



RED BADGER

Rust in the Enterprise

**Taking control with a modern,
sustainable approach
to software engineering**

2023-07-20

Stuart Harris, Founder & Chief Scientist

Viktor Charypar, Technical Director



What are we talking about today?

- How does Rust help solve the problems that Enterprises care about?
 - What do we mean by a modern, sustainable approach?
- Where does the resistance to adopting Rust come from?
- How do we go about introducing Rust into an Enterprise?
- Beer and Pizza
- Crux overview
 - How Rust can help an Enterprise build high quality, sustainable digital product, cheaper and faster.
- Crux deep dive. Demos. Q&A.

Stu

- Software engineer
- Founder of Red Badger



RED BADGER



Viktor

- Software engineer
- Tech Director at Red Badger



RED BADGER



What do enterprises care about?

- **Quality** — they want to build software that works
- **Sustainability** — they want to build software that lasts
- **Security** — they want to be secure
- **Cost/Speed** — they want it yesterday, and within budget
- **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

What do enterprises care about?

-  **Quality** — they want to build software that works
- **Sustainability** — they want to build software that lasts
- **Security** — they want to be secure
- **Cost/Speed** — they want it yesterday, and within budget
- **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Quality

Software that is *delightful*

This means:

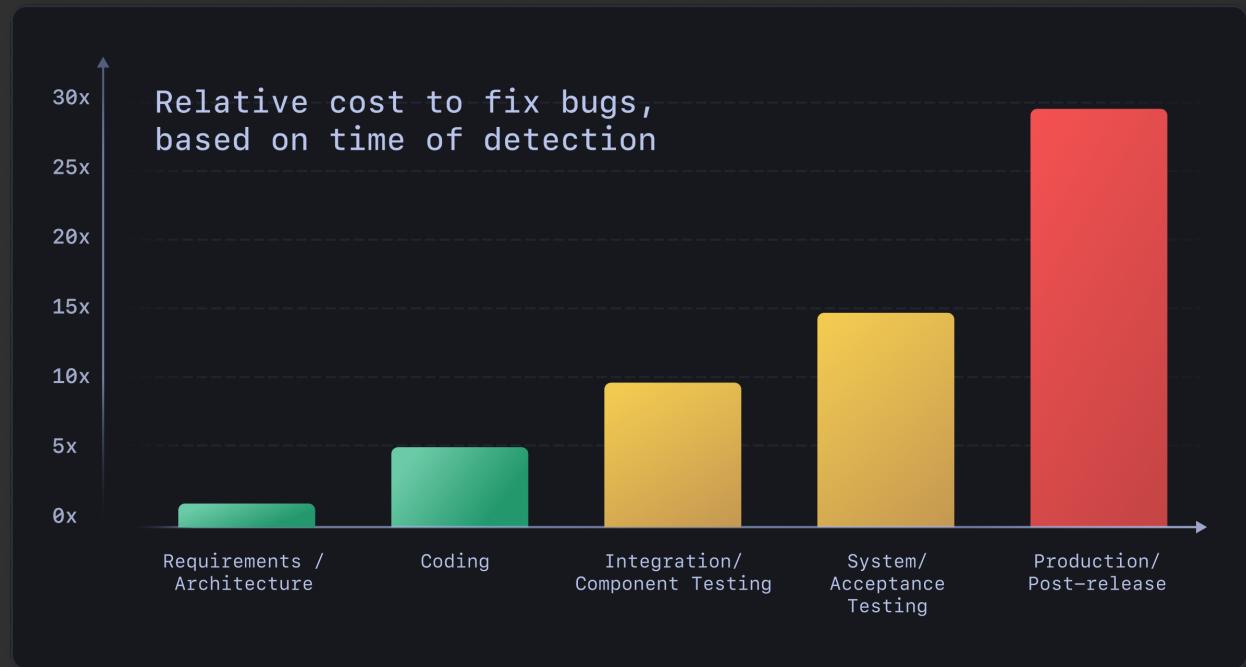
- It is well designed (from a UX perspective)
- It does what it is supposed to do
- It is easy to use and accessible for everyone
- It is fast, reliable and secure
- It is maintainable, extensible and testable



Shifting left

**Remove the long tail of bugs
that are expensive to fix**

"Rust is the language where you
have the hangover first!"



