Hi there!

I’m **Adrian Sieber**.
I work at **Feram**.
More Information at **adriansieber.com**.

History and Future of Rust

Who of you used …

- C: 6/12
- Go: 6/12
- OCaml: 0/12
- Elm: 1/12
- Haskell: 4/12
- PureScript: 0/12
- Rust: 11/12

Short History

2006 Graydon Hoare created Rust as a personal project while working at Mozilla.

2010 Rust was published

2010-06-16 First public commit

2010-06-24 Adding most of the source files

https://github.com/rust-lang/rust/commit/d6b7c96c3eb

- Rust was originally written in **OCaml**
- **Brendan Eich** – the creator of JavaScript – is listed as an author

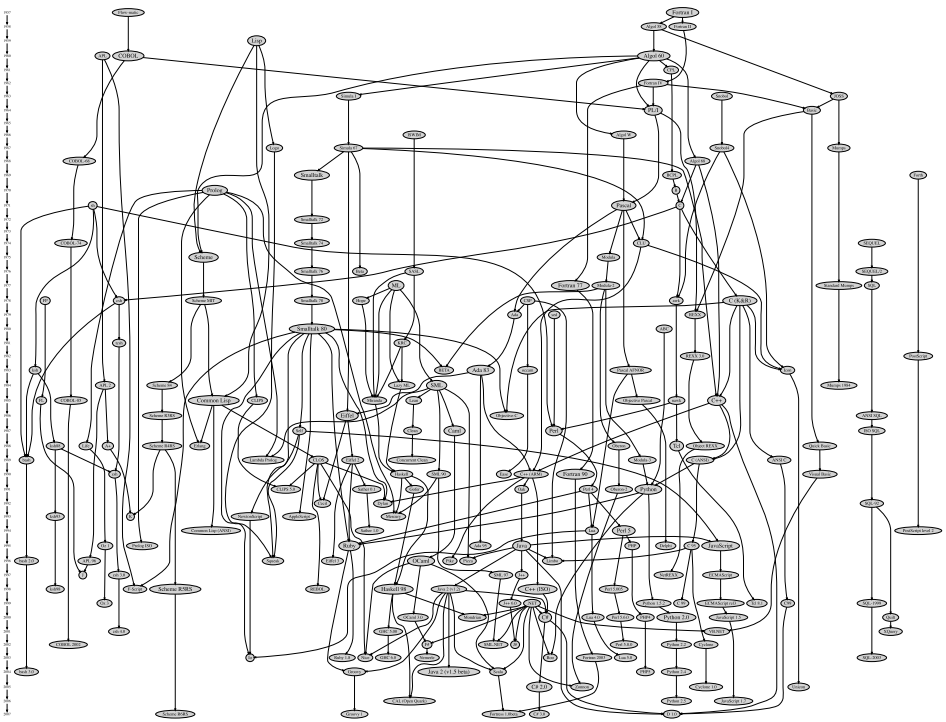
2010-07 Mozilla Annual Summit - Project Servo

- Rust is a language that mostly cribs from past languages. Nothing new.
- The syntax is, really, about the last concern.
- That was just a “taste” so you don’t get all frustrated wondering what it looks like and/or assume that at the last minute it’s going to read like

> Lisp or Haskell - (Hush, I know and love these languages, but there is a time and place).

2015-05-14 **Rust 1.0** was released (first stable version)

How Exactly Was it Influenced?



- **Meta Language Family**
  - **Standard ML**, **OCaml**, **Haskell**:
    - Algebraic data types (-> Enums)
    - Pattern matching
    - Type inference
  - **ML Kit**, **Cyclone**:
    - Region based memory management
  - **Haskell**:
    - Typeclasses (-> Traits)
    - Type families

- **Elm**

  > … the updated `--explain` messages draw heavy inspiration from the Elm approach.

- **C Family**
  - **C++**:
    - References
    - RAII (Resource acquisition is initialization)
    - Smart pointers

