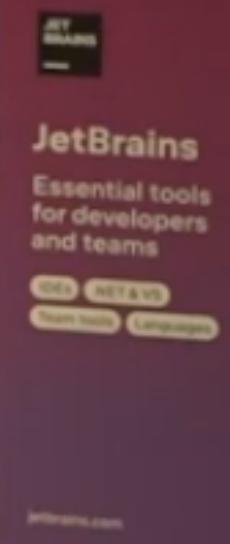
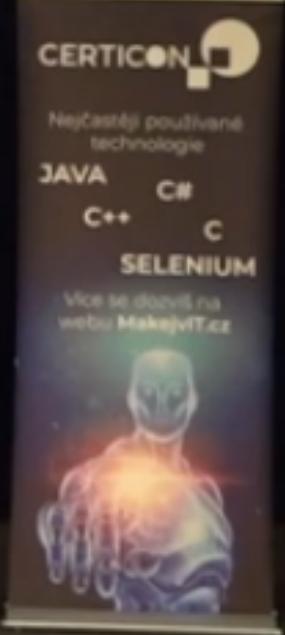


How Rust does open-source



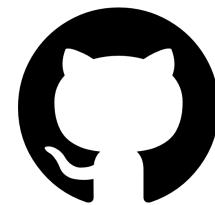
Co vám může nabidnout Rust?

[Představení](#)



Kuba Beránek

- Teaching @ VSB-TUO
- Research @ IT4Innovations
- Open-source contributor @ Rust Project



github.com/kobzol

Compiler performance working area

Improving rustc compilation performance (build times)

Members



Mark Rousskov

GitHub: [Mark-Simulacrum](#)

Team leader



Jakub Beránek

GitHub: [Kobzol](#)



Nicholas Nethercote

GitHub: [nnethercote](#)



Zoxc

GitHub: [Zoxc](#)



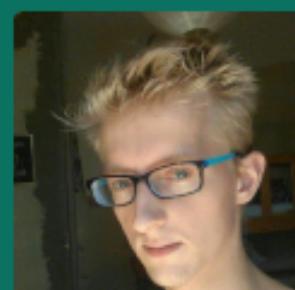
James

GitHub: [Jamesbarford](#)



Rémy Rakic

GitHub: [lqd](#)



Matyáš Racek

GitHub: [panstromek](#)

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



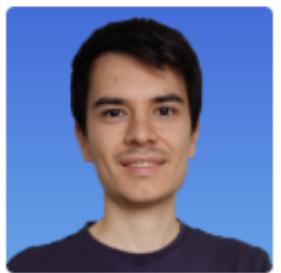
Jieyou Xu

GitHub: [jieyouxu](#)



Jakub Beránek

GitHub: [Kobzol](#)



Marco Ieni

GitHub: [marcoieni](#)



Mark Rousskov

GitHub: [Mark-Simulacrum](#)



Pietro Albini

GitHub: [pietroalbini](#)

Leadership council

Charged with the success of the Rust Project as whole, consisting of representatives from top-level teams

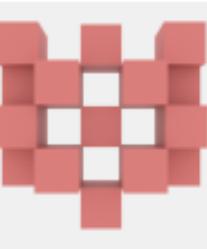
Members



Josh Stone

GitHub: [cuviper](#)

Compiler team



Eric Huss

GitHub: [ehuss](#)

Dev tools team



James Munns

GitHub: [jamesmunns](#)

Launching pad



Jakub Beránek

GitHub: [Kobzol](#)

Infrastructure team



Mara Bos

GitHub: [m-ou-se](#)

Library team



Oliver Scherer

GitHub: [oli-obk](#)

Moderation team



TC

GitHub: [traviscross](#)

Language team



Governance



Governance



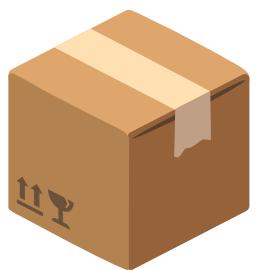
Development



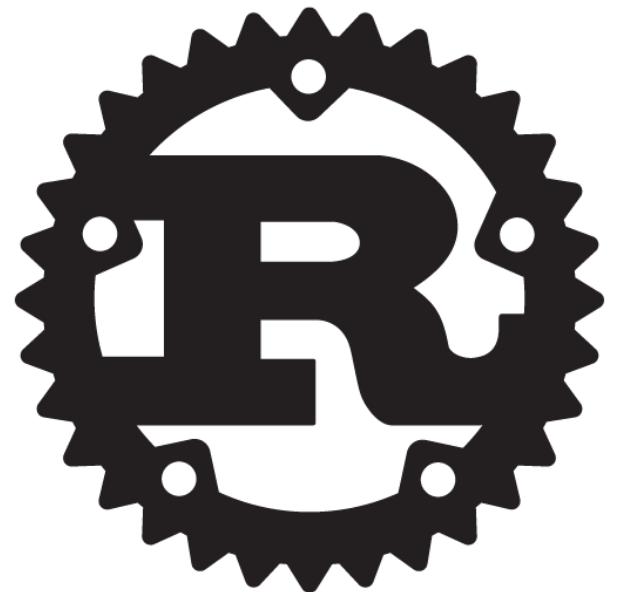
Governance



Development



Deployment



Governance



Development



Deployment



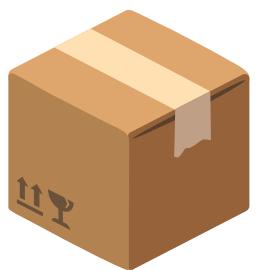
Maintenance



Governance



Development



Deployment



Maintenance



Open-source



Governance



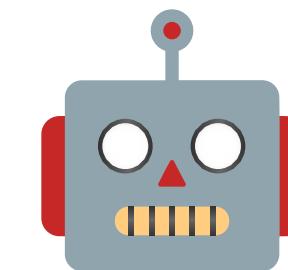
Development



Deployment



Maintenance



Open-source

Automation



Governance



Governance

...what's that?!



Governance

...what's that?!

Decision-making process



Governance

...what's that?!

Decision-making process
designed to evolve <something>



Governance

...what's that?!

Decision-making process
designed to evolve <a programming language>



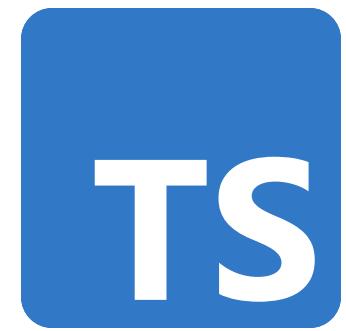
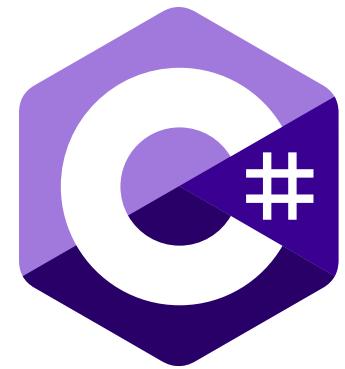
Governance

...what's that?!

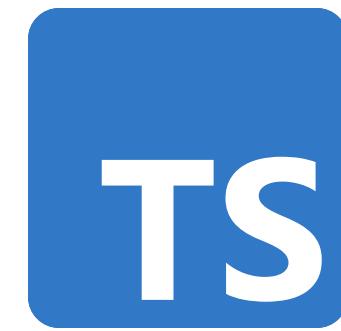
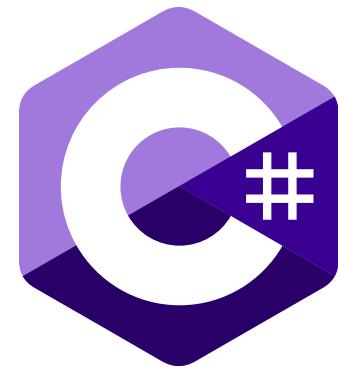
Decision-making process
designed to evolve <a programming language>

Have you ever tried to *change* your favourite programming language?

Company-backed

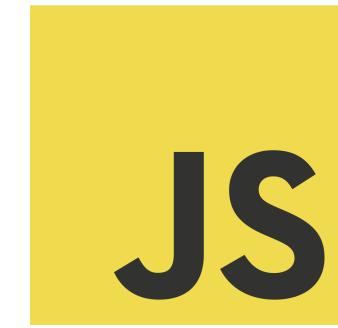


Company-backed

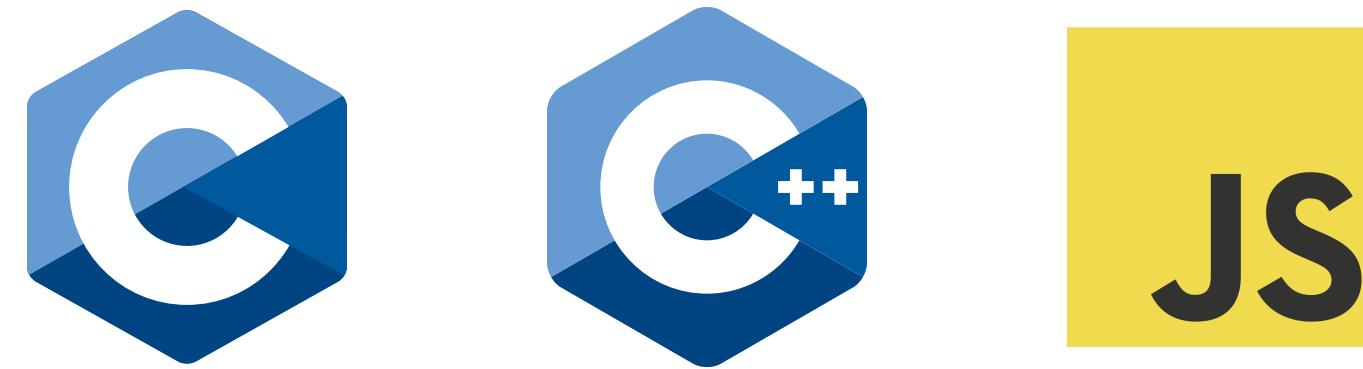


...get hired by Microsoft/Google (?)

Design by committee



Design by committee



(ISO) standardization process

C++ ISO standardization process (shortened version)



C++ ISO standardization process (shortened version)

1. Get invited to a mailing list

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group
5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group
5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group
5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
7. Find a champion to support your paper

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group
5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
7. Find a champion to support your paper
8. Attend a committee meeting
 - as a guest, or
 - by becoming an ISO National Body representative, or
 - by joining on behalf of a company (1-10k \$USD yearly fee)

C++ ISO standardization process (shortened version)

1. Get invited to a mailing list
2. Request a paper number from a registrar by e-mail
3. Write a paper with some proposal
4. Address the paper to a study group
5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
7. Find a champion to support your paper
8. Attend a committee meeting
 - as a guest, or
 - by becoming an ISO National Body representative, or
 - by joining on behalf of a company (1-10k \$USD yearly fee)

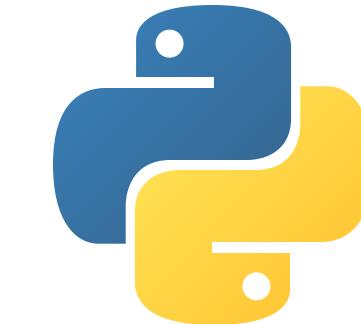
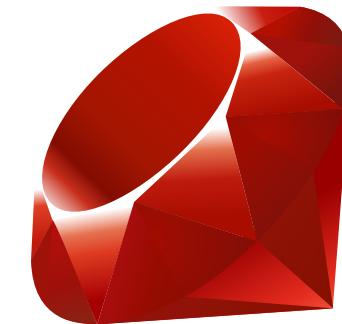
10. If there are concerns, resolve them and GOTO step 4
9. If the paper is decided to be worthy, it is moved along to a higher group
- C++ ISO standardization process (shortened version)
1. Get invited to a mailing list
 2. Request a paper number from a registrar by e-mail
 3. Write a paper with some proposal
 4. Address the paper to a study group
 5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
 6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
 7. Find a champion to support your paper
 8. Attend a committee meeting
 - as a guest, or
 - by becoming an ISO National Body representative, or
 - by joining on behalf of a company (1-10k \$USD yearly fee)