<https://deepsource.com/blog/exponential-cost-of-fixing-bugs>

So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
- **Security** — they want to be secure
- **Cost/Speed** — they want it yesterday, and within budget
- **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Sustainability

Software that is easy to maintain and extend

This means:

- It is easy to understand
- It is easy to test
- It is easy to deploy
- It is easy to change



Where does your brain power go?

"Free up your teams to work on the most important problems — those that are *unique* to your business"

- Rust's guardrails prevent us wasting time on bugs
- Rust's toolchain is the best there is
 - removes bike-shedding
 - reduces yak-shaving
- A large codebase won't descend into chaos
- "It just works"™
 - no long tail — means you can leave it alone



Sustainable Software Engineering

"Everyone has a part to play."

"Sustainability is enough, all by itself, to justify our work."

- Find as many "Carbon Proxies" as you can, e.g.
 - Time-to-interactive and Page weight
 - Average server response time
 - Cost of your services
 - The utilisation of your servers

	Energy	Time	Mb
(c) C	1.00	1.00	1.00
(c) Rust	1.03	1.04	1.05
(c) C++	1.34	1.56	1.17
(c) Ada	1.70	1.85	1.24
(v) Java	1.98	1.89	1.34
(c) Pascal	2.14	2.14	1.47
(c) Chapel	2.18	2.18	1.54
(v) Lisp	2.27	2.83	1.92
(c) Ocaml	2.40	3.02	2.45
(c) Fortran	2.52	3.09	2.57
(c) Swift	2.79	3.14	2.71
(c) Haskell	3.10	3.40	2.80
(v) C#	3.14	3.55	2.82
(c) Go	3.23	4.20	2.85
(i) Dart	3.83	4.20	3.34
(v) F#	4.13	6.30	3.52
(i) JavaScript	4.45	6.52	3.97
(v) Racket	7.91	6.67	4.00
(i) TypeScript	21.50	11.27	4.25
(i) Hack	24.02	26.99	4.59
(i) PHP	29.30	27.64	4.69
(v) Erlang	42.23	36.71	4.90
(i) Lua	45.98	43.44	6.01
(i) Jruby	46.54	46.20	6.62
(i) Ruby	69.91	59.34	6.72
(i) Python	75.88	65.79	7.20
(i) Perl	79.58	71.90	8.64
(i) Dart		82.91	19.84
(i) Lua			

sustainability with rust

sustainable software engineering overview

how to measure and reduce the carbon footprint of your application

So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
-  **Security** — they want to be secure
- **Cost/Speed** — they want it yesterday, and within budget
- **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Security

Software that is secure by design

This means:

- It is easy to reason about
- It is solid and reliable
- It is built on a secure foundation
- It is open



So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
-  **Security** — they want to be secure
-  **Cost/Speed** — they want it yesterday, and within budget
- **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Cost/Speed

"Rust will speed up your development and reduce your costs"

5 Rumours about Rust

1. Rust takes more than 6 months to learn – Debunked! 😊
2. The Rust compiler is not as fast as people would like – Confirmed! 😭
3. Unsafe code and interop are always the biggest challenges – Debunked! 😊
4. Rust has amazing compiler error messages – Confirmed! 😍
5. Rust code is high quality – Confirmed! 😎

from over 1,000 Google developers

<https://opensource.googleblog.com/2023/06/rust-fact-vs-fiction-5-insights-from-googles-rust-journey-2022.html>