- Move semantics
- Monomorphization
- Memory model

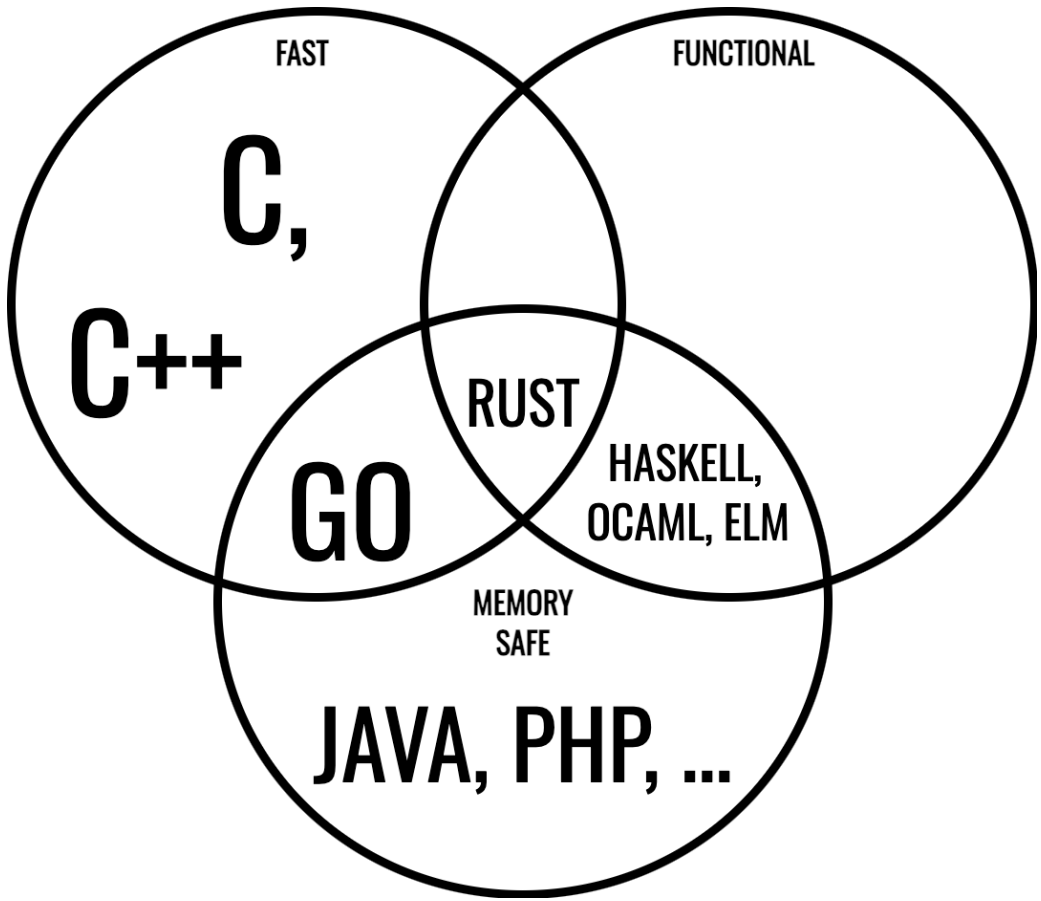
- **Newsqueak**, **Alef**, **Limbo**:
  - Channels
  - Concurrency

- **Erlang**:
  - Message passing
  - Thread failure
- **Swift**: Optional bindings
- **Scheme**: Hygienic macros
- **C#**: Attributes
- **Ruby**: Closure syntax

