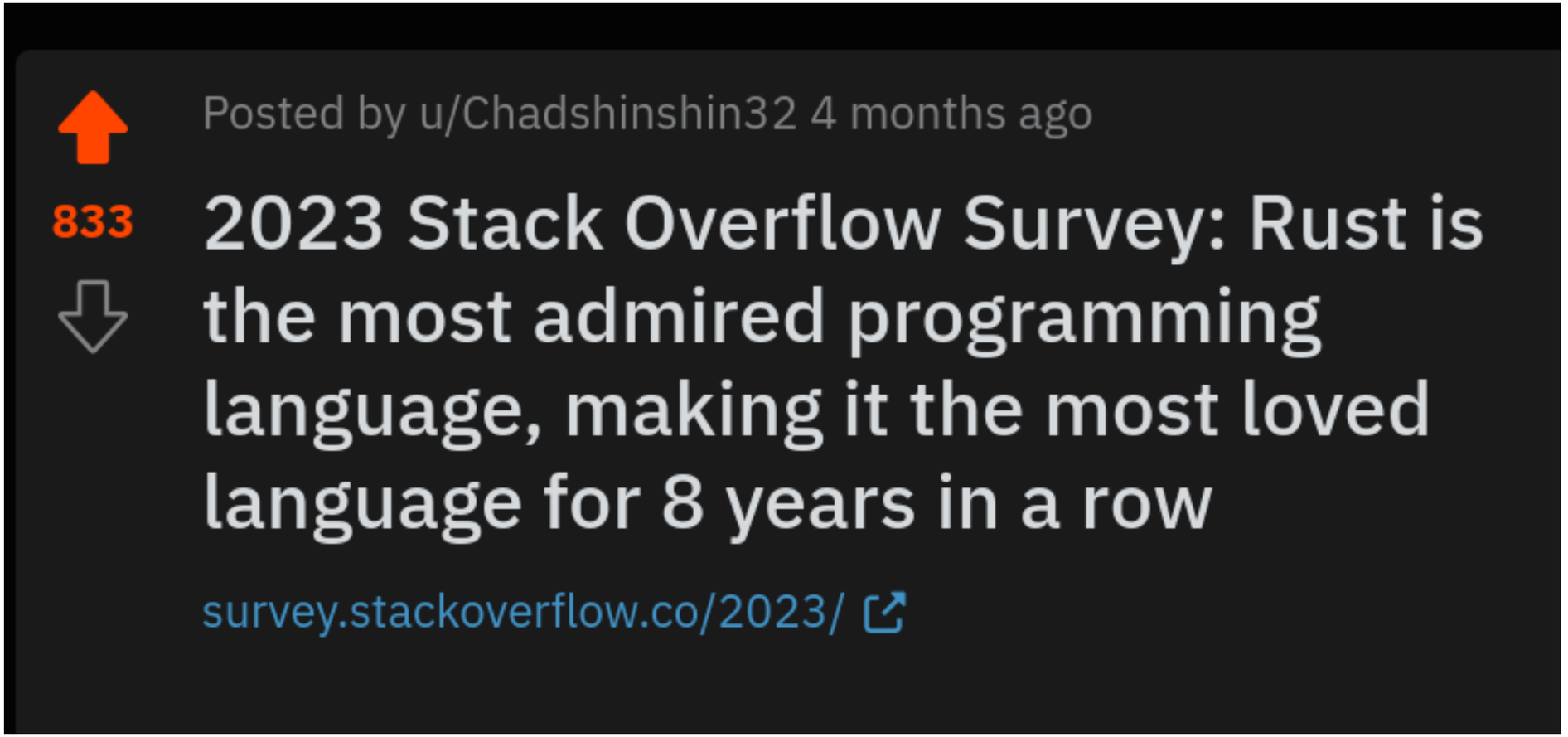
> “Rust has a Lego-like package ecosystem.”
>
> Luca Palmieri

Why Rust?

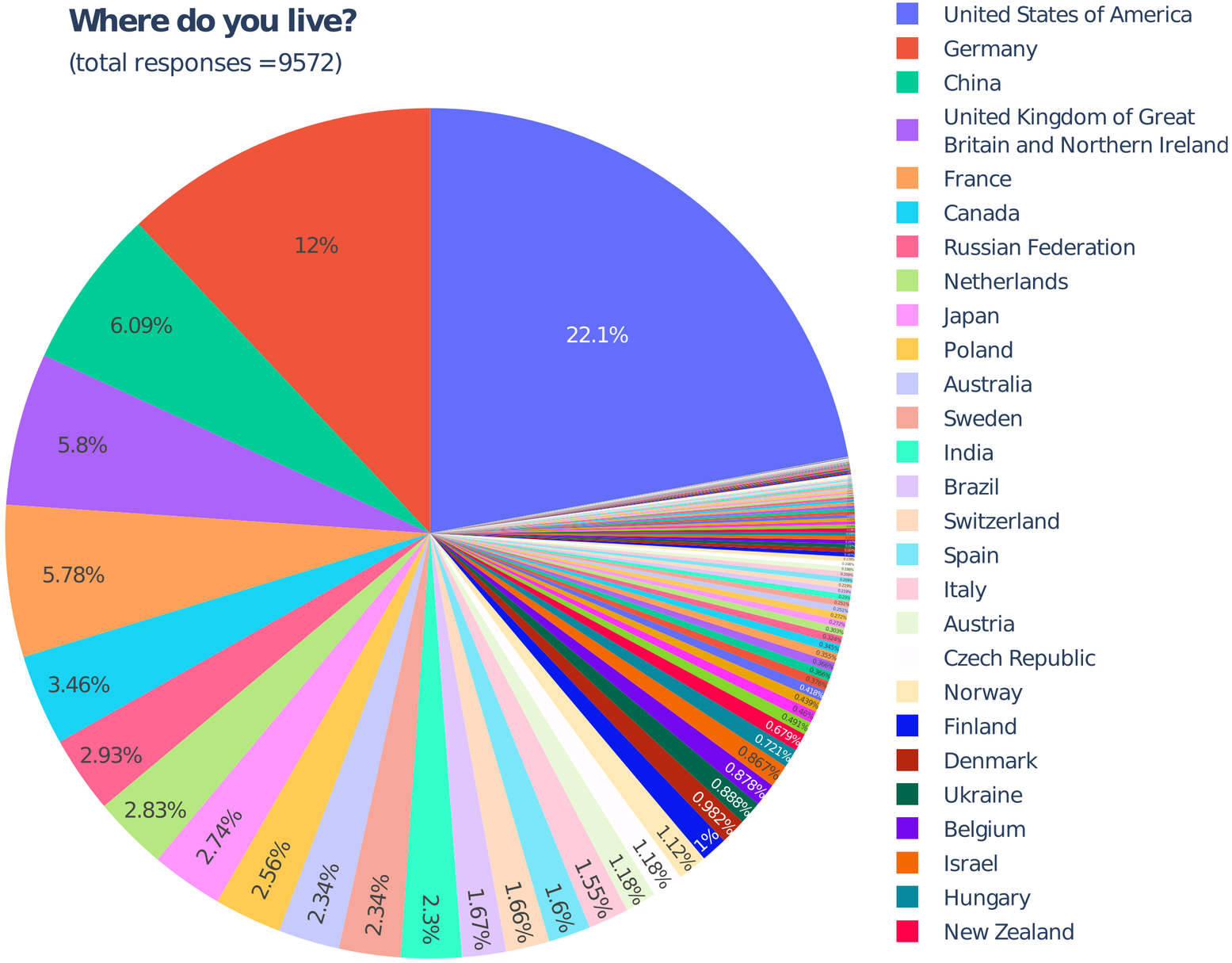
Performance

Reliability

Productivity


Posted by u/Chadshinshin32 4 months ago
833
2023 Stack Overflow Survey: Rust is the most admired programming language, making it the most loved language for 8 years in a row
survey.stackoverflow.co/2023/