Time until confident writing Rust

Still ramping up

8.6%

More than 4 months

9.0%

3 - 4 months

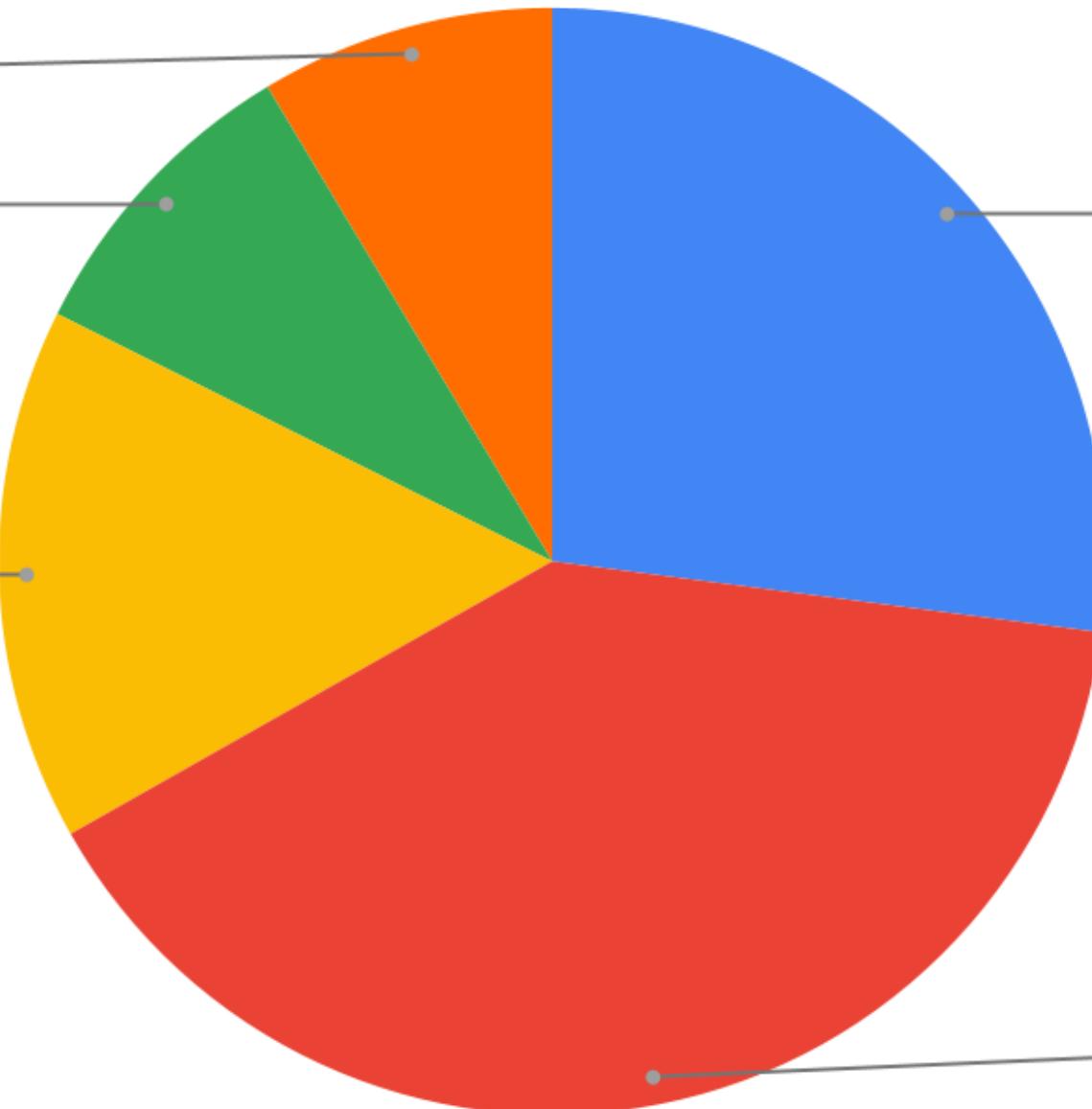
15.6%

2 - 3 weeks

27.0%

1 - 2 months

39.8%



So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
-  **Security** — they want to be secure
-  **Cost/Speed** — they want it yesterday, and within budget
-  **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
- **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Control/Risk/Compliance

- Rust is a trusted, open source language
- you can write software that is
 - memory safe
 - safe from data races and concurrency bugs
 - easy to test
 - easy to audit and maintain
- a great fit for regulated industries



So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
-  **Security** — they want to be secure
-  **Cost/Speed** — they want it yesterday, and within budget
-  **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
-  **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Innovation/Talent/Culture

"Managers need to know that their hard-won employees are gainfully employed and are not frustrated, bored, or flight risks"

- attract and retain talent — become a "destination"
- build a culture of engineering excellence
- Rust helps you become a better software engineer
- "Most loved/desired" on Stack Overflow for 8 years in a row



So what do enterprises care about?

-  **Quality** — they want to build software that works
-  **Sustainability** — they want to build software that lasts
-  **Security** — they want to be secure
-  **Cost/Speed** — they want it yesterday, and within budget
-  **Control/Risk/Compliance** — they want to be in control of their software, reduce risk, and be compliant with regulations
-  **Innovation/Talent/Culture** — they want to innovate, attract and retain talent, and build a culture of engineering excellence

Where does the resistance come from?