10. If there are concerns, resolve them and GOTO step 4
- C++ ISO standardization process (shortened version)
1. Get invited to a mailing list
 2. Request a paper number from a registrar by e-mail
 3. Write a paper with some proposal
 4. Address the paper to a study group
 5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
 6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
 11. Wait ~3 years for the next C++ standard
 7. Find a champion to support your paper
 8. Attend a committee meeting
 - as a guest, or
 - by becoming an ISO National Body representative, or
 - by joining on behalf of a company (1-10k \$USD yearly fee)
 9. If the paper is decided to be worthy, it is moved along to a higher group

10. If there are concerns, resolve them and GOTO step 4
9. If the paper is decided to be worthy, it is moved along to a higher group
- C++ ISO standardization process (shortened version)
1. Get invited to a mailing list
 2. Request a paper number from a registrar by e-mail
 3. Write a paper with some proposal
 4. Address the paper to a study group
 5. Submit the paper by e-mail before a deadline
 - 4-6 weeks before the next meeting, otherwise it will be discarded
 6. Ensure it is presented at a study group
 - by attending a committee meeting
 - or finding someone to present it
 7. Find a champion to support your paper
 8. Attend a committee meeting
 - as a guest, or
 - by becoming an ISO National Body representative, or
 - by joining on behalf of a company (1-10k \$USD yearly fee)
 11. Wait ~3 years for the next C++ standard
 12. Wait for implementation in a major compiler

BDFL

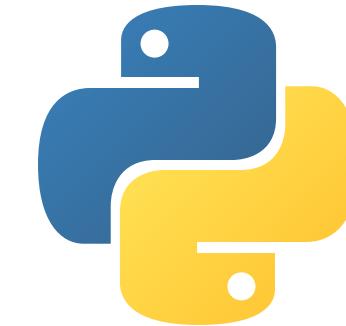
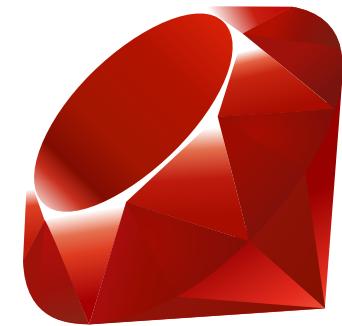
(Benevolent Dictator For Life)



(1995-2018)

BDFL

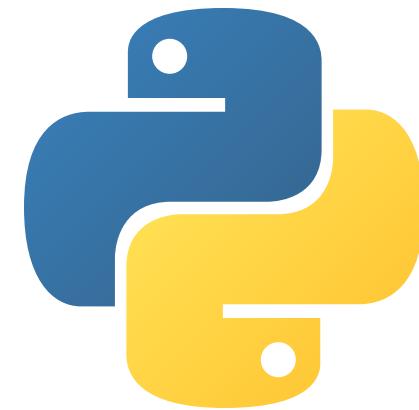
(Benevolent Dictator For Life)



(1995-2018)

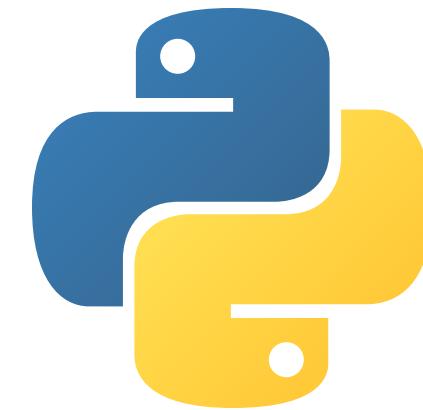
~~Bribe~~ Befriend the BDFL (?)

Open-source RFC process



(2018+)

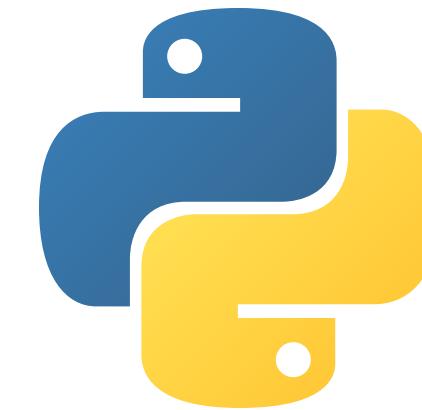
Open-source RFC process



(2018+)

Propose changes with RFCs

Open-source RFC process



(2018+)

Propose changes with RFCs
Gain voting rights by contributing

Evolution of Rust governance



2006

Started as a personal project
by Graydon Hoare (@Mozilla)



Project Servo

Technology from the past
come to save the future
from itself

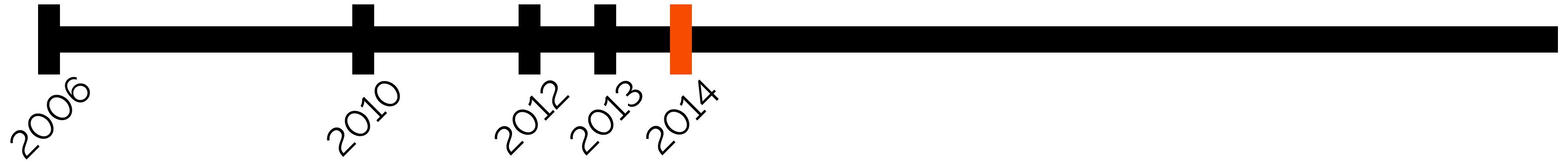
Mozilla Annual Summit, July 2010
[<graydon@mozilla.com>](mailto:graydon@mozilla.com)



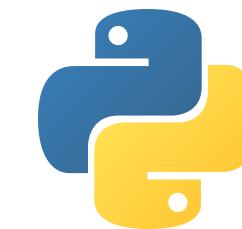
First "core team" formed (~5 people from Mozilla)

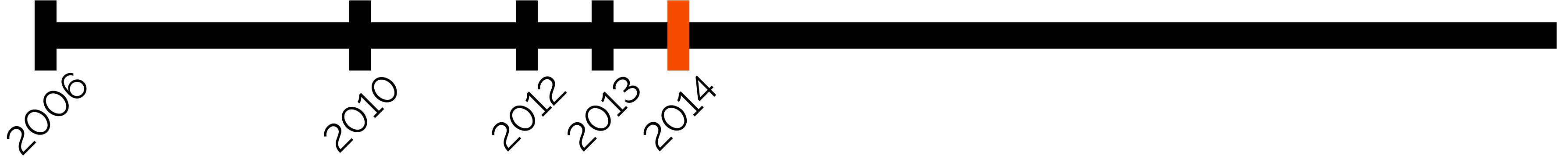


Graydon steps down as BDFL



Rust RFC process created (inspired by PEP)

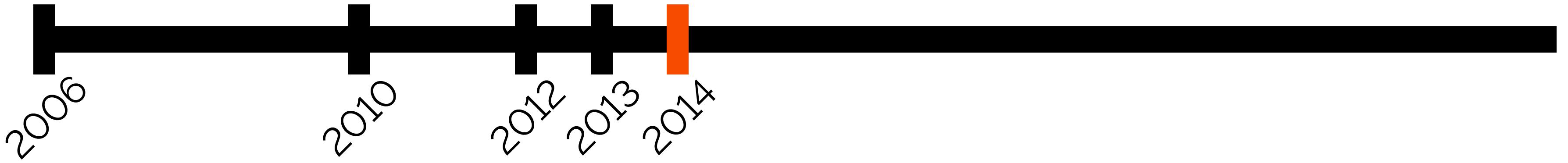




- Start Date: 2014-03-11
- RFC PR: [rust-lang/rfcs#1](#)
- Rust Issue: [rust-lang/rust#8122](#)

Summary

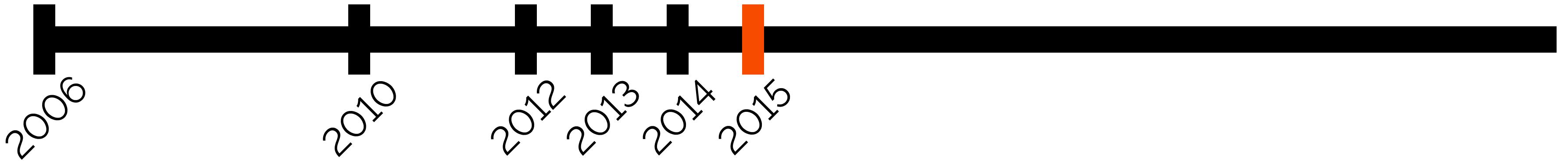
This is an RFC to make all struct fields private by default. This includes both tuple structs and structural structs.



- Start Date: 2014-03-11
- RFC PR: [rust-lang/rfcs#2](#), [rust-lang/rfcs#6](#)
- Rust Issue: N/A

Summary

The "RFC" (request for comments) process is intended to provide a consistent and controlled path for new features to enter the language and standard libraries, so that all stakeholders can be confident about the direction the language is evolving in.

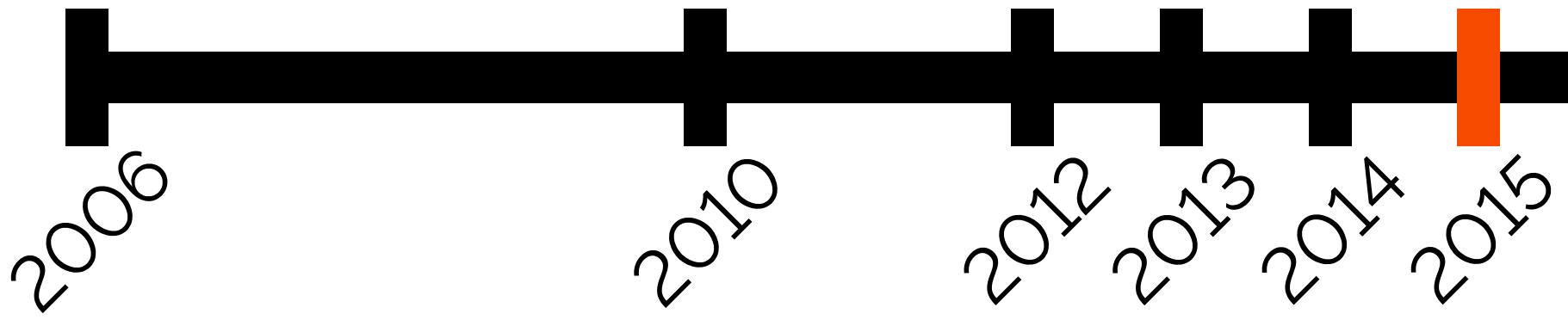


- Feature Name: not applicable
- Start Date: 2015-02-27
- RFC PR: [rust-lang/rfcs#1068](#)
- Rust Issue: N/A

Summary

This RFC proposes to expand, and make more explicit, Rust's governance structure. It seeks to supplement today's core team with several *subteams* that are more narrowly focused on specific areas of interest.

Thanks to Nick Cameron, Manish Goregaokar, Yehuda Katz, Niko Matsakis and Dave Herman for many suggestions and discussions along the way.



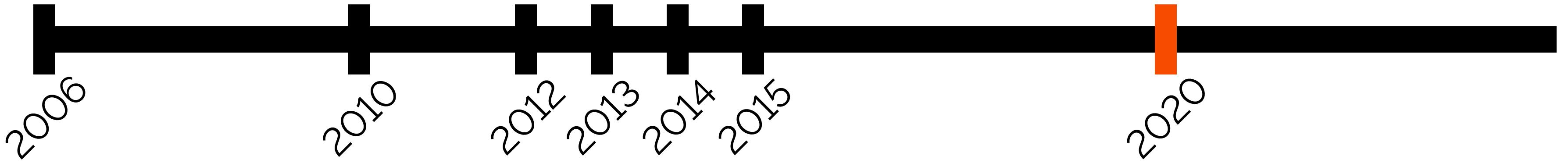
- Feature Name: not applicable
- Start Date: 2015-02-27
- RFC PR: [rust-lang/rfcs#1068](#)
- Rust Issue: N/A

- Language
- Library
- Compiler
- Infrastructure
- Moderation

Summary

This RFC proposes to expand, and make more explicit, Rust's governance structure. It seeks to supplement today's core team with several *subteams* that are more narrowly focused on specific areas of interest.