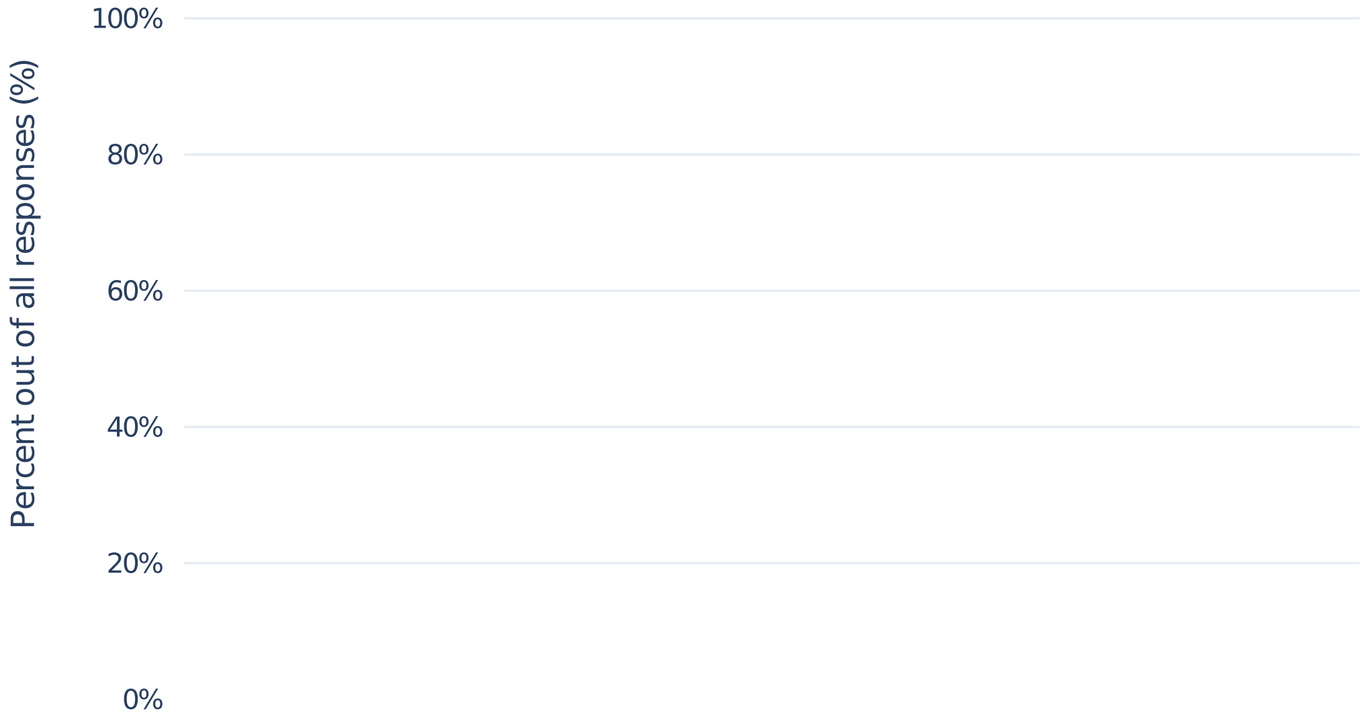
Where is Rust being used today?



Source: Rust Annual 2023 survey

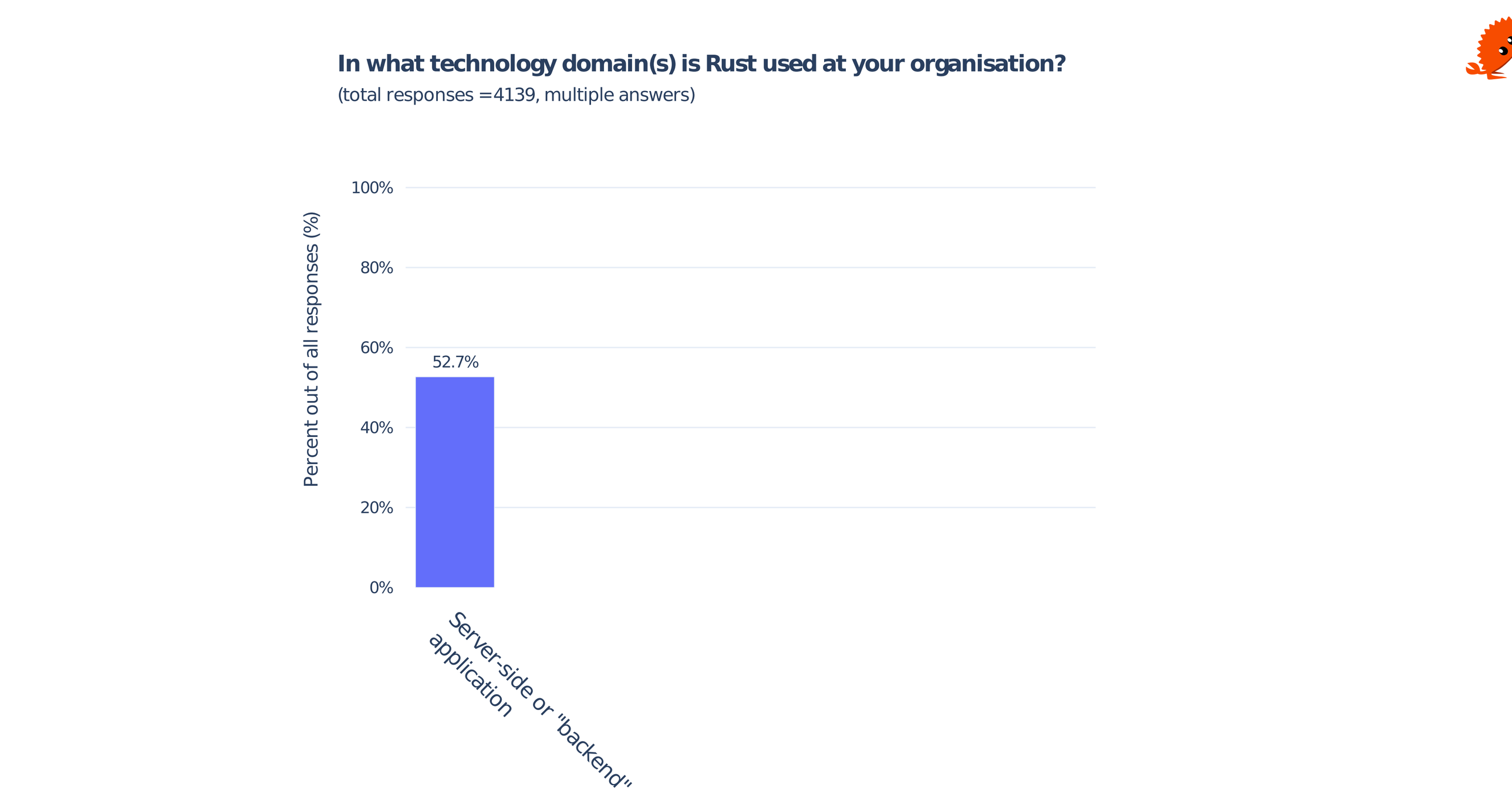
In what technology domain(s) is Rust used at your organisation?

(total responses =4139, multiple answers)



Not all answers are displayed

Source: Rust Annual 2023 survey



Not all answers are displayed

Source: Rust Annual 2023 survey

In what technology domain(s) is Rust used at your organisation?

(total responses =4139, multiple answers)

100%

GitHub built a new search engine for code 'from scratch' in Rust

GitHub built a new code-focused search engine in Rust because popular text search engines couldn't scale enough.

Written by **Liam Tung,** Contributing Writer

Feb. 9, 2023 at 3:24 a.m. PT

side or "backend"