Takeaway

2 Main Influences: **ML Family** and **C Family**

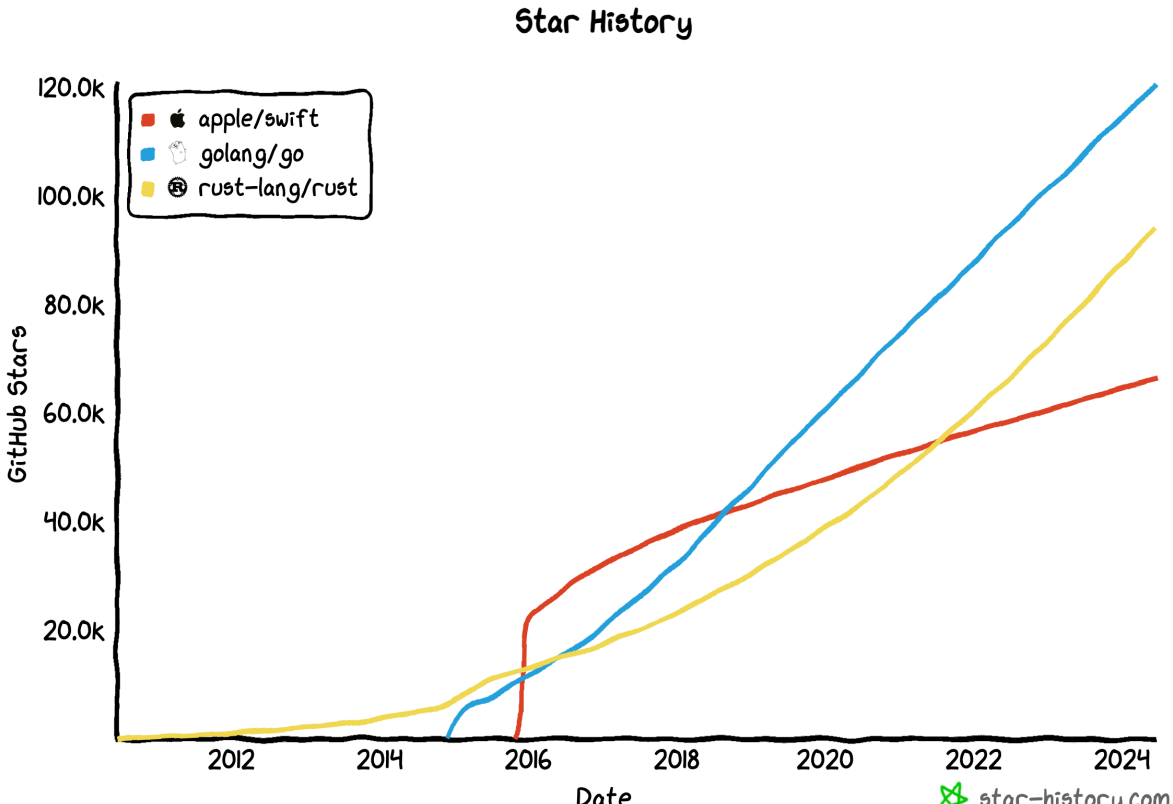
Cherry picking from other languages

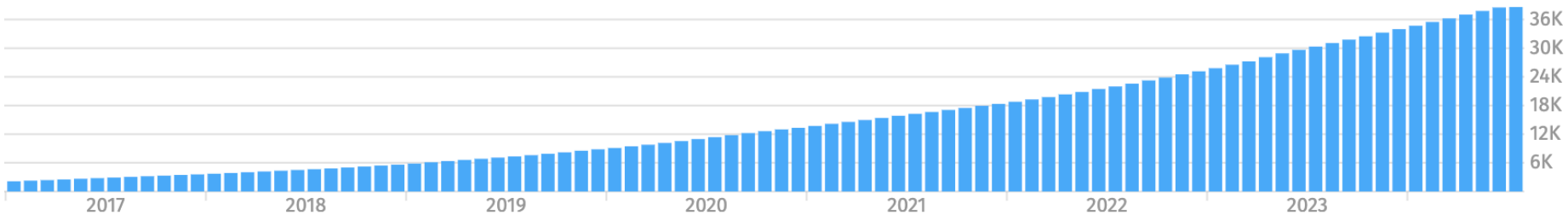

FAST
FUNCTIONAL
C,
C++
RUST
GO
HASKELL,
OCAML, ELM
MEMORY
SAFE
JAVA, PHP, ...

Other Influences

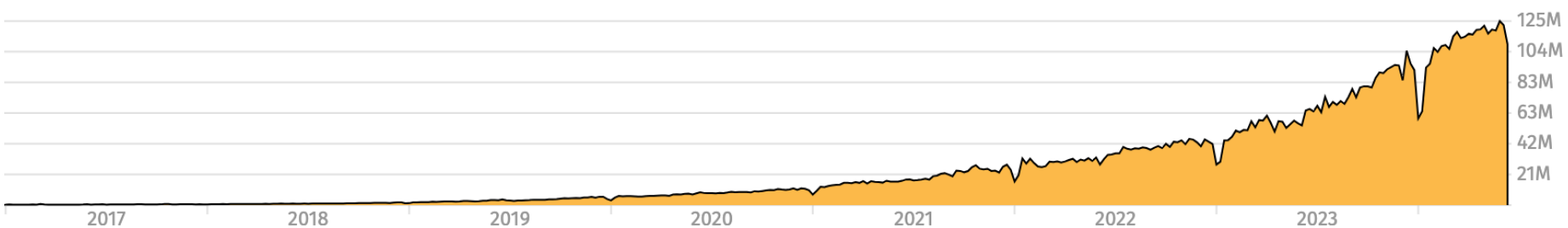
- npm, pip, bundler, … -> Cargo package manager
- `go fmt` -> Rustfmt
- Javadoc -> Rustdoc
- …

The Future of Rust


Star History
120.0k
100.0k
80.0k
60.0k
40.0k
20.0k
GitHub Stars
apple/swift
golang/go
rust-lang/rust
2012
2014
2016
2018
2020
2022
2024
Date
star-history.com



Number of users/teams owning a crate on crates.io



Daily downloads since Rust 1.0, 7-day average

Roadmap

Rust's goal is to **empower everyone to build reliable and efficient software**. Success requires not only designing and implementing a great language with great libraries and great tools, but also maintaining a great and supportive community.