Thanks to Nick Cameron, Manish Goregaokar, Yehuda Katz, Niko Matsakis and Dave Herman for many suggestions and discussions along the way.



Mozilla lays off most employees paid to work on Rust

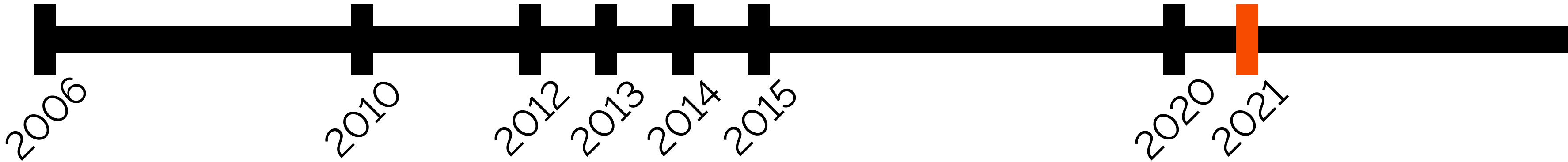
FROM THE TEAM

Changing World, Changing Mozilla

AUGUST 11, 2020

MITCHELL BAKER

This is a time of change for the internet and for Mozilla. From combatting a lethal virus and battling systemic racism to protecting individual privacy — one thing is clear: an open and accessible internet is essential to the fight.



Rust Foundation was established



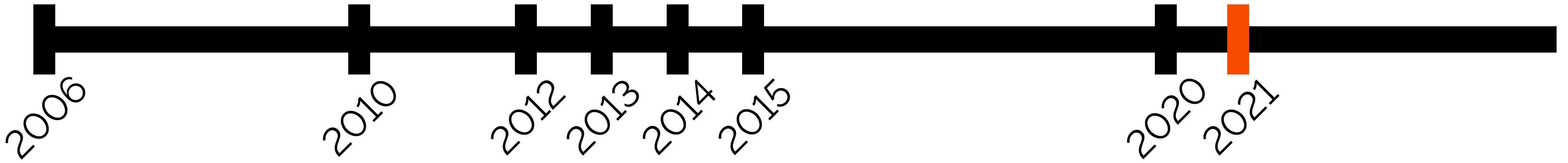
[AWS Open Source Blog](#)

Congratulations, Rustaceans, on the creation of the Rust Foundation!

by Shane Miller | on 08 FEB 2021 | in [Announcements](#), [Open Source](#) | [Permalink](#) | [Comments](#) |

[Share](#)

We celebrate today's [announcement](#) of the creation of the Rust Foundation with the rest of the Rust community. As a founding member of the Rust Foundation, AWS is dedicated to the Rust Foundation's mission to empower Rust maintainers to joyfully do their best work.



mod team resignation #671

Merged

aidanhs merged 1 commit into `rust-lang:master` from `BurntSushi:ag/mod-t`

Conversation 10

Commits 1

Checks 0

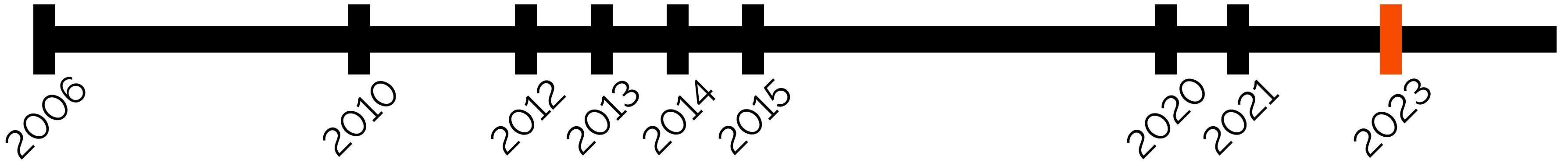
Files changed 1



BurntSushi commented on Nov 22, 2021 • edited

Memb

The entire moderation team resigns, effective immediately. This resignation is done in protest of the Core Team placing themselves unaccountable to anyone but themselves.



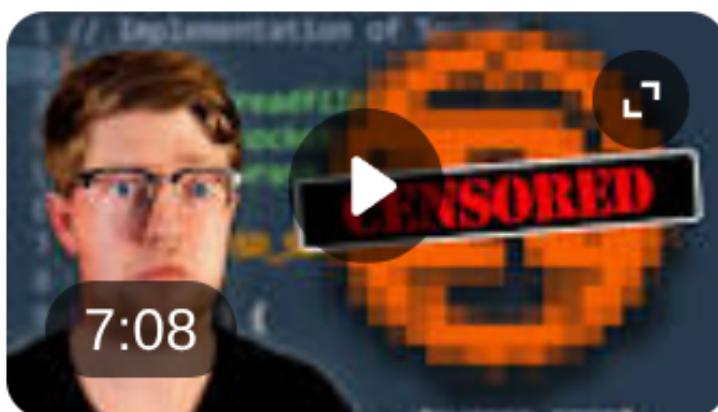
Rust Foundation Has A Serious Trademark Problem

YouTube · Brodie Robertson ·
13 Apr 2023



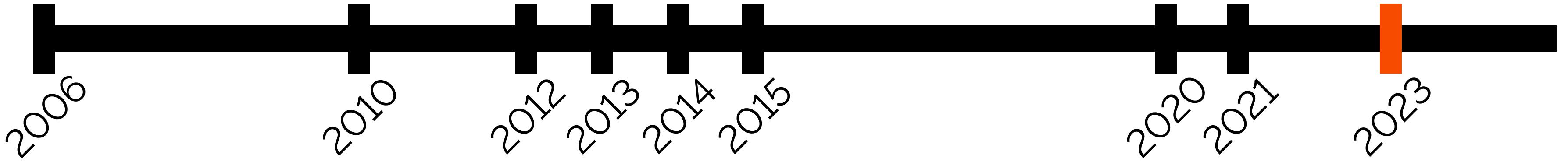
Rust Foundation IS DOING WHAT????

YouTube · ThePrimeagen ·
11 Apr 2023



The Ru\$t Foundation Made a HUGE Mistake.

YouTube · Low Level ·
18 Apr 2023

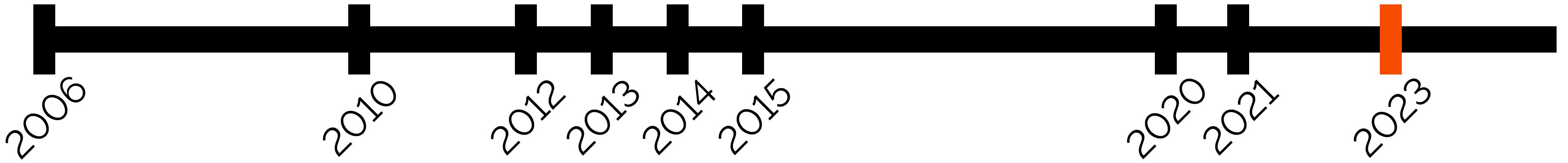


CrabLang

[Home](#) [Discord](#) [Twitter](#) [GitHub](#)

The Official Home of CrabLang!

A community fork of a language named after a plant fungus. All of the memory-safe features you love, now with 100% less bureaucracy!



- Feature Name: leadership-council
- Start Date: 2022-08-01
- RFC PR: [rust-lang/rfcs#3392](#)
- Rust Issue: N/A

Summary

This RFC establishes a Leadership Council as the successor of the core team¹ and the new governance structure through which Rust Project members collectively confer the authority² to ensure successful operation of the Project. The Leadership Council delegates much of this authority to teams (which includes subteams, working groups, etc.³) who autonomously make decisions concerning their purviews. However, the Council retains some decision-making authority, outlined and delimited by this RFC.



Compiler team

Developing and managing compiler
internals and optimizations

Dev tools team

Contributing to and creating the Rust
development tools

[MEMBERS & CONTACTS](#)

[MEMBERS & CONTACTS](#)

Infrastructure team

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

[MEMBERS & CONTACTS](#)

Language team

Designing and helping to implement new
language features

[MEMBERS & CONTACTS](#)

Launching pad

An interim home for teams

[MEMBERS & CONTACTS](#)

Library team

Managing and maintaining the Rust
standard library and official rust-lang
crates

[MEMBERS & CONTACTS](#)

Moderation team

Helping uphold the code of conduct and
community standards and mediating
conflicts between team members

[MEMBERS & CONTACTS](#)

Leadership council

Charged with the success of the Rust Project as whole, consisting of representatives from top-level teams

[MEMBERS & CONTACTS](#)

Compiler team

Developing and managing compiler internals and optimizations

Dev tools team

Contributing to and creating the Rust development tools

Infrastructure team

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

[MEMBERS & CONTACTS](#)

Language team

Designing and helping to implement new language features

Launching pad

An interim home for teams

[MEMBERS & CONTACTS](#)

[MEMBERS & CONTACTS](#)

[MEMBERS & CONTACTS](#)

Library team

Managing and maintaining the Rust standard library and official rust-lang crates

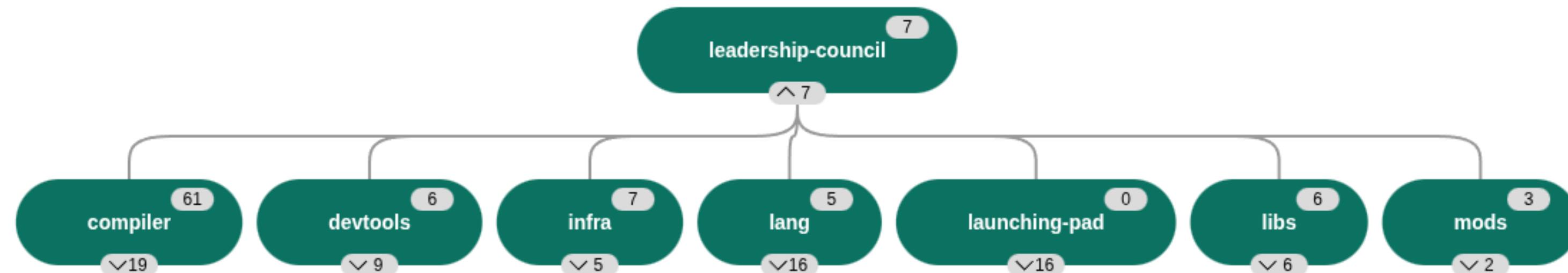
[MEMBERS & CONTACTS](#)

Moderation team

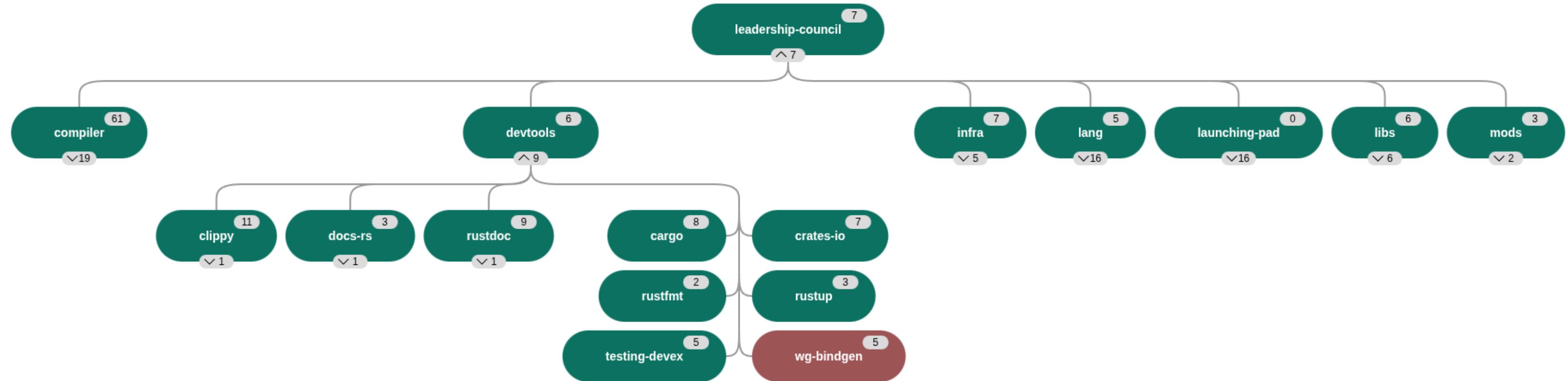
Helping uphold the code of conduct and community standards and mediating conflicts between team members