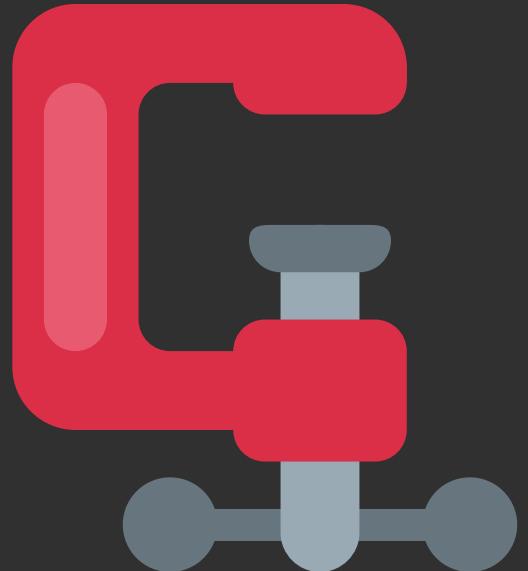
- **Risk** — "We don't want to be the first to do this"
- **Cost** — "We don't have the budget"
- **Speed** — "We don't have the time"
- **Culture** — "We don't have the talent"



| This is all FEAR talking — FUD!

Pincer movement!

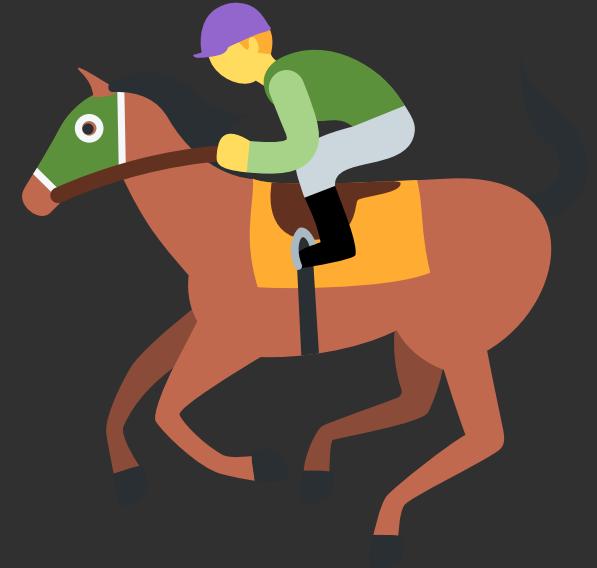
- **Top down** — "We need to do this"
- **Bottom up** — "We want to do this"



How do we get there?

Top down

- Talk to the CIO. Find a buddy to help you
- Find the open source owner (OSPO)
- Find a champion (and a project goal)
 - Might not be a technology champion
 - Would be great to have a product sponsor
- Run a 2-horse race



Skunk Works

a group within an organization
given a high degree of autonomy
and unhampered by bureaucracy,
with the task of working on
advanced or secret projects



"We don't want to be first, but we can't be third!"

But if you're not starting now, you'll already be behind when
it matters

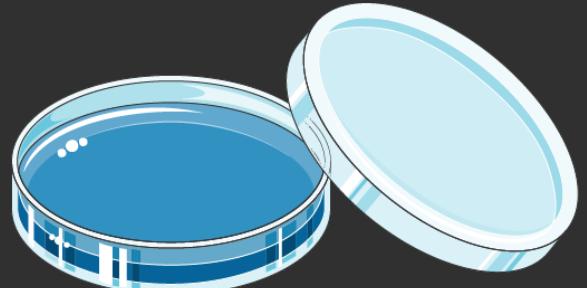


How do we get there?

Bottom up

- Learn it yourself
- Write some tools
- Start small and stay small
- Build a mini community, hold meetups
- Infect your organisation

Think of a petri dish — a culture pops up in a few places and then eventually it's everywhere