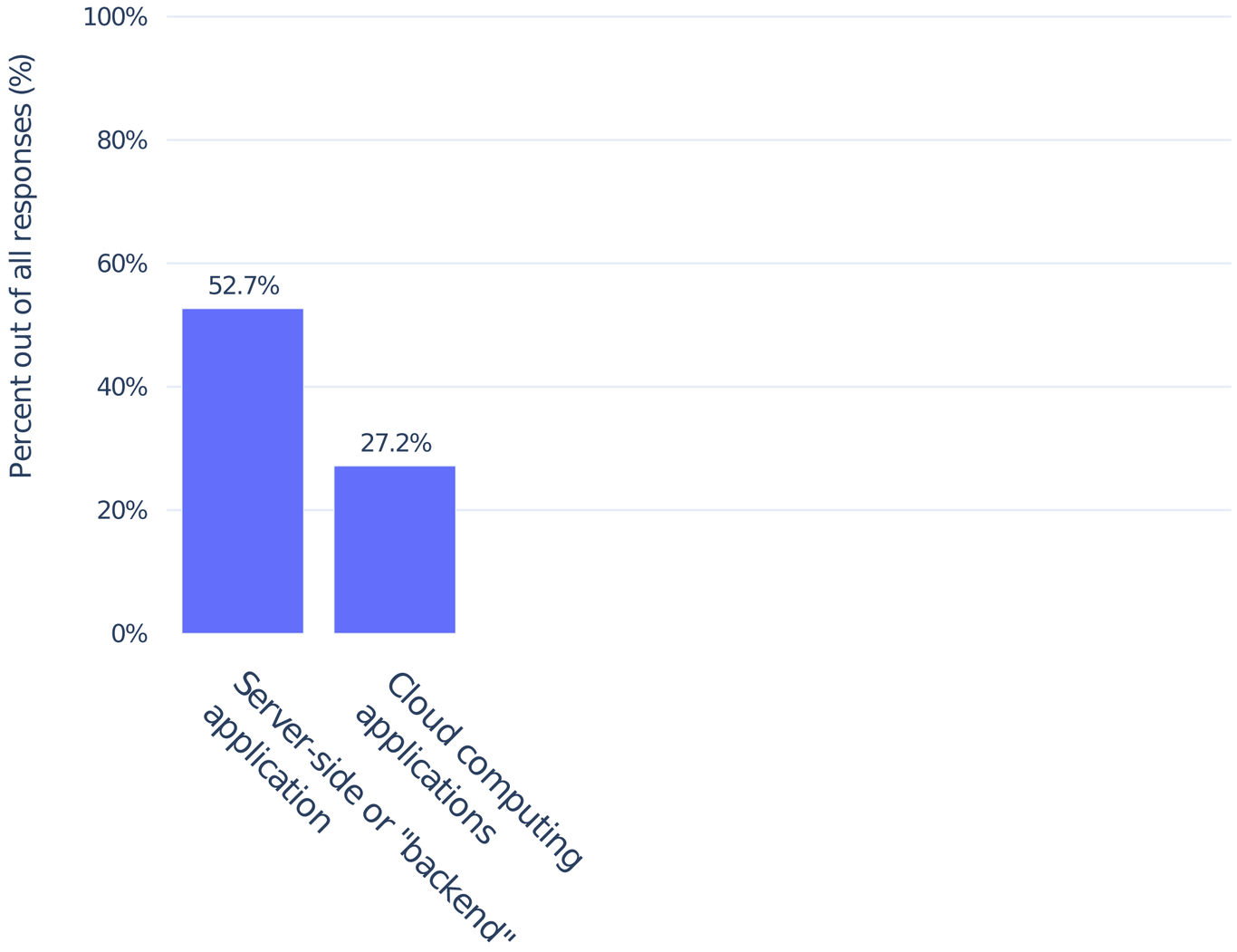
ication

Not all answers are displayed

Source: Rust Annual 2023 survey

In what technology domain(s) is Rust used at your organisation?

(total responses =4139, multiple answers)



Not all answers are displayed

Source: Rust Annual 2023 survey

In what technology domain(s) is Rust used at your organisation?

(total responses =4139, multiple answers)

Serverless Rust with Cloudflare Workers

10/16/2018

Steven Pack

CLOUDFLARE

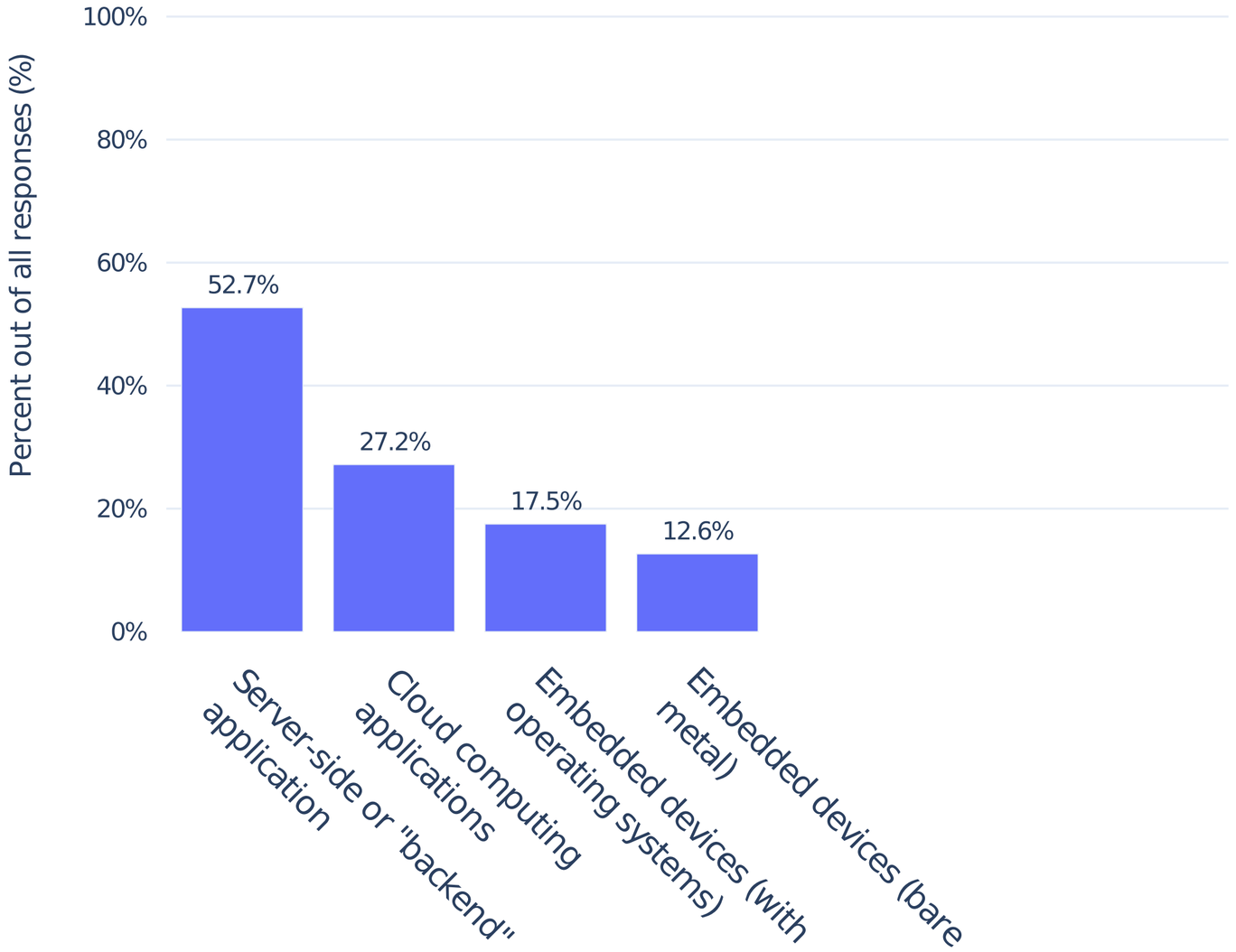
side or "backend"

Not all answers are displayed

Source: Rust Annual 2023 survey

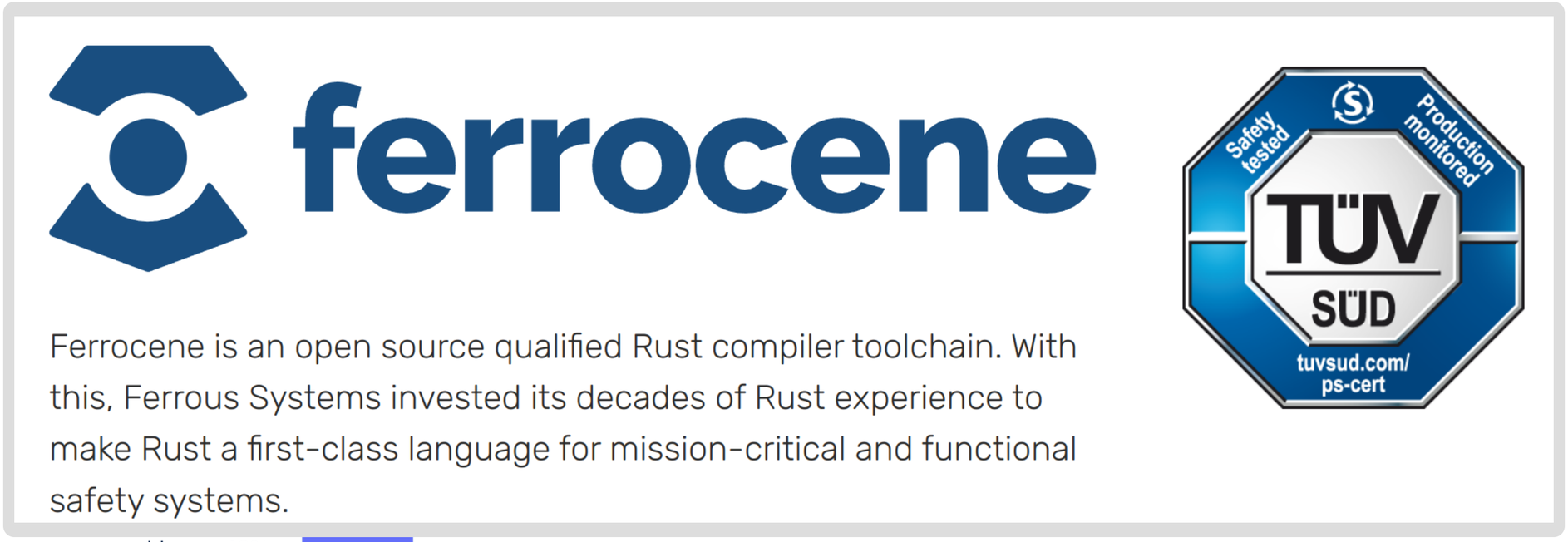
In what technology domain(s) is Rust used at your organisation?