[MEMBERS & CONTACTS](#)

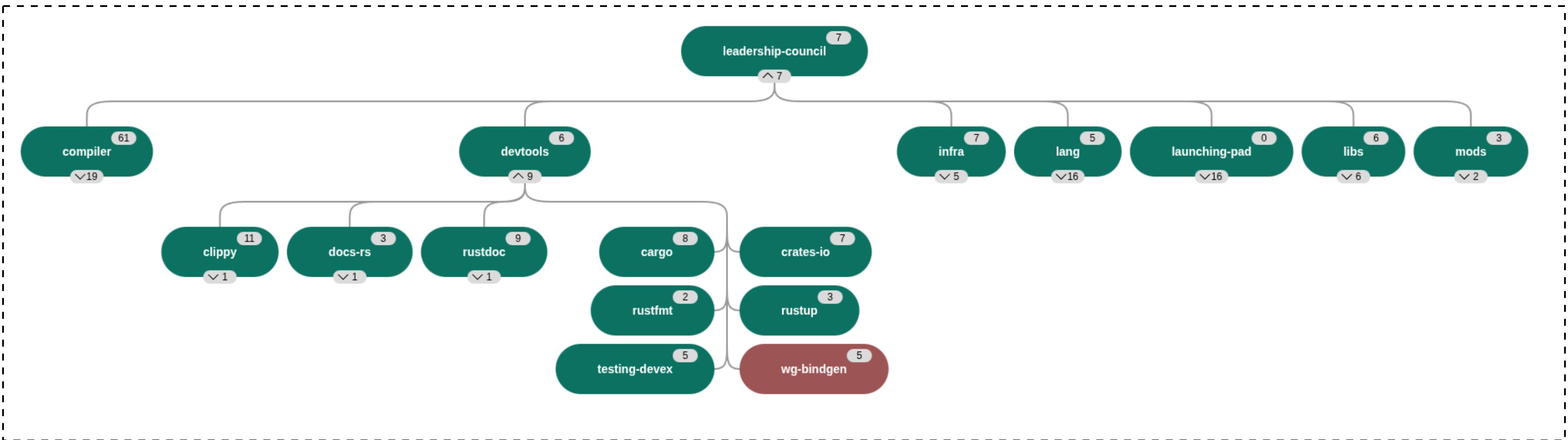




Source: Rust Org Chart by Eric Huss



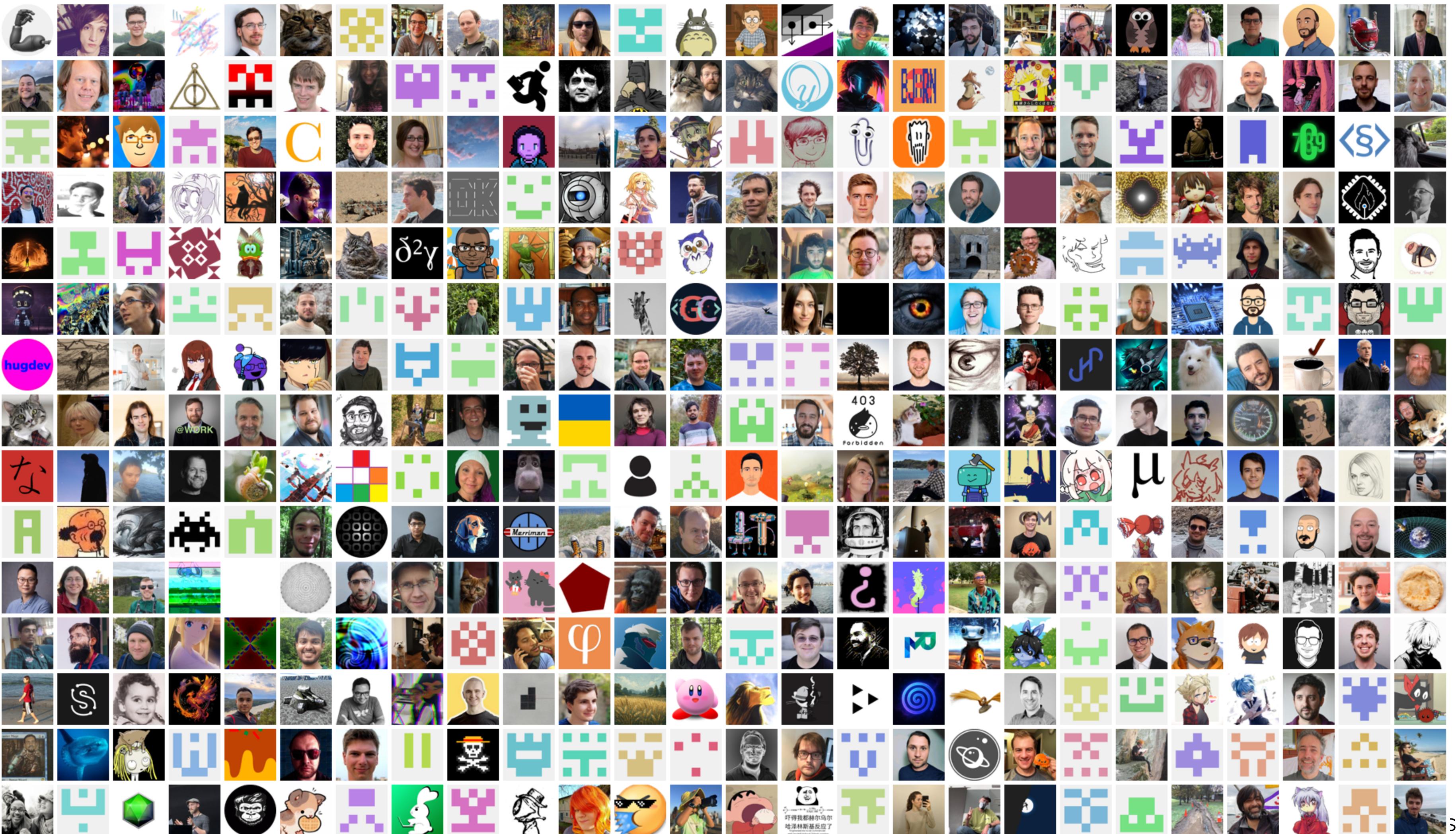
Source: Rust Org Chart by Eric Huss

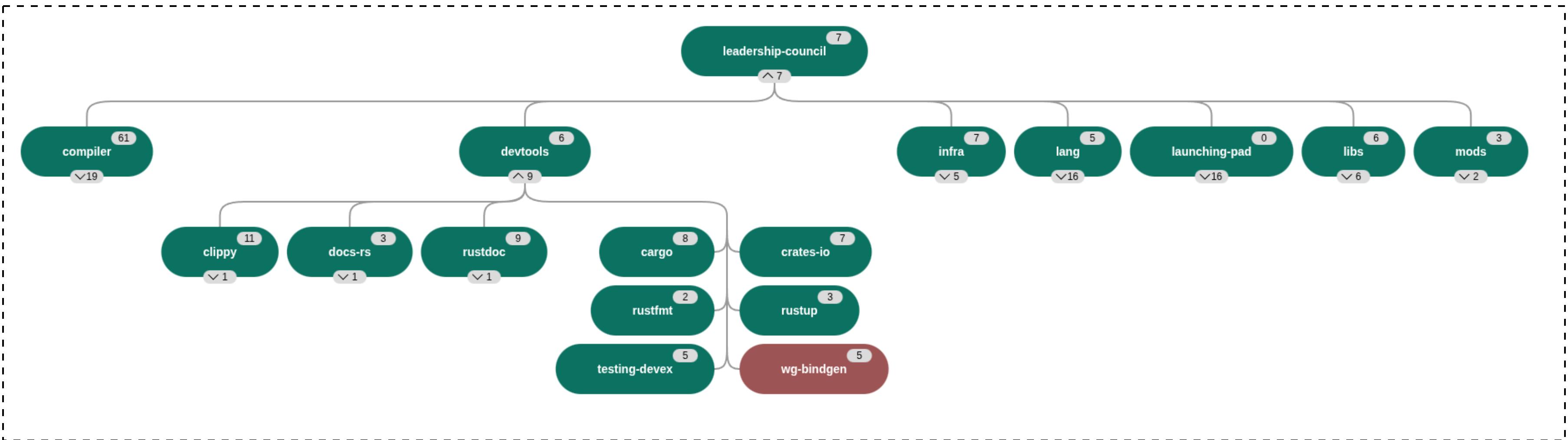


Rust Project

~300 people, ~100 teams

Source: Rust Org Chart by Eric Huss

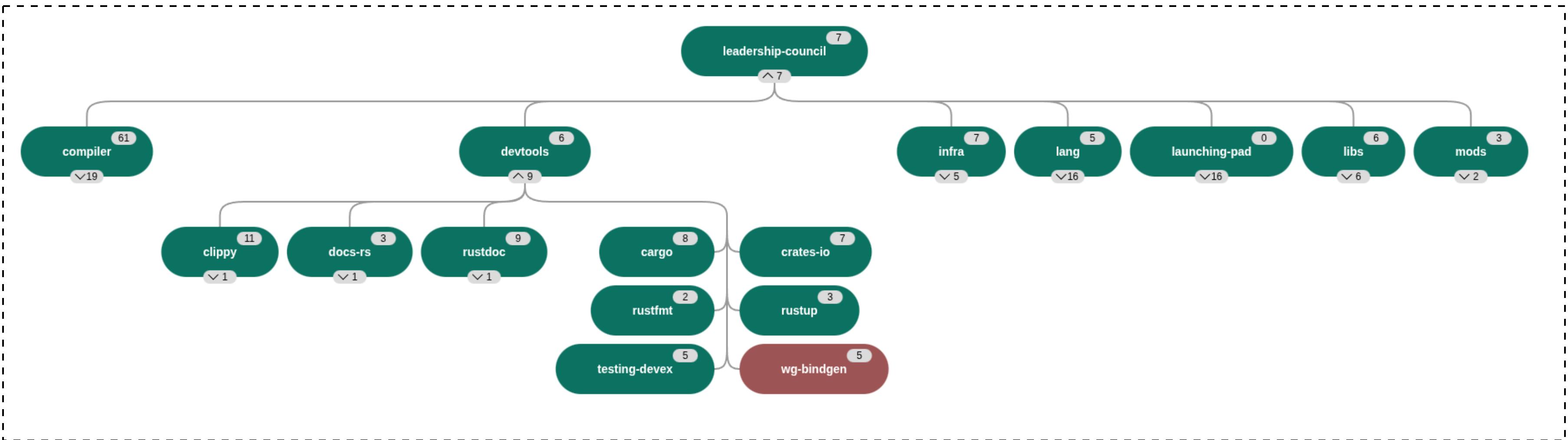




Rust Project

~300 people, ~100 teams

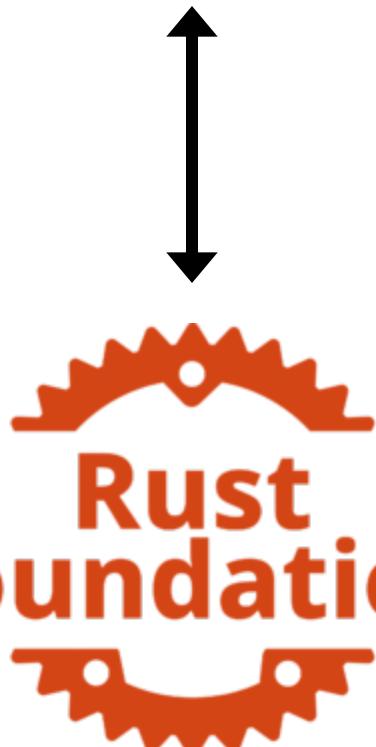
Development, maintenance, governance, ...