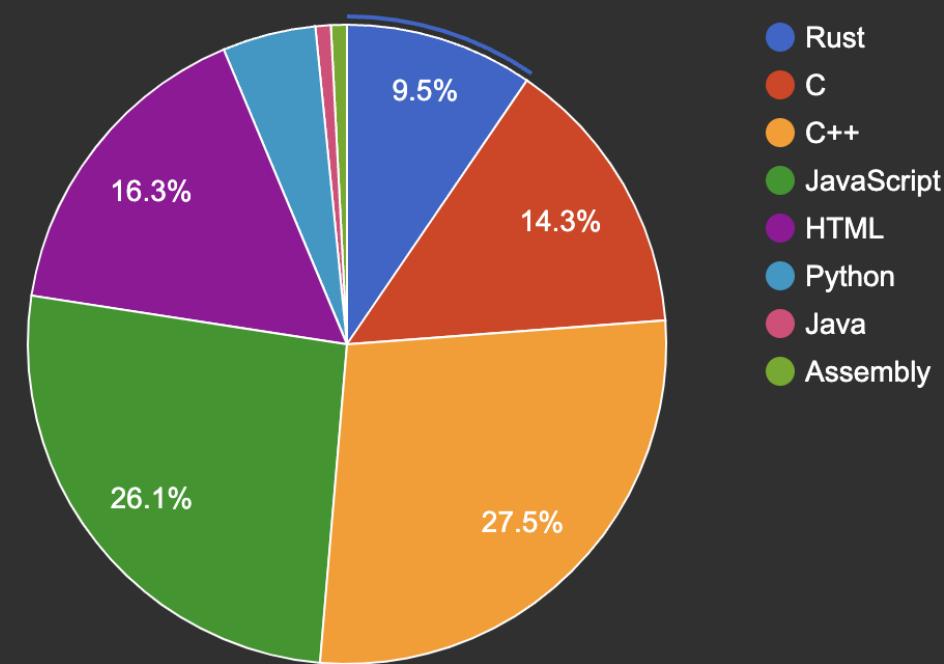
Also...

Don't "Rewrite it in Rust"™!

Rust was born at Mozilla in 2009...

...but in 2023, Firefox is still less than 10% Rust

Firefox languages in SLOC



And finally...

Rust and WebAssembly

- Rust and Wasm are a perfect match
- Wasm is becoming a game-changer for cloud native
- The sandboxed execution environment is a great fit for enterprise



Something to think about...

Rust and Copilot

Despite being risk averse, enterprises need the productivity gains of AI coding assistants...

...thank goodness the Rust compiler is strict and precise!



Let's have a break!





Let's talk Crux!

Crux is a great example of how
Rust can help the enterprise build
better quality software with less
effort (cost) and more Joy™