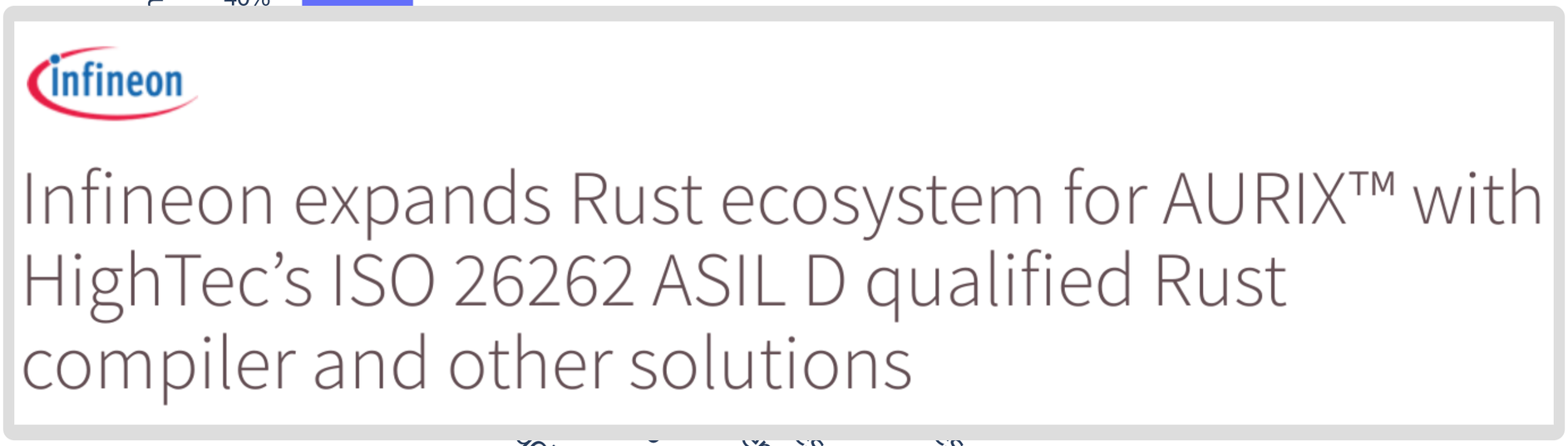
(total responses =4139, multiple answers)