Rust Project

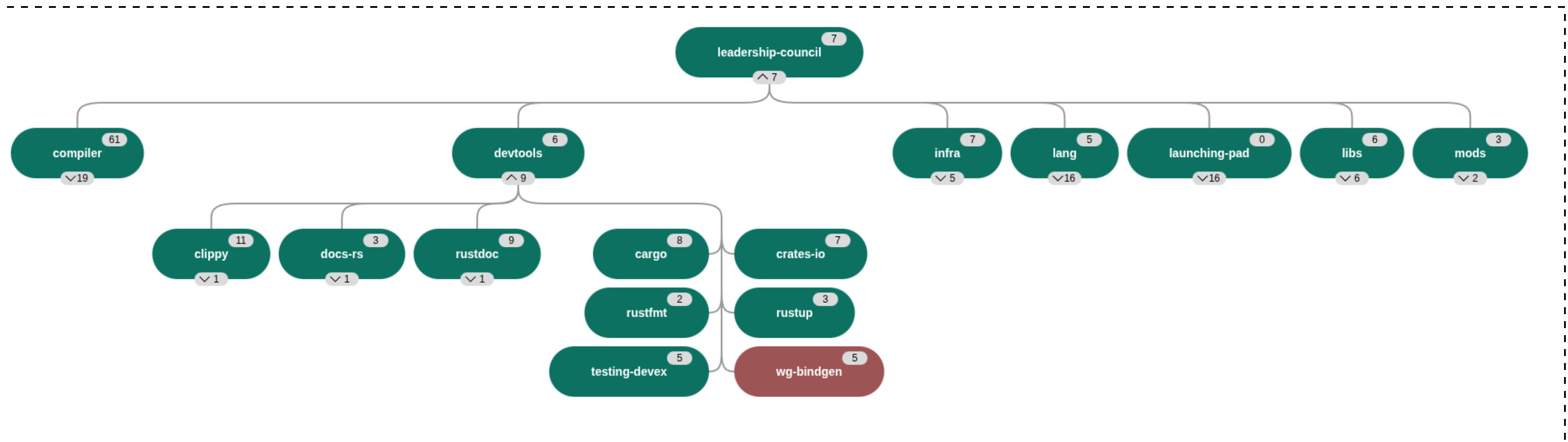
~300 people, ~100 teams

Development, maintenance, governance, ...



Legal, infra, domains, trademark, funding, ...

Source: Rust Org Chart by Eric Huss



Rust Project

~300 people, ~100 teams

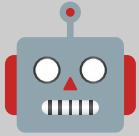
Development, maintenance, governance, ...



Legal, infra, domains, trademark, funding, ...

(Hundreds of) contributors

Source: Rust Org Chart by Eric Huss

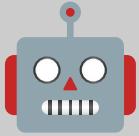


sync-team



https://github.com/rust-lang/team

arm.toml	Split Arm team to distinguish maintainers.	3 months ago
bootstrap.toml	make onur-ozkan co-lead of bootstrap team	5 months ago
cargo.toml	Add devtools alumni (#1137)	last month
clippy.toml	Backfill all currently listed alumni's teams (#1134)	last month
cloud-compute.toml	Give dev-desktop permission to t-libs	2 months ago



<https://github.com/rust-lang/team>

```
1  name = "infra"
2  top-level = true
3
4  [people]
5  leads = ["shepmaster", "jdno"]
6  members = ["Mark-Simulacrum", "pietroalbini", "shepmaster", "jdno", "Kobzol", "marcoieni", "jieyouxu"]
7  alumni = ["alexcrichton", "aturon", "brson", "sgrif", "nellshamrell", "ashleygwilliams", "aidanhs",
8    "jyn514", "TimNN", "erickt", "frewsxcv", "tomprince", "carols10cents", "rylev", "kennytm"]
9
10 [[github]]
11 orgs = ["rust-lang", "rust-lang-nursery"]
12
13 [permissions]
14 perf = true
15 crater = true
16 dev-desktop = true
17 bors.rust.review = true
```

Zulip



- ▶ DIRECT MESSAGES
- ▼ OTHER
 - 🌐 rust-for-linux
 - 🌐 rustc-codegen-gcc
 - 🌐 t-cargo
 - 🌐 t-cargo/build-integration
 - 🌐 t-cargo/PubGrub
 - 🌐 t-community/rust-survey
 - 🌐 t-compiler
 - Next steps for contracts
 - Mitigation enforcement (`-C allow-partial-mitigations`)
 - valgrind slice::Iterator::position complaint
 - `box_new` as a macro?
 - error code tests
 - Grouping target modifier options?
 - Add `supports_xray` to `aarch64_unknown_none.rs`
 - normalizing DefIds for matching expected MIR in tests
 - SHOW ALL TOPICS



RalfJ EDITED

to call that we'd have to make it a lang item

21:21

this is MIR being generated by the `write_box_via_move` intrinsic

21:21

we can't convert the Box to a raw ptr before invoking the intrinsic because then the box would not get freed if evaluating `$x` panics

21:24



deltragon

Just for completeness sake, I assume that `super let` also has the same issues of not preserving lifetimes/type inference correctly?

22:36

OCT 29



RalfJ

I can only assume so; I don't know how one would user super let here

08:51

- 🌐 t-compiler/call-for-participation
- 🔒 t-compiler/contrib-private
- 🌐 t-compiler/debuginfo



Async chat meetings



t-release > meeting 2025 09 19 ✓ ⏱ ⋮ SEP 19

 **simulacrum**
@T-release meeting!
 Josh Stone (cuviper), Pietro Albini 18:32

release retro + task assignment today 18:32

re:release -- not sure there's much to say, I think the release went pretty smoothly 18:33
 Josh Stone (cuviper), Pietro Albini

any further thoughts on release? 18:33

@triagebot 📺 end-topic

 **triagebot** 📺
Does anyone have something to add on the current topic?
React with 🛠 if you have something to say.
React with ✅ if not.
 1  3 18:33

 **simulacrum**
previous -- 1.90:

- Blog + release - Mark
- Release notes - Boxy
- Crater - Josh

 1.91: 18:35



Async chat meetings

t-lang/meetings > Design meeting 2025-08-27: Ergonomic RC ✓ ⏱ ⋮ AUG 27

TC 07:54
The design meeting for today will be ergonomic reference counting.
<https://github.com/rust-lang/lang-team/issues/343>

Document/minutes: https://hackmd.io/L7RV27tgTqm7Xem_xddmBQ 08:13
(It's a long one; you may want to start reading early.)

TC 18:30
@T-lang : It's time to design.

@triagebot 📖 read 18:30

triagebot 📖 18:30
Click on the 📖 when you start reading (and leave it clicked).
Click on the 🏁 when you finish reading.

7 7



How to change something in Rust?



How to change something in Rust?

- "Simple" change => send a pull request



How to change something in Rust?

- "Simple" change => send a pull request
- Complex change => write an RFC



Rust RFC process

(RFC = Request for Comments)



Rust RFC process

(RFC = Request for Comments)

“

The “RFC” (request for comments) process is intended to provide a consistent and controlled path for new features to enter the language and standard libraries, so that all stakeholders can be confident about the direction the language is evolving in.

Rust RFC#2 ,”



Rust RFC process

(RFC = Request for Comments)

“

The “RFC” (request for comments) process is intended to provide a consistent and controlled path for new features to enter the language and standard libraries, so that all stakeholders can be confident about the direction the language is evolving in.

Rust RFC#2 ,”

- Technical (e.g. new language features)



Rust RFC process

(RFC = Request for Comments)

“

The “RFC” (request for comments) process is intended to provide a consistent and controlled path for new features to enter the language and standard libraries, so that all stakeholders can be confident about the direction the language is evolving in.

Rust RFC#2 ,”

- Technical (e.g. new language features)
- Non-technical (e.g. changes to governance)



Rust RFC process

(RFC = Request for Comments)

“

The “RFC” (request for comments) process is intended to provide a consistent and controlled path for new features to enter the language and standard libraries, so that all stakeholders can be confident about the direction the language is evolving in.

Rust RFC#2 ,”

- Technical (e.g. new language features)
- Non-technical (e.g. changes to governance)
- "One-way door" decisions

Step 0: Vibe-check

 **Pre-RFC: `#[derive(Default)]` on enums with a `#[default]` attribute**

 language design

 **jhpratt** 2  Apr '21

I've extracted and slightly modified some text from [Centril's draft RFC 12](#) from a while back that proposed this and much more. He has granted permission for me to use the text.

- Feature Name: `derive_enum_default`
- Start Date: 2021-04-07
- RFC PR: TODO
- Rust Issue: TODO



Step 1: Write the RFC

Step 1: Write the RFC

1. Motivation

Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation
4. Drawbacks



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation
4. Drawbacks
5. Rationale and alternatives



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation
4. Drawbacks
5. Rationale and alternatives
6. Prior art



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation
4. Drawbacks
5. Rationale and alternatives
6. Prior art
7. Unresolved questions



Step 1: Write the RFC

1. Motivation
2. Guide-level explanation (how to teach this?)
3. Reference-level explanation
4. Drawbacks
5. Rationale and alternatives
6. Prior art
7. Unresolved questions
8. Future possibilities



Step 2: Propose the RFC with a PR

[derive(Default)] on enums with a #[default] attribute #3

Merged pnkfelix merged 1 commit into rust-lang:master from jhpratt:derive-enum-default on Jul 27, 2021

Conversation 73 Commits 1 Checks 0 Files changed 1

jhpratt commented on Apr 12, 2021 • edited

Member ...

Pre-RFC on IRLO

TLDR allow this:

```
[#] [derive(Default)]
enum Option<T> {
    #[default]
    None,
    Some(T),
}
```

Rendered

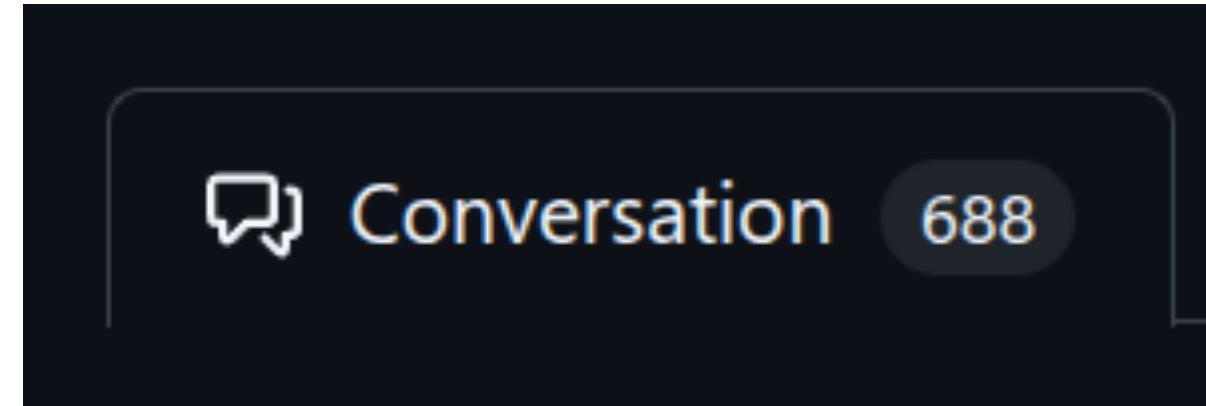
48 9 13

The screenshot shows a GitHub pull request page for a merged commit. The title of the PR is "# [derive(Default)] on enums with a #[default] attribute #3". A purple button indicates it is "Merged" by pnkfelix on Jul 27, 2021. Below the title, there are navigation links for Conversation (73), Commits (1), Checks (0), and Files changed (1). A comment from user jhpratt is displayed, dated Apr 12, 2021, with an edit history. The comment text is: "Pre-RFC on IRLO" and "TLDR allow this:" followed by a code snippet. The code snippet shows an enum Option<T> with a #[default] attribute for the None variant. Below the code, the word "Rendered" is shown, followed by reaction counts: 48, 9, and 13.