Our focus for Rust 2024 is to **scale empowerment** in many different ways. As we grow, we face increasing challenges in how we can scale the ways in which we empower people to an increasing number of people. This roadmap presents three general themes we plan to focus on:

- **Flatten the (learning) curve**: scaling to new users and new use cases
  - Make Rust more accessible to new and existing users alike, and make solving hard problems easier.
- **Help Rust's users help each other**: scaling the ecosystem
  - Empower library authors so they can---in turn---empower their users.
- **Help the Rust project scale**: scaling the project
  - Develop processes to scale to the needs and use cases of a growing number of users; evaluate and finish projects we've started.

… but also

Rust's speed and resource efficiency comes at a cost:

- Hard to learn and master
- Development is slower
- Thinking about often unimportant details
  (lifetimes, borrowing, memory management, …)

Rust as a gateway drug to Haskell

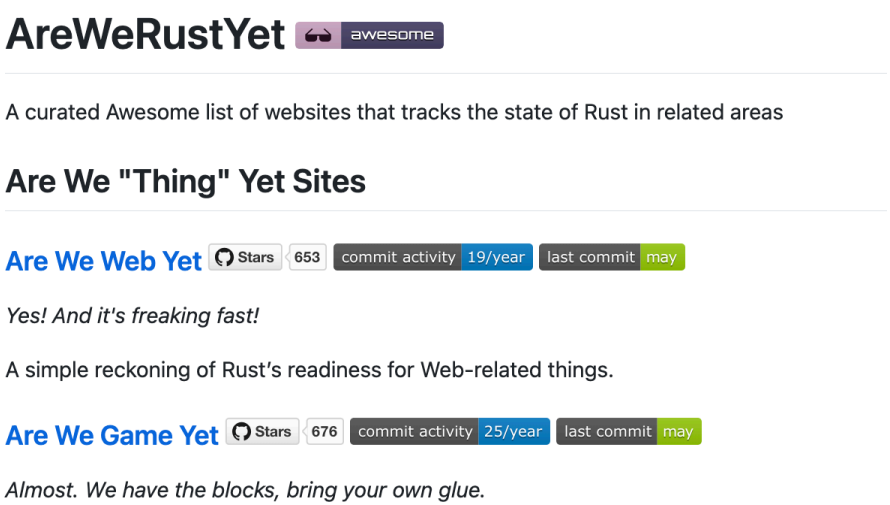
Posted on Tue 13 June 2017 in Programming

For work-related reasons, I had to recently get up to speed on programming in Haskell.

Before that, I had very little actual experience with the language, clocking probably at less than a thousand lines of working code over a couple of years. Nothing impressive either: some wrapper script here, some experimental rewrite there…