Not all answers are displayed

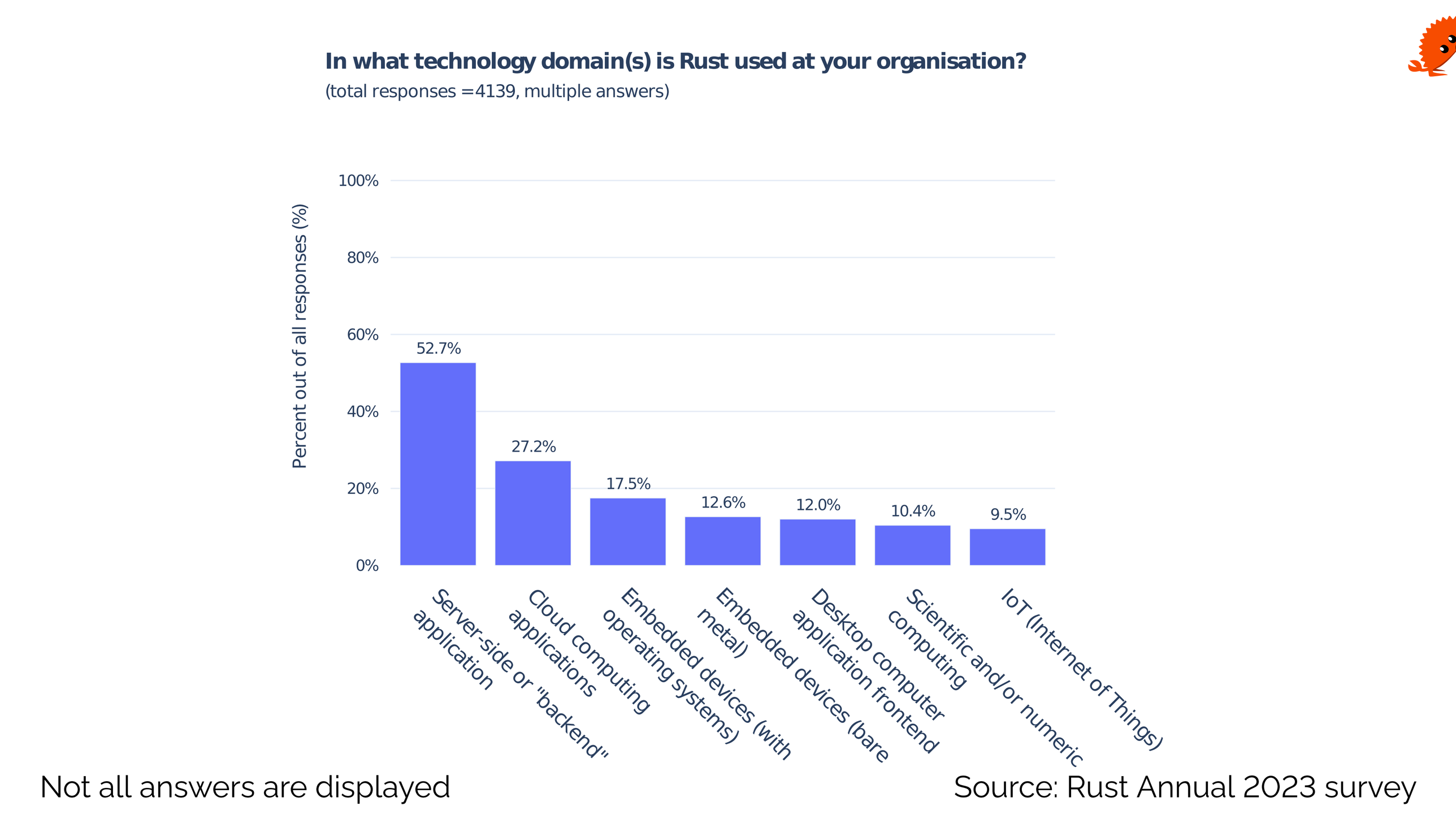
Source: Rust Annual 2023 survey





Not all answers are displayed

Source: Rust Annual 2023 survey

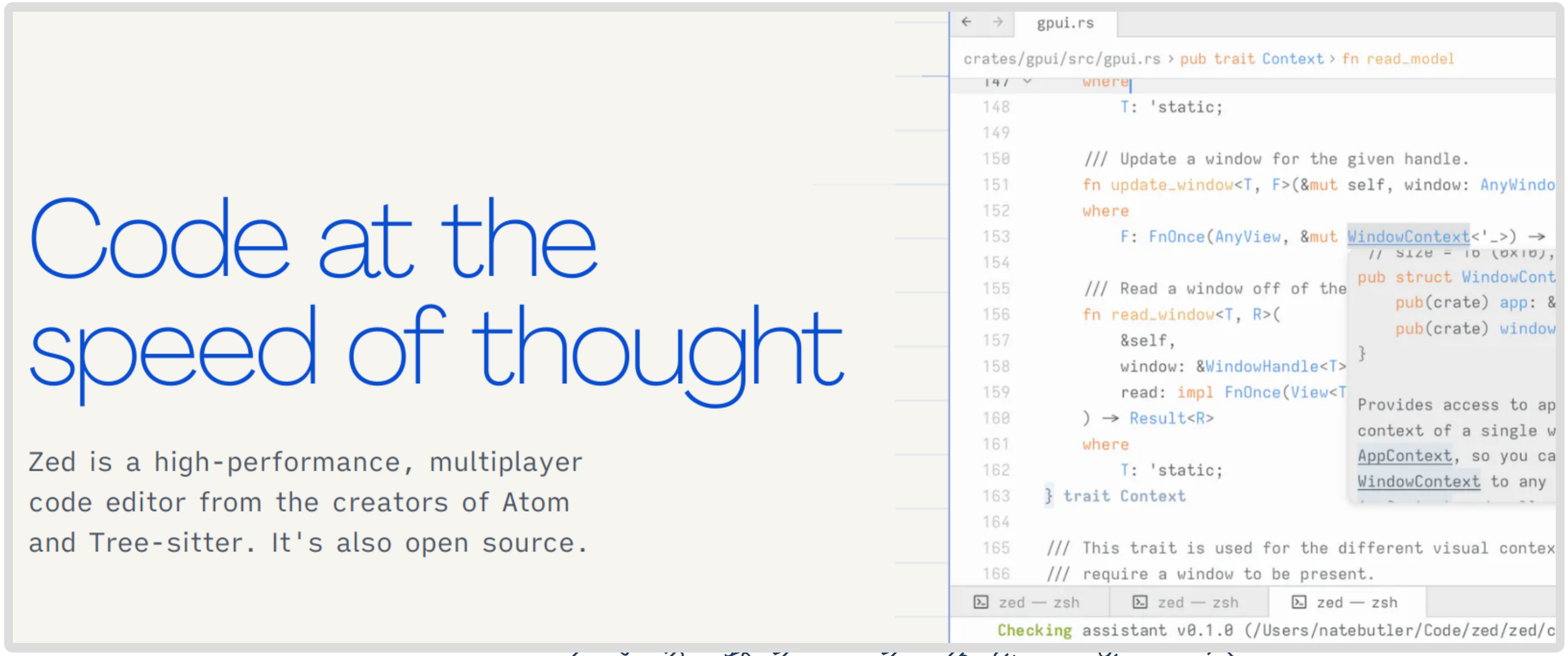


Not all answers are displayed

Source: Rust Annual 2023 survey

In what technology domain(s) is Rust used at your organisation?

(total responses =4139, multiple answers)



backend"

ng

ystems)

ices (with

ices (bare

uter

rontend

or numeric

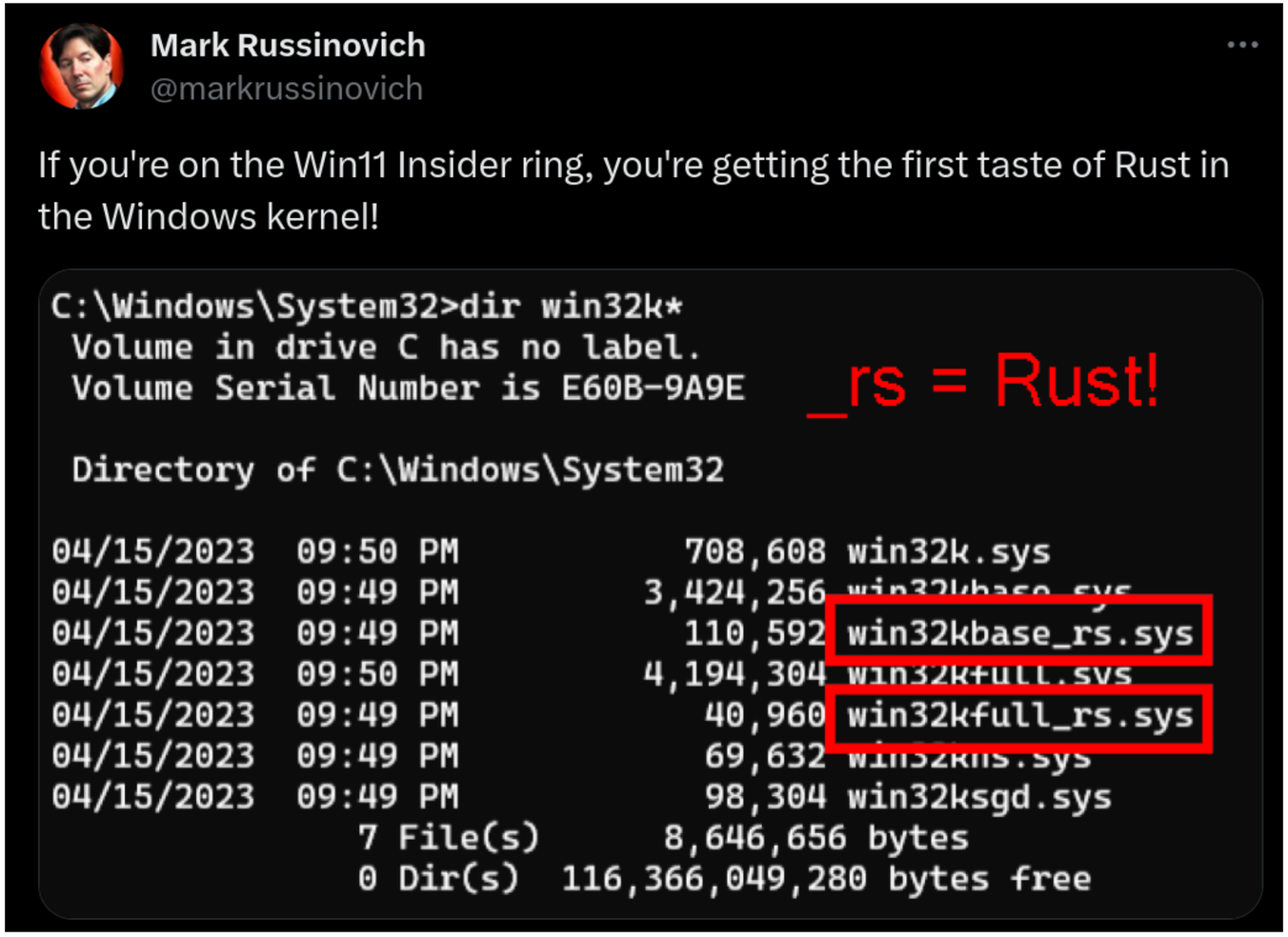
Things)

Not all answers are displayed

Source: Rust Annual 2023 survey

Microsoft posts ‘early stages’ code for developing Windows drivers in Rust

By **Tim Anderson** - September 25, 2023



Rust Programming Language To Land in Linux Kernel 6.1

By Ian Evenden published October 06, 2022

Linux will support the Rust programming language in its kernel from version 6.1.

Rust Programming Language To Land in Linux Ke

By Ian Evenden pub

Linux will sup e in its kernel from version 6.1.

> “
> …on the whole, I don't hate it.
> Linus Torvalds ”

Make Ship Happen

Turbo is an incremental bundler and build system optimized for JavaScript and TypeScript, written in Rust.

TURBOREPO

High-performance build system for JavaScript and TypeScript codebases.

BETA

TURBOPACK

Introducing the Rust-powered successor to Webpack.

VERSION
v0.4.3

RUFF

Lint at lightspeed

An extremely fast Python linter, written in Rust.

Rust anywhere

Rust anywhere

- Linux, Windows, macOS, ...