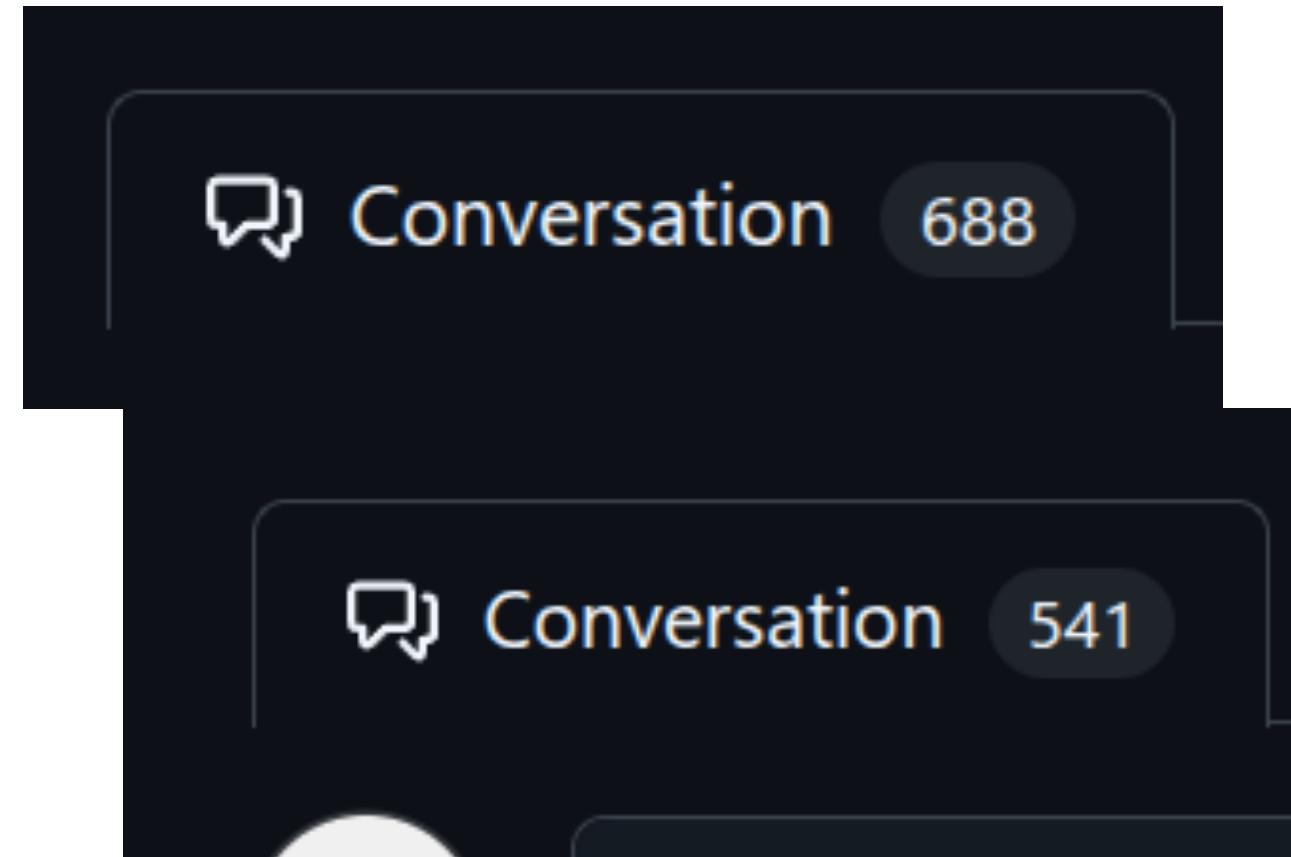
Step 3: Receive (a LOT of) comments



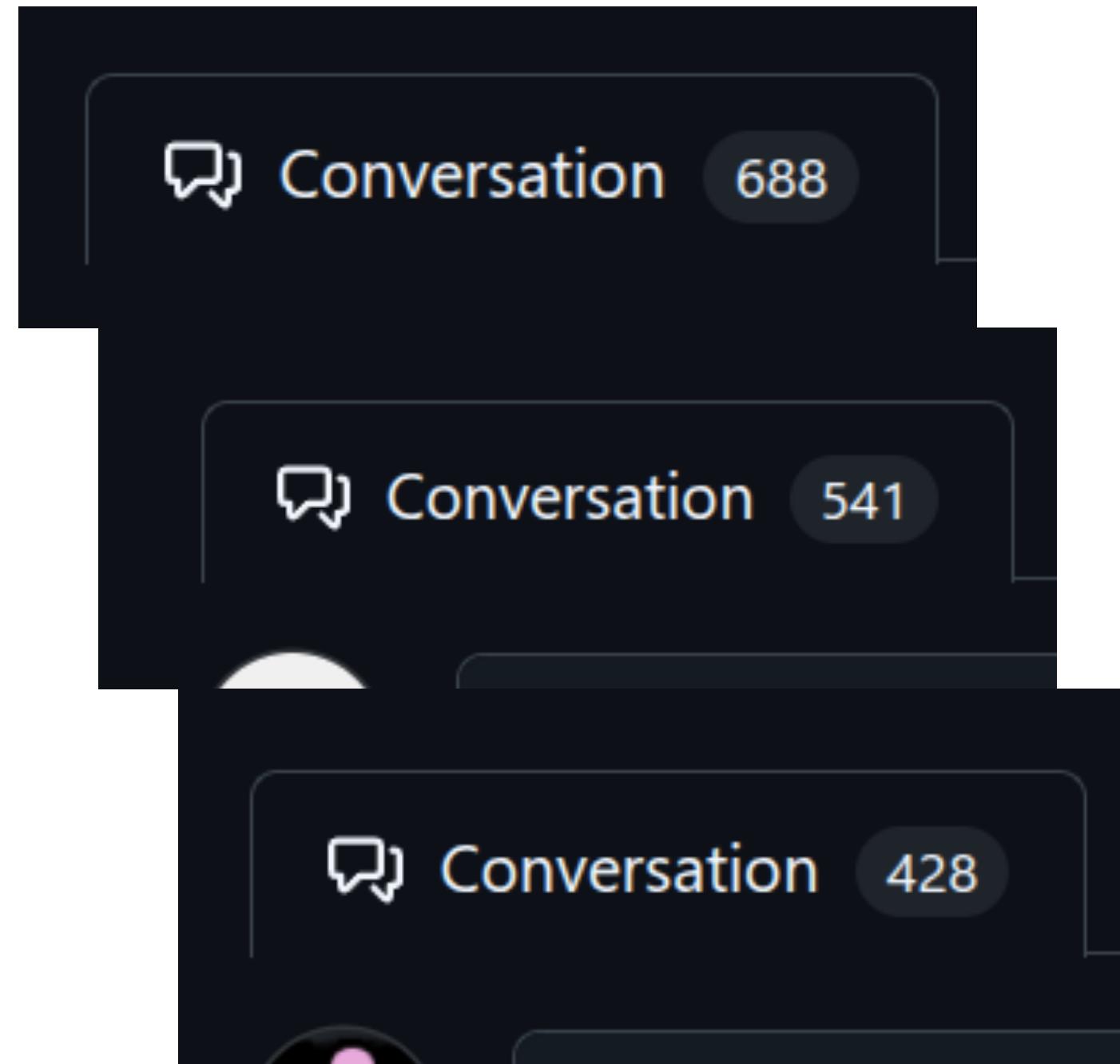
Step 3: Receive (a LOT of) comments



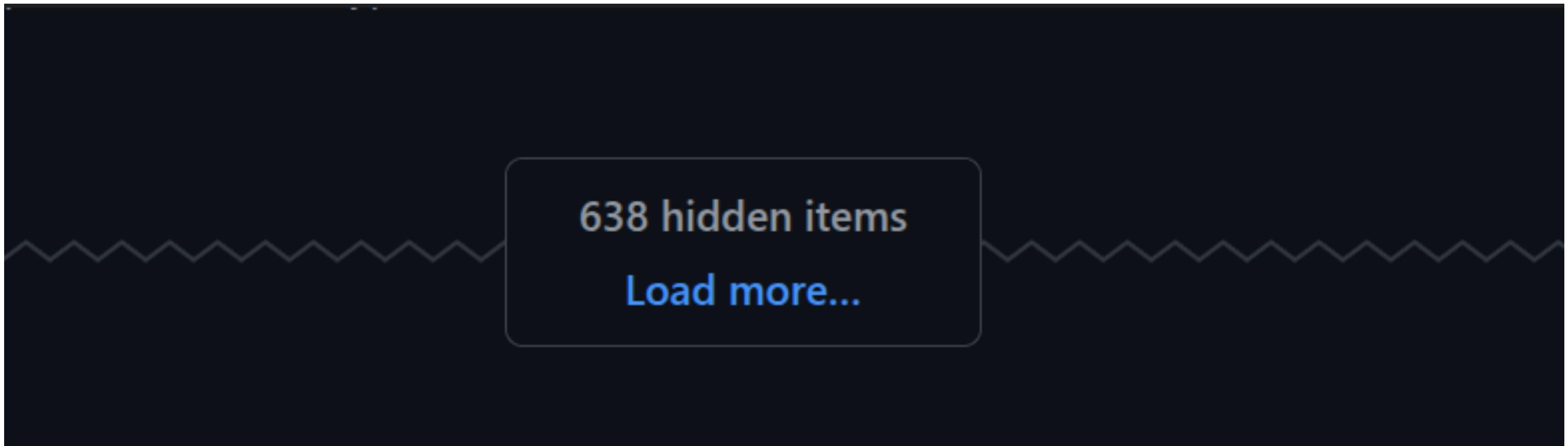
Step 3: Receive (a LOT of) comments



Step 3: Receive (a LOT of) comments



Step 3: Receive (a LOT of) comments





Step 3: Receive (a LOT of) comments

Lokathor commented on Apr 15, 2021

please don't use "iff", that's a very jargon-y variant of the word that honestly looks like a typo

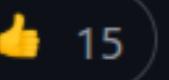
18

Step 3: Receive (a LOT of) comments

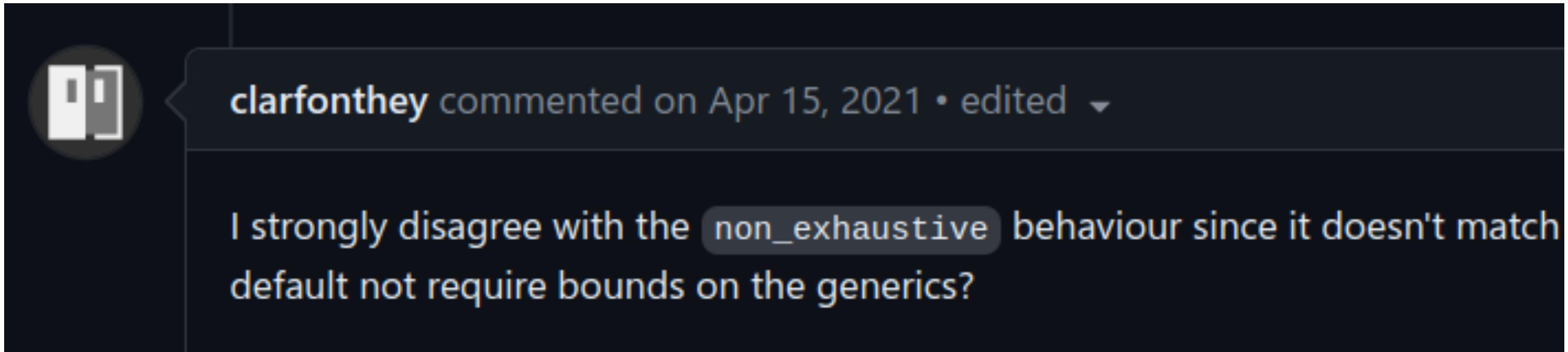


 **steveklabnik** commented on Mar 1, 2016 Contributor •

Bikeshed checking in: I strongly prefer a keyword like `impl Trait` over `@Trait`, as it's only barely longer, and is significantly more discoverable.