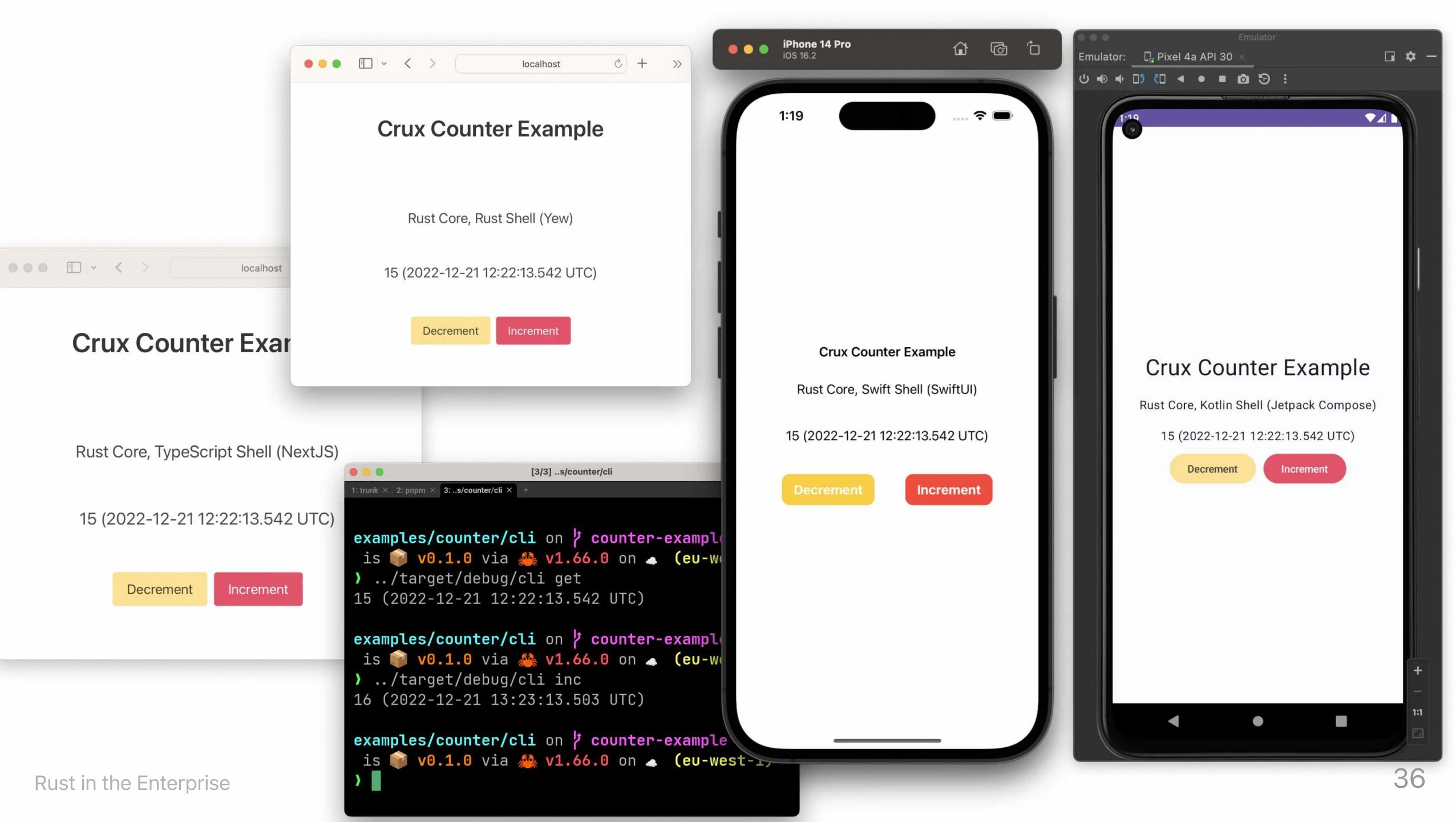


But first...



Or 17ms...

```
crux/examples/notes on ⚡ remix [!] via 🐀 v1.71.0 on ⚡ (eu-west-1)
❯ cargo nextest run
    Finished test [unoptimized + debuginfo] target(s) in 0.06s
    Starting 25 tests across 3 binaries
        PASS [  0.005s] shared app::editing_tests::removes_selection_on_backspace
        PASS [  0.006s] shared app::editing_tests::inserts_text_at_cursor_and_renders
        PASS [  0.006s] shared app::editing_tests::changes_selection
        PASS [  0.006s] shared app::editing_tests::handles_emoji
        PASS [  0.005s] shared app::editing_tests::renders_text_and_cursor
        PASS [  0.006s] shared app::editing_tests::removes_character_after_cursor
        PASS [  0.006s] shared app::editing_tests::removes_selection_on_delete
        PASS [  0.006s] shared app::editing_tests::removes_character_before_cursor
        PASS [  0.006s] shared app::editing_tests::moves_cursor
        PASS [  0.007s] shared app::editing_tests::replaces_empty_range_and_renders
        PASS [  0.005s] shared app::editing_tests::replaces_range_and_renders
        PASS [  0.005s] shared app::note::test::splices_text
        PASS [  0.005s] shared app::save_load_tests::creates_a_document_if_it_cant_open_one
        PASS [  0.006s] shared app::editing_tests::replaces_selection_and_renders
        PASS [  0.005s] shared app::note::test::inserts_text
        PASS [  0.005s] shared app::save_load_tests::opens_a_document
        PASS [  0.006s] shared app::save_load_tests::starts_a_timer_after_an_edit
        PASS [  0.007s] shared app::save_load_tests::saves_document_when_typing_stops
        PASS [  0.007s] shared app::sync_tests::concurrent_clean_edits
        PASS [  0.007s] shared app::sync_tests::concurrent_conflicting_edits
        PASS [  0.005s] shared app::sync_tests::one_way_sync
        PASS [  0.005s] shared app::sync_tests::receiving_own_edits
        PASS [  0.005s] shared app::sync_tests::two_way_sync
        PASS [  0.005s] shared app::sync_tests::remote_insert_behind_cursor
        PASS [  0.005s] shared app::sync_tests::remote_delete_moves_cursor
-----
Summary [  0.017s] 25 tests run: 25 passed, 0 skipped
```