Rust anywhere

- Linux, Windows, macOS, …
- Cloud, backend, frontend, desktop, mobile, …

Rust anywhere

- Linux, Windows, macOS, …
- Cloud, backend, frontend, desktop, mobile, …
- From embedded devices to supercomputers

Rust anywhere

- Linux, Windows, macOS, …
- Cloud, backend, frontend, desktop, mobile, …
- From embedded devices to supercomputers
- Code reuse

Rust anywhere

- Linux, Windows, macOS, …
- Cloud, backend, frontend, desktop, mobile, …
- From embedded devices to supercomputers
- Code reuse
- Interoperable with C, C++, Python, WebAssembly, …

Adoption of Rust


aws

Google

HUAWEI

Meta

Microsoft

npm

Dropbox

Discord

CLOUDFLARE

ATLASSIAN

1Password

arm

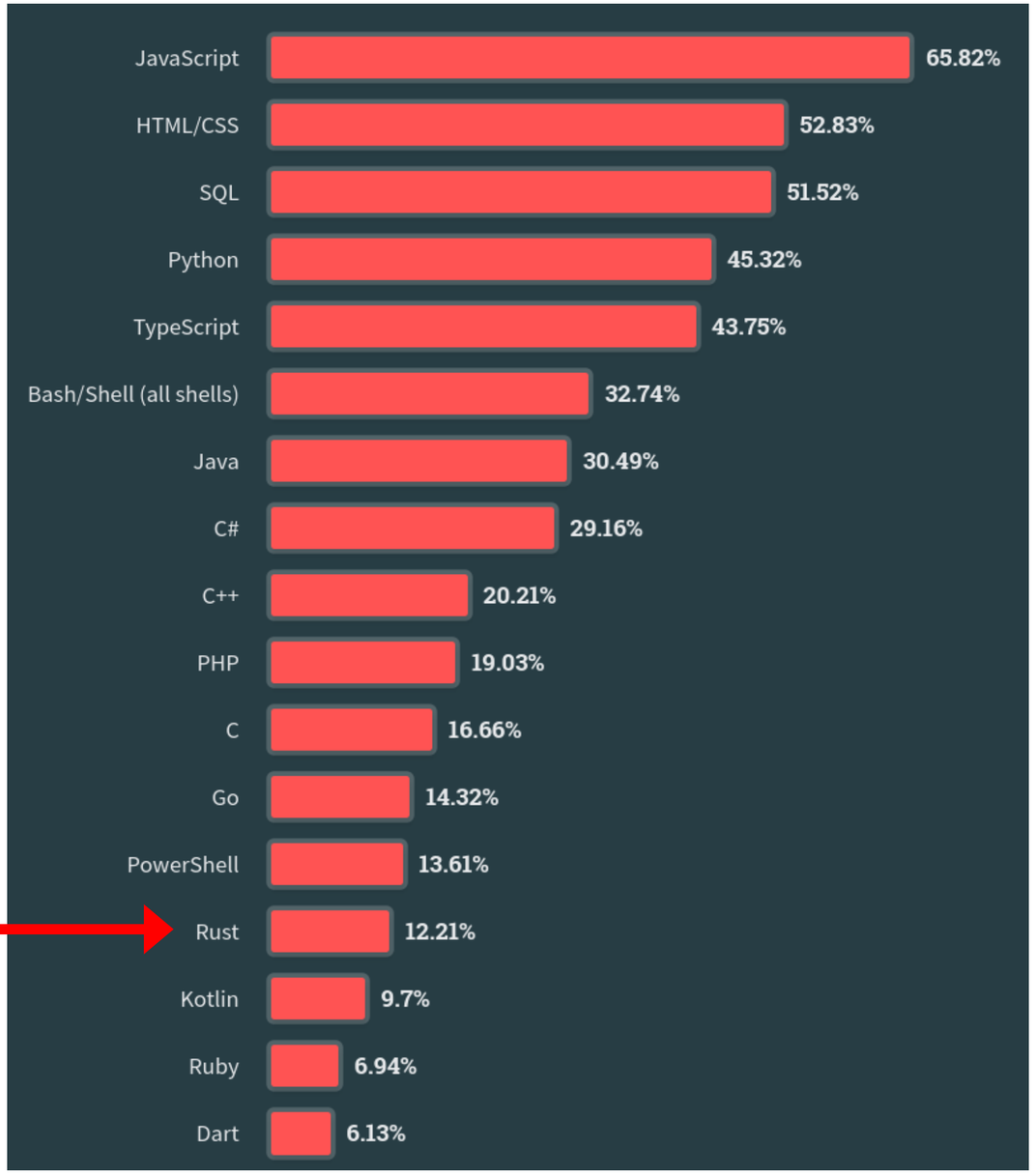
JET
BRAINS

Figma

SENTRY

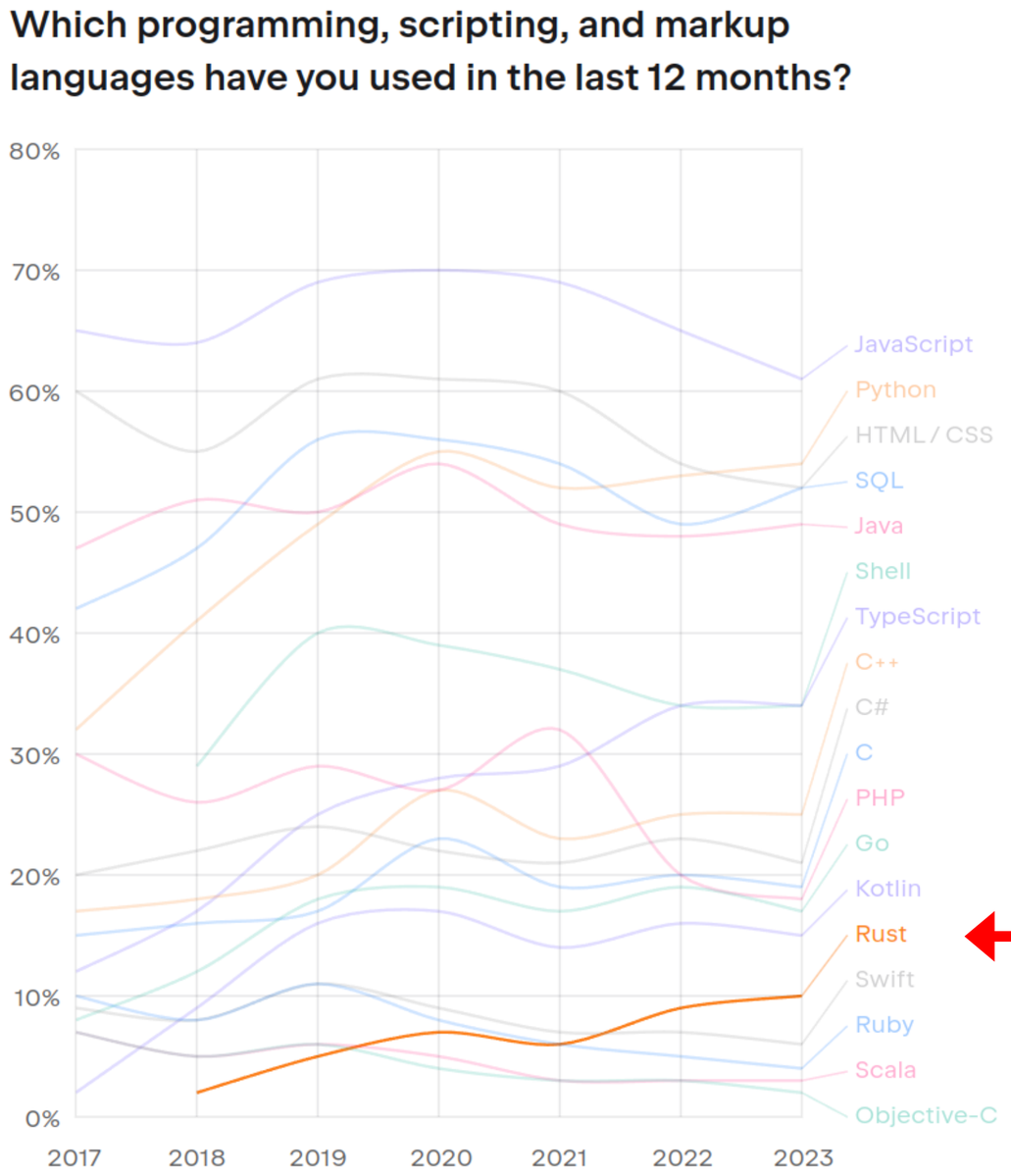
shopify

...