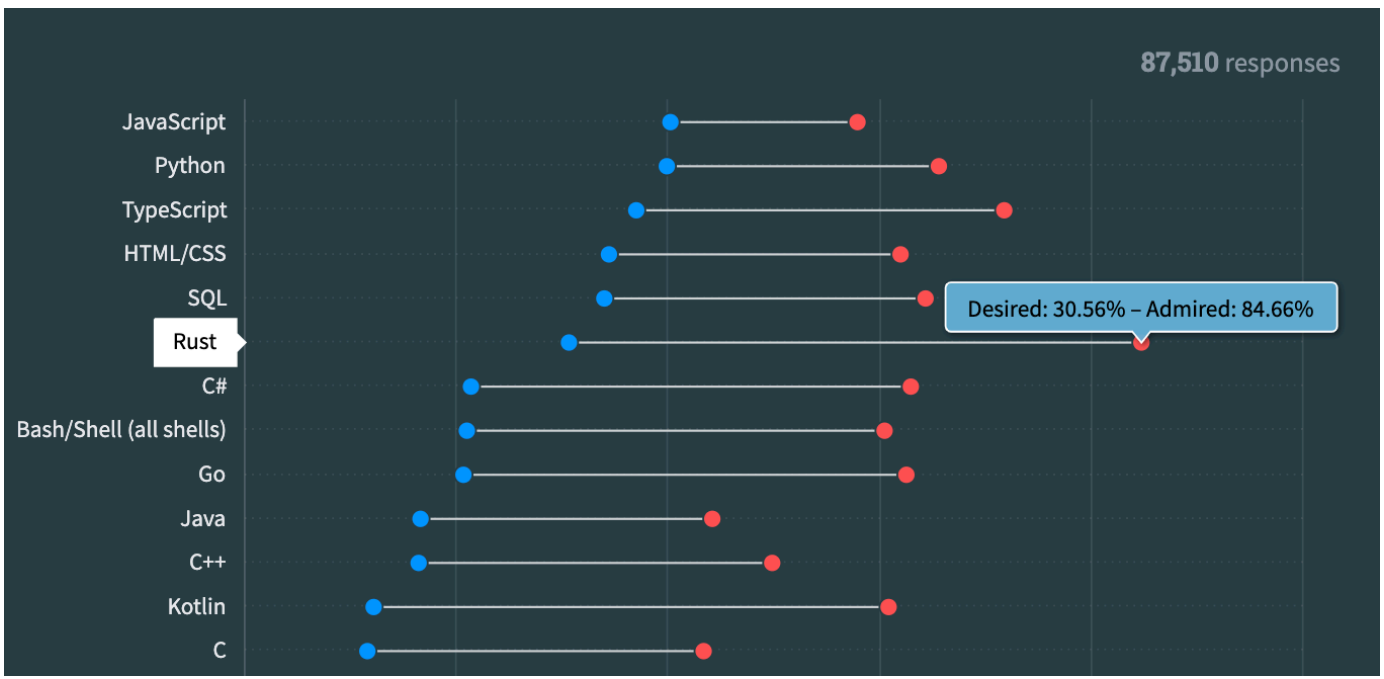
Some Areas are Still Immature

github.com/UgurcanAkkok/AreWeRustYet



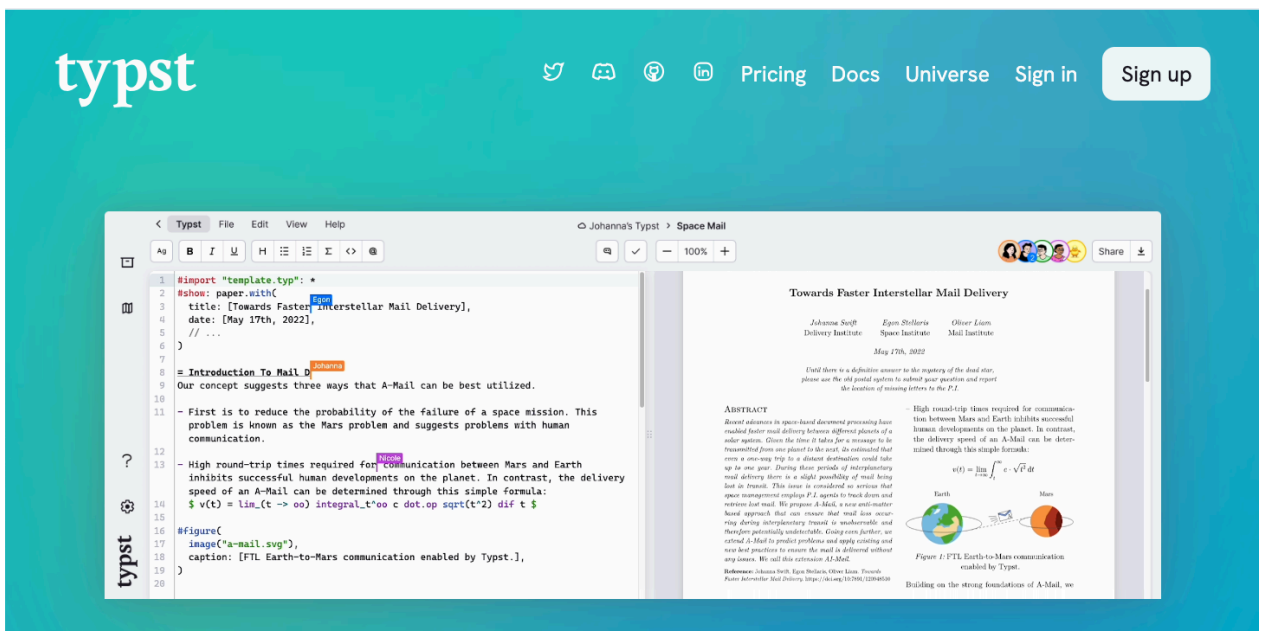
- Game development
- Machine Learning
- GUI frameworks
- Web development
- Embedded systems

Rust has been the most admired programming language for 8 years in a row



FYI: This presentation was built with Typst

An open source LaTeX successor written in Rust



Thank you for your attention!

I’m **Adrian Sieber**.
I work at **Feram**.
More Information at **adriansieber.com**.