iOS, Android and Web apps in Rust

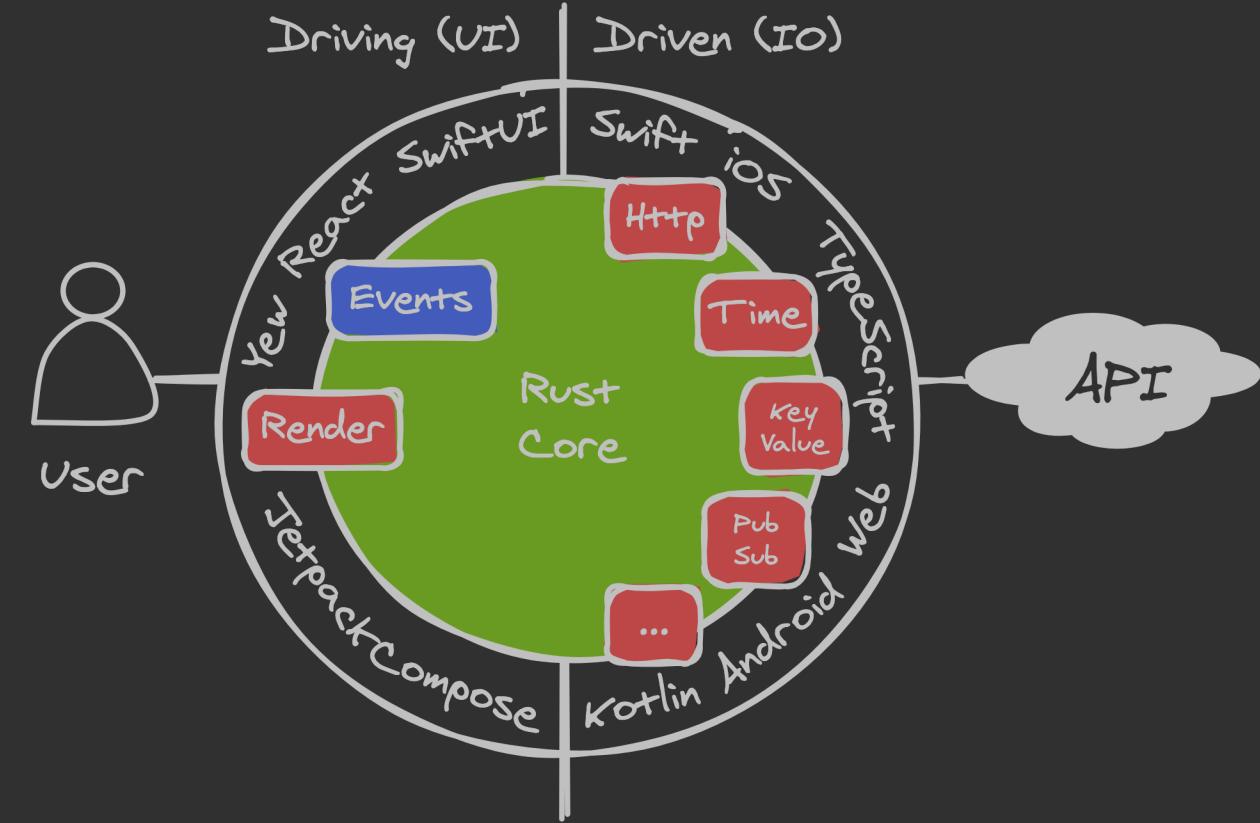
a.k.a. Headless apps

17.feb.2023

Stuart Harris

Founder & Chief Scientist, Red Badger

Rust in the Enterprise



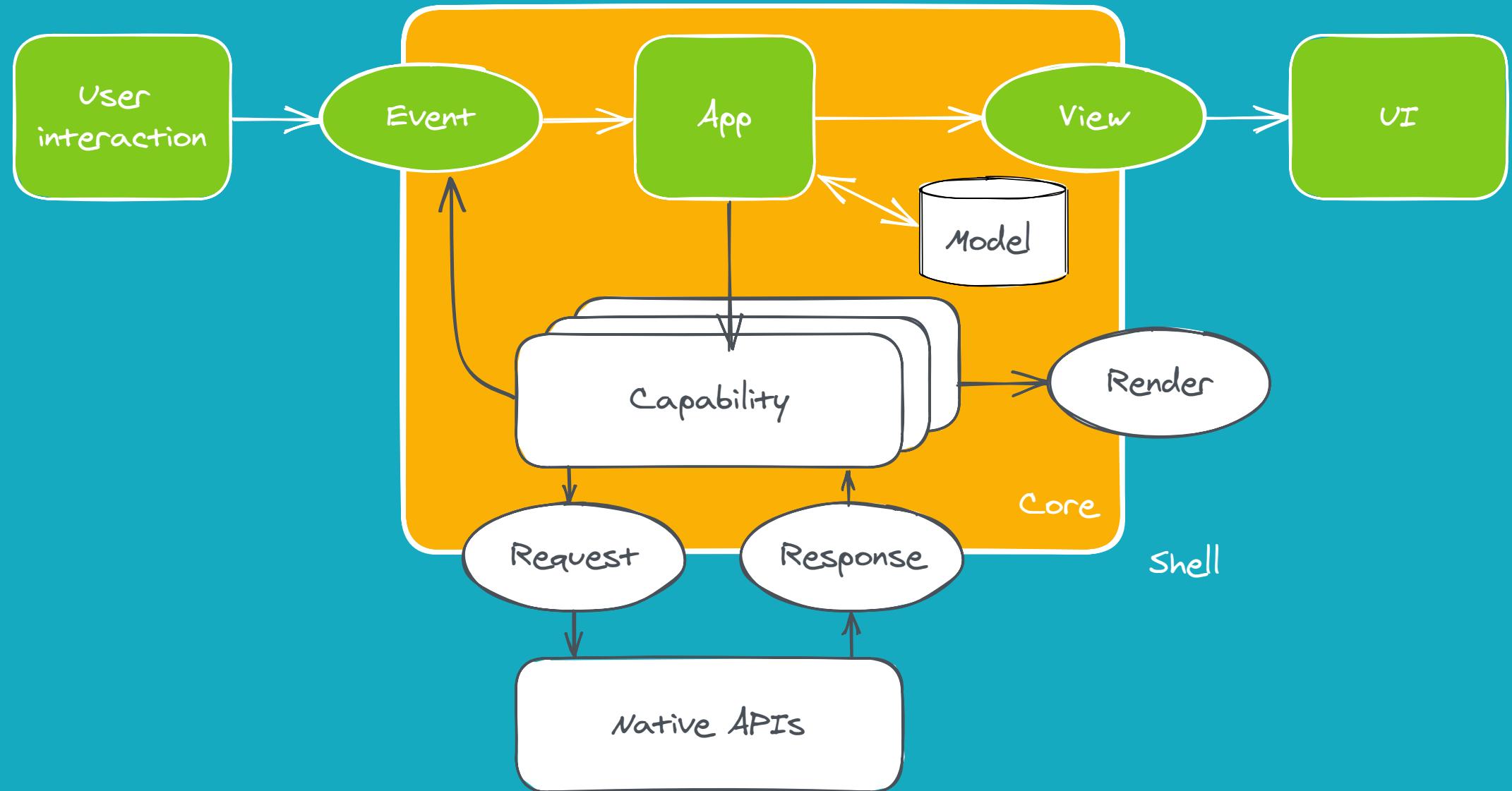


Rust Nation '23



Building a multi-platform app (don't @ me!)

	Platform Native	Kotlin MM	React Native	Capacitor Ionic	Flutter	Crux
Native UX	✓	✓	😐	✗	✗	✓
Web?	✗	😐	😐	✓	✓	✓
Development	😐	✓	😐	✓	✓	✓
Testing	😐	😐	😱	😱	😐	🤩
Maintenance	😐	✓	😡	😡	✓	✓
Effort	3x	2x	2x	1.5x	1.4x	1.4x



Deep dive into Crux



Thank you!



Why Rust?

- <https://aws.amazon.com/blogsopensource/sustainability-with-rust/>
- <https://www.wired.com/story/rust-secure-programming-language-memory-safe/>
- <https://content.red-badger.com/resources/how-to-use-rust-to-build-networked-services>
- <https://www.technologyreview.com/2023/02/14/1067869/rust-worlds-fastest-growing-programming-language>
- <https://www.techspot.com/news/97654-how-broken-elevator-led-one-most-loved-programming.html>

Rust adoption in the enterprise

- <https://www.infoq.com/presentations/rust-adoption-journey/>
- <https://rustmagazine.org/issue-1/2022-review-the-adoption-of-rust-in-business/>
- <https://vercel.com/blog/turborepo-migration-go-rust>
- <https://opensource.googleblog.com/2023/06/rust-fact-vs-fiction-5-insights-from-googles-rust-journey-2022.html>
- <https://www.infoq.com/articles/rust-ecosystem-review-2023>

Crux

- <https://github.com/rebdar/crux>
- <https://rebdar.github.io/crux/>
- <https://red-badger.com/crux>
- https://docs.rs/crux_core/latest/crux_core/
- <https://www.youtube.com/watch?v=cWCZms92-1g&t=5s>