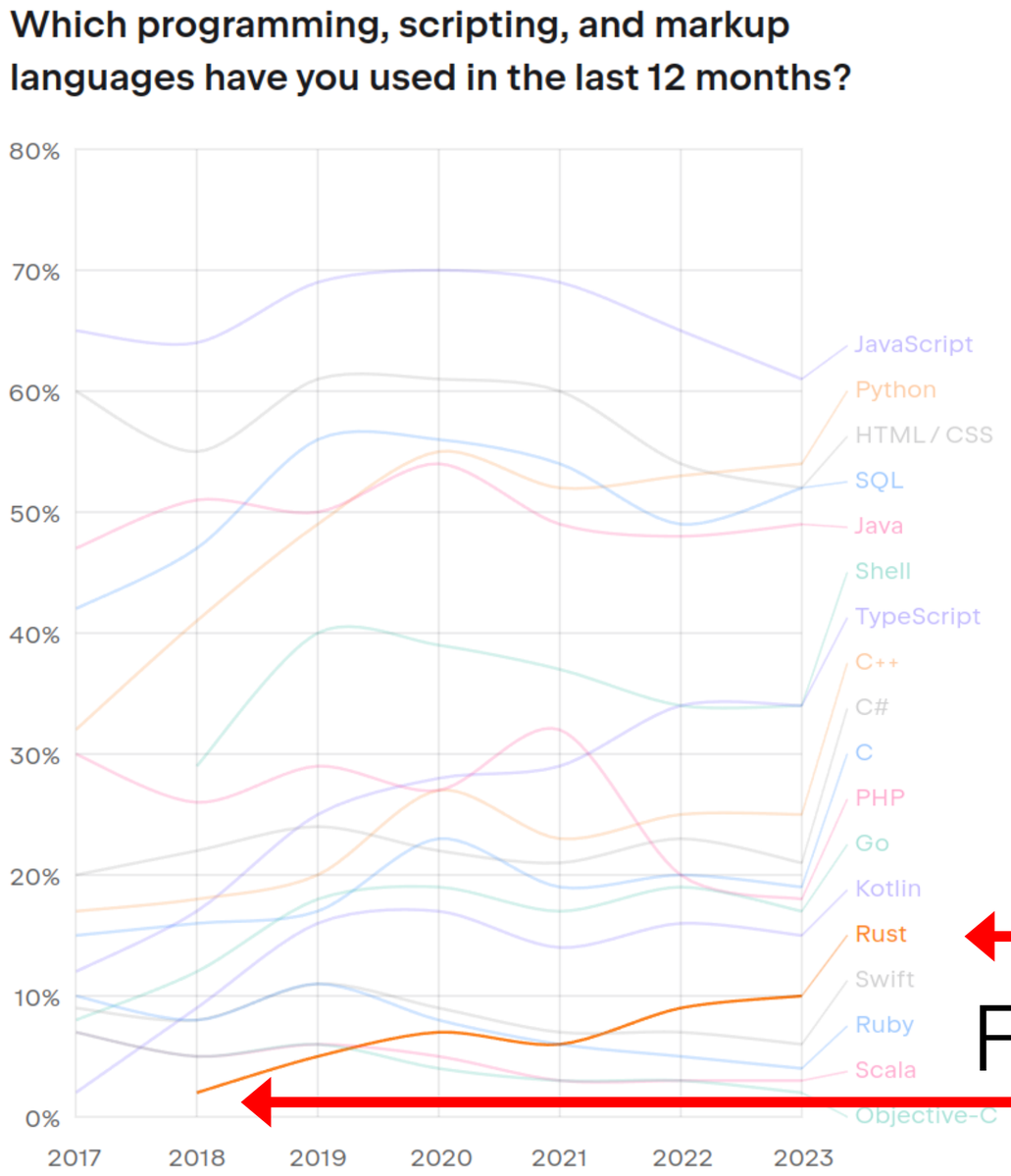
~12%

Source: Stack Overflow Developer Survey 2023



~10%

Source: The State of Developer Ecosystem in 2023 (JetBrains)



~10%

Five years ago: 2%

Source: The State of Developer Ecosystem in 2023 (JetBrains)

Package ecosystem size

Source: `crates.io`, `pypistats.org`, `all-the-package-names`

Package ecosystem size

- Rust ~140k

Source: `crates.io`, `pypistats.org`, `all-the-package-names`

Package ecosystem size

- Rust ~140k
- Python ~520k

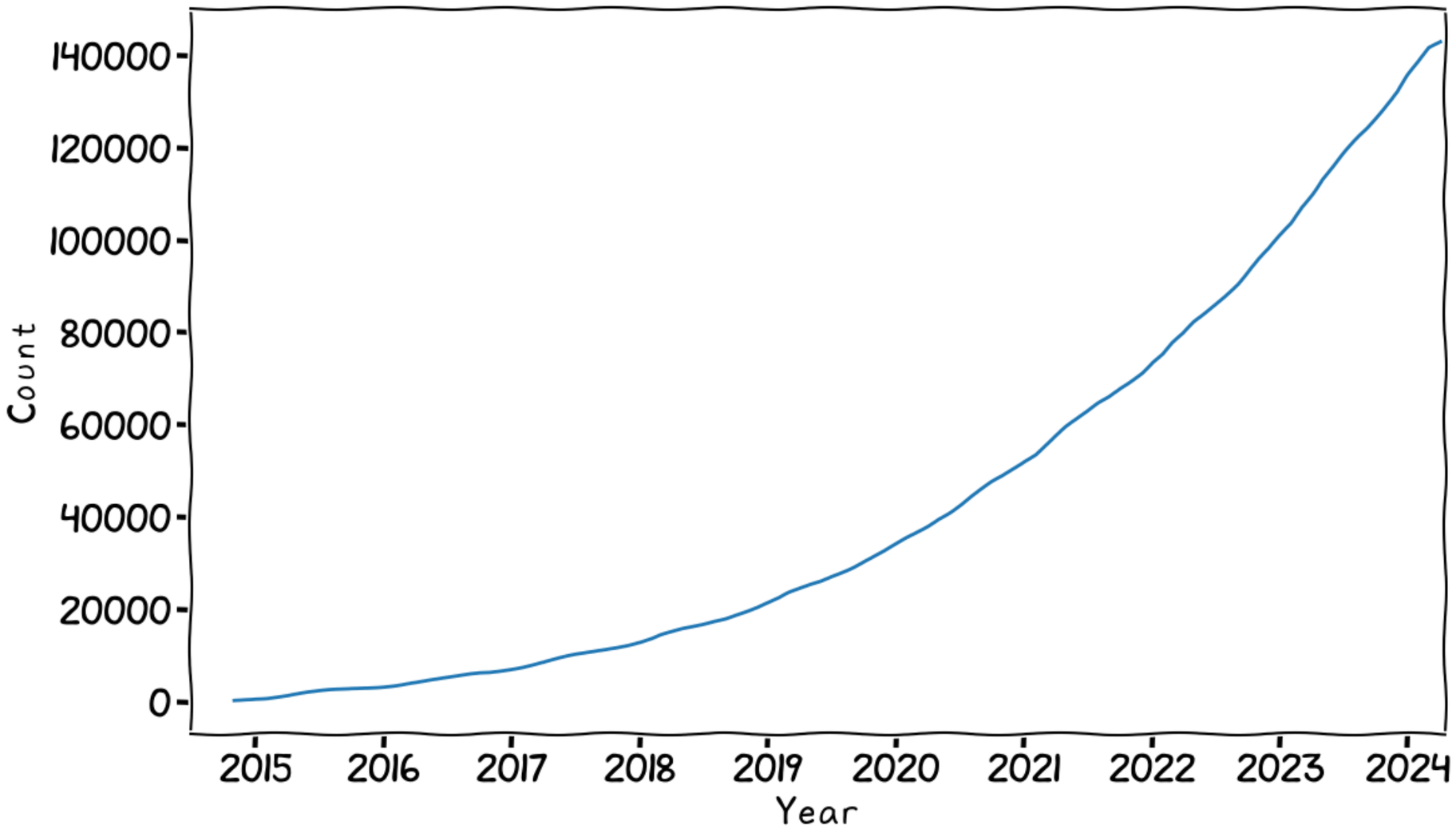
Source: `crates.io`, `pypistats.org`, `all-the-package-names`