  15

Step 3: Receive (a LOT of) comments



A screenshot of a GitHub comment card. The profile picture of the user 'clarfonthey' is on the left, showing a white icon on a dark background. The user's name 'clarfonthey' is in blue, followed by 'commented on Apr 15, 2021 • edited'. The comment text is: 'I strongly disagree with the non_exhaustive behaviour since it doesn't match default not require bounds on the generics?'

clarfonthey commented on Apr 15, 2021 • edited

I strongly disagree with the non_exhaustive behaviour since it doesn't match default not require bounds on the generics?

Step 4: Integrate feedback & repeat

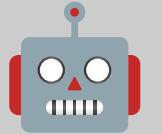
The screenshot shows a GitHub pull request interface with two comments from the user 'Kobzol'. The first comment was posted on July 9, 2023, and the second on July 19, 2023. Both comments include a smiley face emoji.

Kobzol commented on Jul 9

Moved the discussion of the `From` direction to `Rationale and alternatives`, and briefly mentioned `#[derive(Into)]` as a future possibility.

Kobzol commented on Jul 19

@BurntSushi Hey, could you please clarify if my changes have resolved your concerns, or if not, what would you like to see here? Thanks!



rfcbot

Step 5: Vote

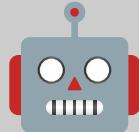
scottmcm commented on Sep 11, 2024

I've wanted something like this for ages. I took a fresh vestigial section, it looks good to go to me! Let's see

@rfcbot fcp merge

5

A screenshot of a GitHub pull request comment. The comment is from a user named scottmcm, posted on September 11, 2024. The text of the comment reads: "I've wanted something like this for ages. I took a fresh vestigial section, it looks good to go to me! Let's see". Below the text, there is a mention of "@rfcbot fcp merge". At the bottom of the comment, there are two icons: a smiley face and a rocket ship with the number "5" next to it, indicating five upvotes or approvals.



rfcbot

Step 5: Vote

 rfcbot commented on Sep 11, 2024 • edited by traviscross ...

Team member [@scottmcm](#) has proposed to merge this. The next step is review by the rest of the tagged team members:

- [@joshtriplett](#)
- [@nikomatsakis](#)
- [@pnkfelix](#)
- [@scottmcm](#)
- [@tmandry](#)
- [@traviscross](#)

No concerns currently listed.

Once a majority of reviewers approve (and at most 2 approvals are outstanding), this will enter its final comment period. If you spot a major issue that hasn't been raised at any point in this process, please speak up!

cc [@rust-lang/lang-advisors](#): FCP proposed for lang, please feel free to register concerns.
See [this document](#) for info about what commands tagged team members can give me.

  3



Consensus

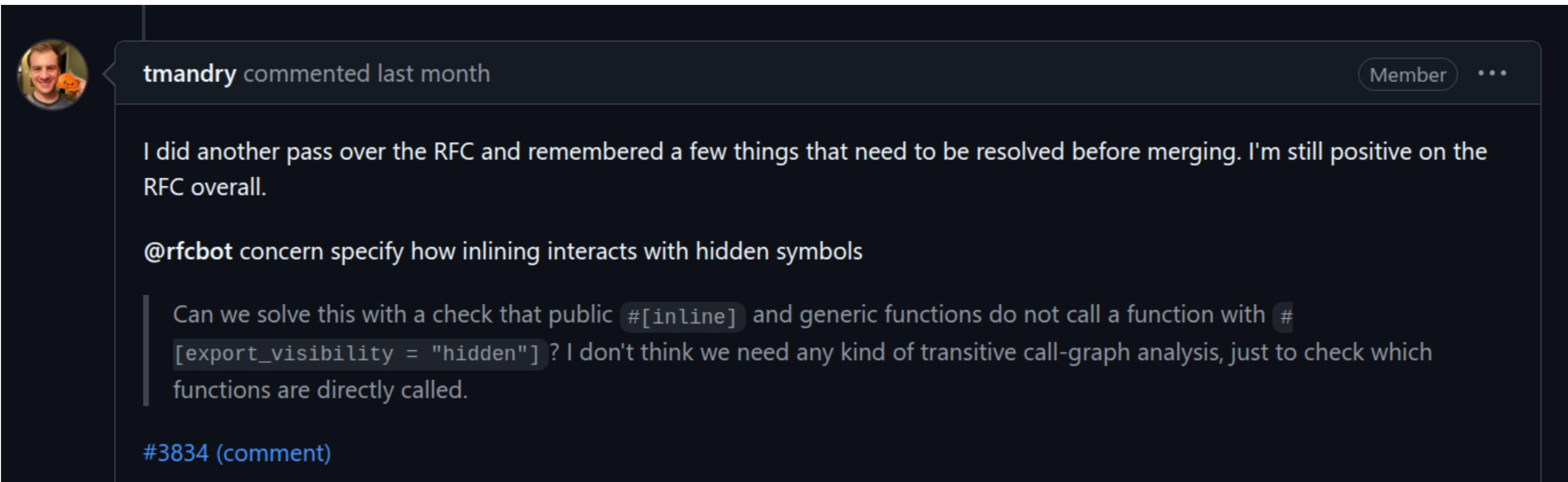


Consensus

- At most two votes missing

Consensus

- At most two votes missing
- No concerns



 **tmandry** commented last month

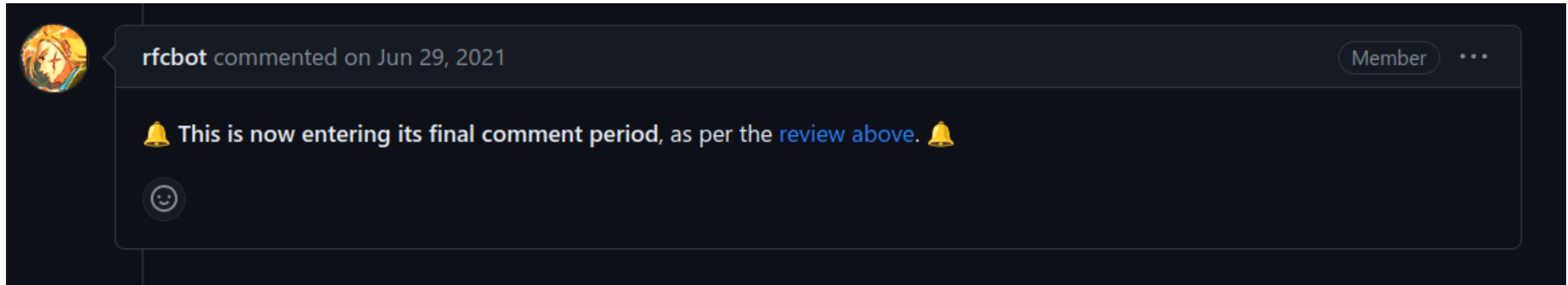
I did another pass over the RFC and remembered a few things that need to be resolved before merging. I'm still positive on the RFC overall.

@rfcbot concern specify how inlining interacts with hidden symbols

Can we solve this with a check that public `#[inline]` and generic functions do not call a function with `#[
export_visibility = "hidden"]`? I don't think we need any kind of transitive call-graph analysis, just to check which functions are directly called.

[#3834 \(comment\)](#)

Step 6: FCP (Final Comment Period)



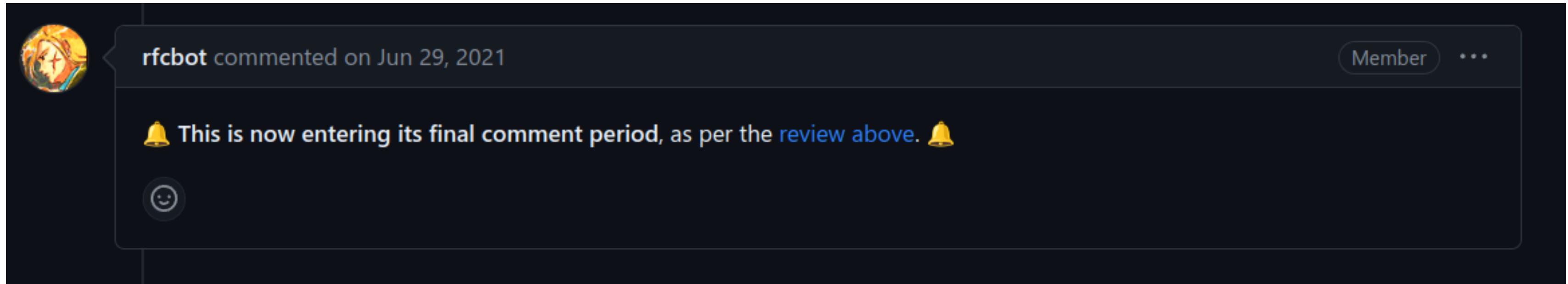
A screenshot of a GitHub comment card. The profile picture of the user 'rfcbot' is on the left, followed by the text 'rfcbot commented on Jun 29, 2021'. To the right of the date are two status indicators: 'Member' with a gear icon and three vertical dots. The main message in the body of the comment is: '🔔 This is now entering its final comment period, as per the review above. 🔔'. Below the message is a small circular emoji icon containing a smiley face.

rfcbot commented on Jun 29, 2021

Member ...

🔔 This is now entering its final comment period, as per the review above. 🔔

Step 6: FCP (Final Comment Period)



- Lasts for 10 days



Step 6: FCP (Final Comment Period)

rfcbot commented on Jun 29, 2021

Member ...

🔔 This is now entering its final comment period, as per the review above. 🔔

😊

- Lasts for 10 days
- Last chance for someone to object



Step 6: FCP (Final Comment Period)

rfcbot commented on Jun 29, 2021

Member ...

🔔 This is now entering its final comment period, as per the review above. 🔔

😊

- Lasts for 10 days
- Last chance for someone to object
- Announced in various communication channels

Step 7: RFC done



 rfcbot commented on Jul 9, 2021 Member ...

The final comment period, with a disposition to **merge**, as per the [review above](#), is now **complete**.

As the automated representative of the governance process, I would like to thank the author for their work and everyone else who contributed.

The RFC will be merged soon.





Step 7: RFC done



 **rfcbot** commented on Jul 9, 2021 Member ...

The final comment period, with a disposition to **merge**, as per the [review above](#), is now **complete**.

As the automated representative of the governance process, I would like to thank the author for their work and everyone else who contributed.

The RFC will be merged soon.



(this can take months or even years)



Step 7: RFC done



rfcbot commented on Jul 9, 2021

The final comment period, with a disposition to **merge**, as per the [review above](#), is now **complete**.

As the automated representative of the governance process, I would like to thank the author for their work and everyone else who contributed.

The RFC will be merged soon.

(this can take months or even years)

Next step: (find someone to) implement it



Sometimes, making decisions is *hard*



Great int debate (2014)



Great int debate (2014)

“

We have been reading these threads and have also done a lot of internal experimentation, and we believe we've come to a final decision on the fate of integers in Rust.

Core team (2014) ,”



"No new rationale" rule

“

Decisions must be made only on the basis of rationale already debated in public (to a steady state).