Package ecosystem size

- Rust ~140k
- Python ~520k
- JavaScript ~2,800k

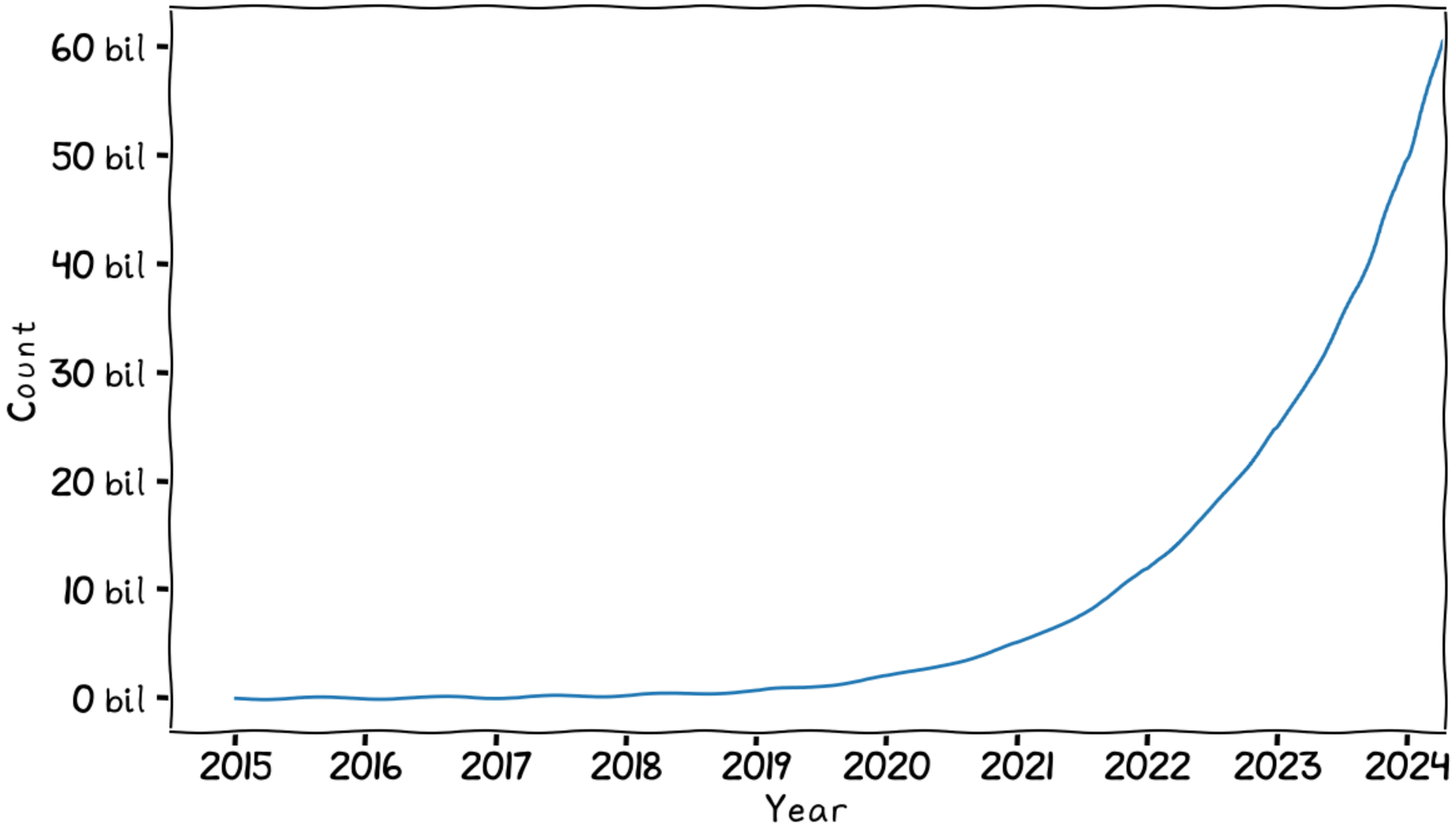
Source: `crates.io`, `pypistats.org`, `all-the-package-names`

Number of Rust packages



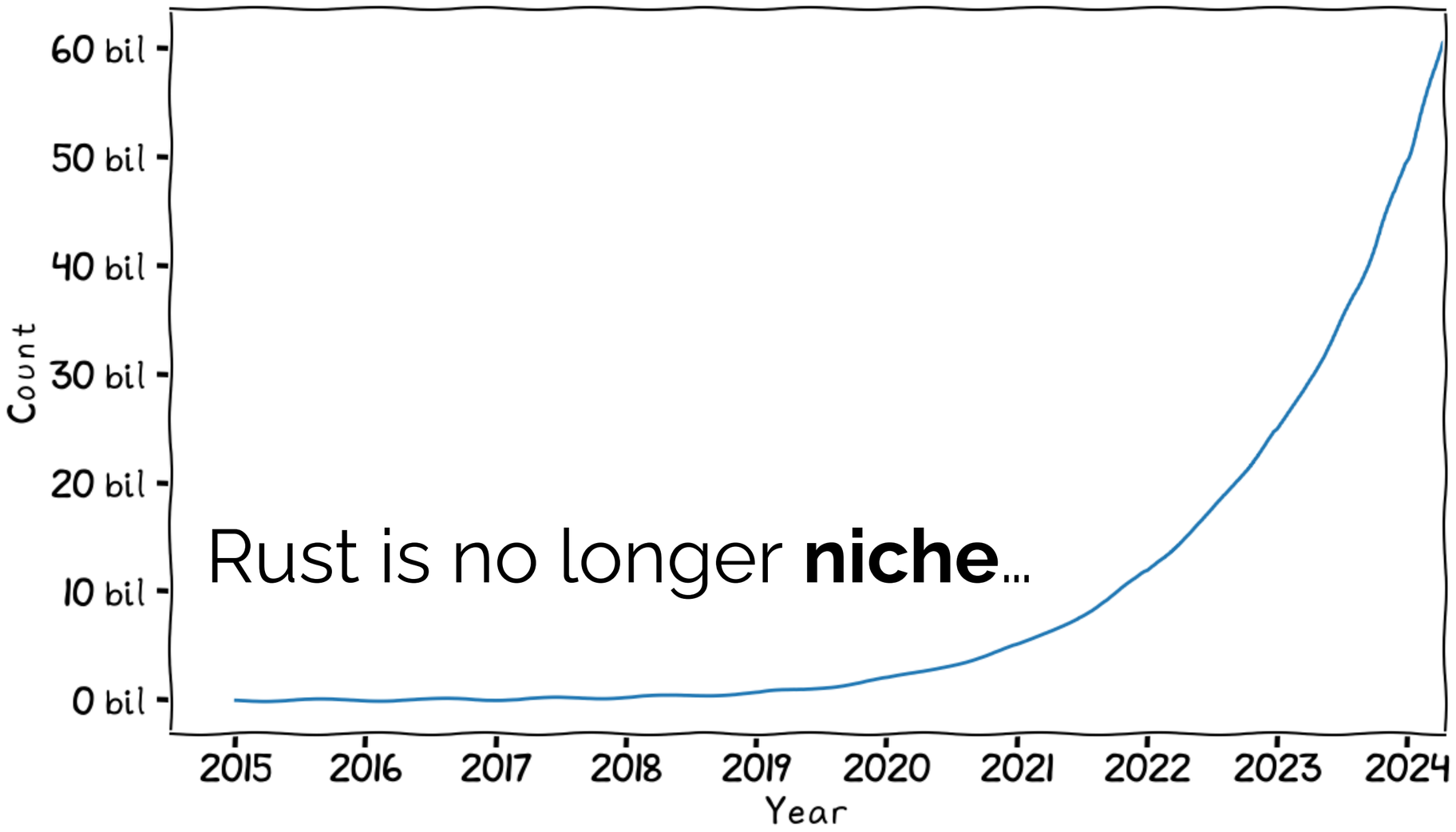
Source: `crates.io`

Rust package downloads



Source: `crates.io`

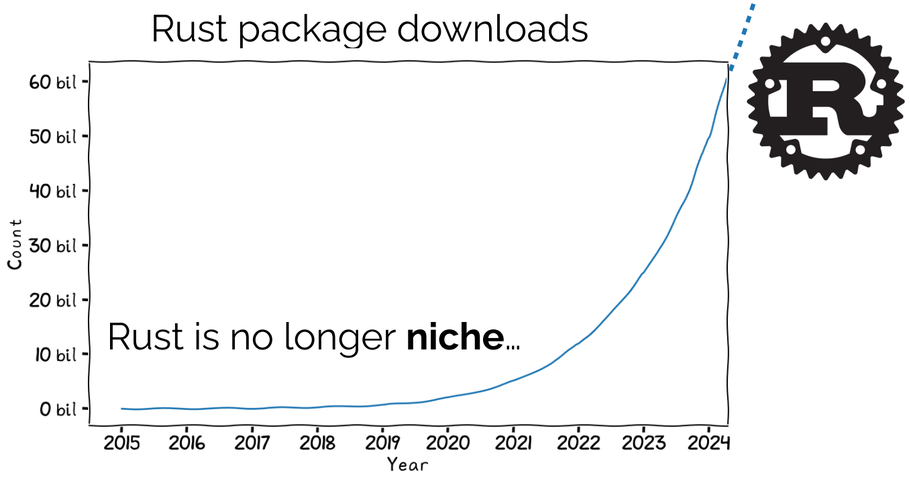
Rust package downloads



Rust is no longer **niche**...

Source: `crates.io`

…but it is not yet fully **mainstream**.



Source: `crates.io`

How Rust is Tilde's Competitive Advantage

Source: Rust Case Study: How Rust is Tilde's Competitive Advantage

Thank you for your attention!

Rust Annual Survey 2023:

Slides were created using `github.com/spirali/elsie`