”



Awaiting a solution (2018/2019)



Awaiting a solution (2018/2019)

```
await!(fut);  
await fut;  
await { fut };  
fut.await;  
fut.await();  
fut.await!;  
fut@await;
```



Awaiting a solution (2018/2019)

```
await!(fut);  
await fut; ←—————  
await { fut };  
fut.await;  
fut.await();  
fut.await!;  
fut@await;
```

Rust community wanted this



Awaiting a solution (2018/2019)

```
await!(fut);  
await fut; ← blue arrow  
await { fut };  
fut.await; ← red arrow  
fut.await();  
fut.await!;  
fut@await;
```

Rust community wanted this

Lang team wanted this



arthureroberer

Apr 2019

the Orthogonality argument convinced me
im now on the postfix camp

7 ❤️ @ ...



drwwlkr

Apr 2019

The orthogonality argument feels really unconvincing since it requires breaking what people have already learned about field access syntax.

Having magical implicit fields feels strictly worse than having to type extra parens.

2 Replies ▾

23 ❤️ @ ...





Darksonn · 4 yr. ago
tokio · rust-for-linux

I also remember someone making a survey asking people which syntax they wanted, and the Rust team decided to not follow the majority of that survey. I am happy they took the decision to do that.



203



Reply

Share

...



othermike · 4 yr. ago

Yeah, I'm not sure surveys of the general userbase are that useful in language design. I can remember plenty of examples of younger-me voting for a faster horse.



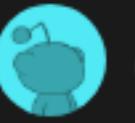
138



Reply

Share

...



crabbytag · 4 yr. ago

a faster horse

I googled this. It's based on a quote attributed to Henry Ford

If I had asked people what they wanted, they would have said faster horses.



114



Reply

Share

...



sepease · 4 yr. ago

I objected.

I was wrong.

▲ 304 ▾ Reply Share ...



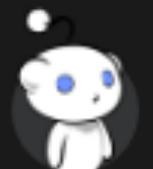
disintegore · 4 yr. ago

I fought the crab and the crab won

▲ 172 ▾ Reply Share ...



Narishma · 4 yr. ago



dpc_pw · 4 yr. ago

The crab always wins.

▲ 12 ▾ Reply Share ...



Open-source (governance) is hard...



Open-source (governance) is hard...

- Team-based decision-making



Open-source (governance) is hard...

- Team-based decision-making
- Bottom-up development
- Rust is NOT a company!



Open-source (governance) is hard...

- Team-based decision-making
- Bottom-up development
 - Rust is NOT a company!
- With several hundreds of people
 - Most of them unpaid volunteers



Open-source (governance) is hard...

- Team-based decision-making
- Bottom-up development
 - Rust is NOT a company!
- With several hundreds of people
 - Most of them unpaid volunteers
- Burnout, personal conflicts, lack of transparency, ...



Open-source (governance) is hard...

- Team-based decision-making
- Bottom-up development
 - Rust is NOT a company!
- With several hundreds of people
 - Most of them unpaid volunteers
- Burnout, personal conflicts, lack of transparency, ...

...but also fun & impactful!



Development

Sending a PR



Cache derive proc macro expansion with incremental query #145354

 Open

Kobzol wants to merge 6 commits into [rust-lang:master](#) from [Kobzol:cache-derive-macros](#) 

 Conversation 38

 Commits 6

 Checks 10

 Files changed 17



Kobzol commented on Aug 13 • edited 

Member 

This is a revival of [#129102](#), originally implemented by @futile. Since it looks like they are not active currently, I'd like to push this work forward.

The first commit is squashed and rebased work from the original PR, with author attribution to futile. The rest of the commits are some additional comments that I created mostly for myself to understand what happens here. I also did some cleanups based on Vadim's review comments on the original PR, plus I refactored the TLS access a bit using [scoped_tls](#).

Reviewers

 cjgillot

 petrochenkov

 bjorn3

Still in progress? [Cor](#)



Sending a PR



A screenshot of a GitHub pull request comment. The comment is from the user "rustbot" and was posted "last week". The comment text reads: "r? @Mark-Simulacrum" followed by "rustbot has assigned @Mark-Simulacrum. They will have a look at your PR within the next two weeks and either review your PR or reassign to another reviewer." Below the text, there is a note: "Use r? to explicitly pick a reviewer" and a small smiley face emoji.

rustbot commented last week

Member ⋮

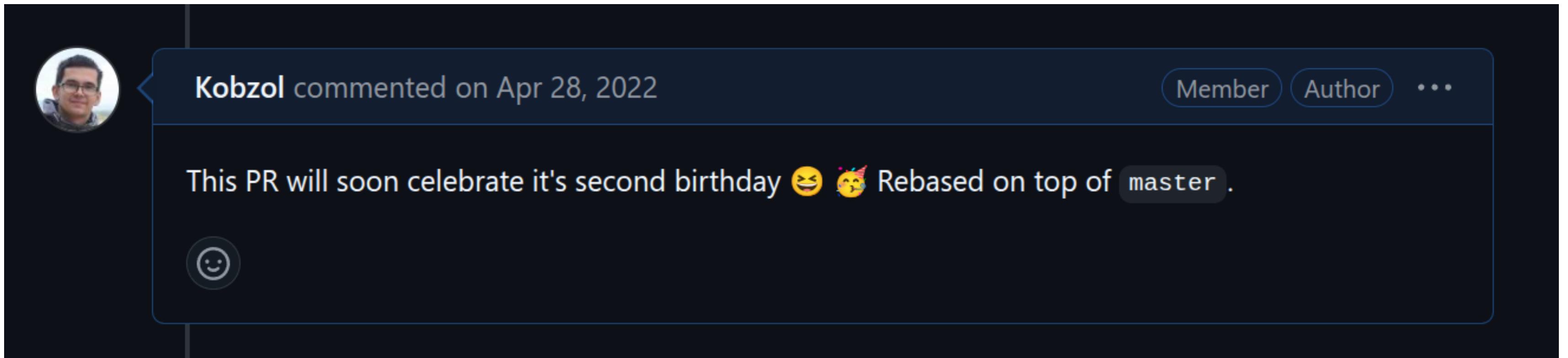
r? @Mark-Simulacrum

rustbot has assigned @Mark-Simulacrum.
They will have a look at your PR within the next two weeks and either review your PR or reassign to another reviewer.

Use r? to explicitly pick a reviewer

😊

Sending a PR



A screenshot of a GitHub pull request comment. The comment is from a user named Kobzol, posted on April 28, 2022. The comment text reads: "This PR will soon celebrate it's second birthday 😂 🎉 Rebased on top of master." There is a small smiley face emoji icon below the comment text.

Kobzol commented on Apr 28, 2022

This PR will soon celebrate it's second birthday 😂 🎉 Rebased on top of master .

Member Author ...



Sending a PR



The screenshot shows a GitHub pull request timeline with the following events:

- Urgau commented yesterday**: Urgau added the `s-waiting-on-author` label.
- @rustbot author**: A bot comment.
- rustbot added the `s-waiting-on-author` label yesterday**: rustbot added the `s-waiting-on-author` label.
- dangling ptr lint cleanup**: A commit by **hkBst** with a green checkmark and hash `0bd6a03`.
- rust-cloud-vms (bot) force-pushed the `dangling-ptr-lint-1` branch from `ce22bdc` to `0bd6a03`**: A force-push by rust-cloud-vms (bot) from branch `ce22bdc` to `0bd6a03`, made 20 hours ago. A "Compare" link is provided.
- hkBst commented 20 hours ago**: hkBst responded with `@rustbot ready`.



Sending a PR



 **rustbot** commented yesterday Member ...

This PR was rebased onto a different master commit. Here's a [range-diff](#) highlighting what actually changed.

Rebasing is a normal part of keeping PRs up to date, so no action is needed—this note is just to help reviewers.





Sending a PR



A screenshot of a GitHub pull request interface. The top comment is from **jhpratt**, posted 4 hours ago, reading "@bors try @rust-timer queue". Below it is a small smiley face emoji. The bottom comment is from **rust-timer**, also posted 4 hours ago, reading "Awaiting bors try build completion." and "@rustbot label: +S-waiting-on-perf". This comment also has a small smiley face emoji below it. The GitHub interface includes standard UI elements like user avatars, status indicators (Member, Author, three-dot menu), and a dark theme background.

jhpratt commented 4 hours ago

@bors try @rust-timer queue

rust-timer commented 4 hours ago

Awaiting bors try build completion.

@rustbot label: +S-waiting-on-perf

Sending a PR

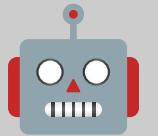


	Range	Mean	Count	?
✗	[+0.04%, +18.35%]	+1.21%	136 (42)	
✓	[-20.35%, -0.03%]	-2.43%	329 (47)	
✗,✓	[-20.35%, +18.35%]	-1.36%	465 (51)	

Aggregations ►

Primary benchmarks ↓

	Benchmark	Profile	Scenario	Backend	Target	% Change	Significance Threshold ?	Significance Factor ?
►	libc-0.2.172	check	full	llvm	x64	-16.02%	0.08%	212.46x
►	libc-0.2.172	debug	full	llvm	x64	-15.67%	0.07%	231.42x
►	libc-0.2.172	opt	full	llvm	x64	-15.45%	0.06%	239.83x
►	libc-0.2.172	check	incr-full	llvm	x64	-15.40%	0.05%	282.64x
►	libc-0.2.172	debug	incr-full	llvm	x64	-15.04%	0.07%	224.52x
►	libc-0.2.172	opt	incr-full	llvm	x64	-14.79%	0.06%	238.06x
►	bitmaps-3.2.1	check	incr-unchanged	llvm	x64	-12.02%	0.14%	85.56x
►	bitmaps-3.2.1	debug	incr-unchanged	llvm	x64	-11.38%	0.14%	81.75x
►	bitmaps-3.2.1	opt	incr-unchanged	llvm	x64	-11.35%	0.17%	65.66x
►	bitmaps-3.2.1	check	full	llvm	x64	-10.96%	0.03%	423.78x
►	bitmaps-3.2.1	check	incr-full	llvm	x64	-10.00%	0.03%	328.54x
►	bitmaps-3.2.1	debug	full	llvm	x64	-9.76%	0.06%	156.49x



bors

Sending a PR



 ✓ Urgau approved these changes 19 hours ago [View reviewed changes](#)

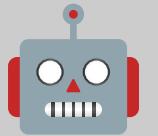
 Urgau commented 19 hours ago Member ...

 @bors r+
 

 bors commented 19 hours ago Member ...

 ➡ Commit `0bd6a03` has been approved by Urgau
 It is now in the [queue](#) for this repository.
 

 bors added `s-waiting-on-bors` and removed `s-waiting-on-review` labels 19 hours ago



bors



Not Rocket Science Rule

The Not Rocket Science Rule Of Software Engineering:

Automatically maintain a repository of code that always passes all the tests.



CI costs





CI costs



For each merged PR:

- ~3 hours walltime



CI costs

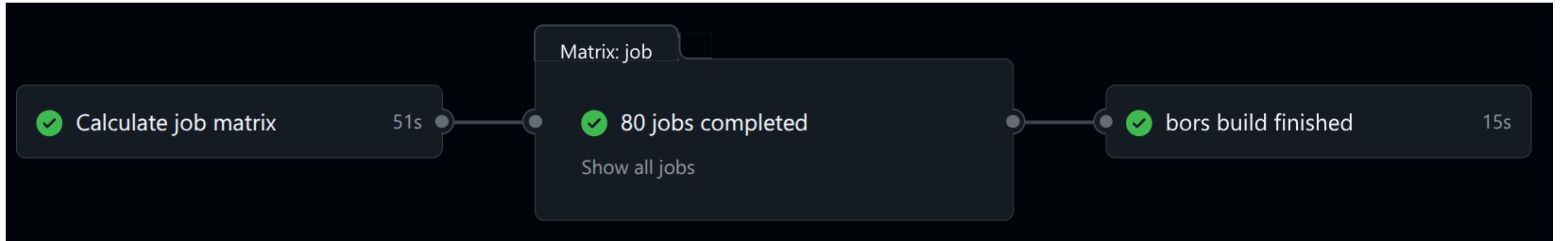


For each merged PR:

- ~3 hours walltime
- ~150 runner hours



CI costs

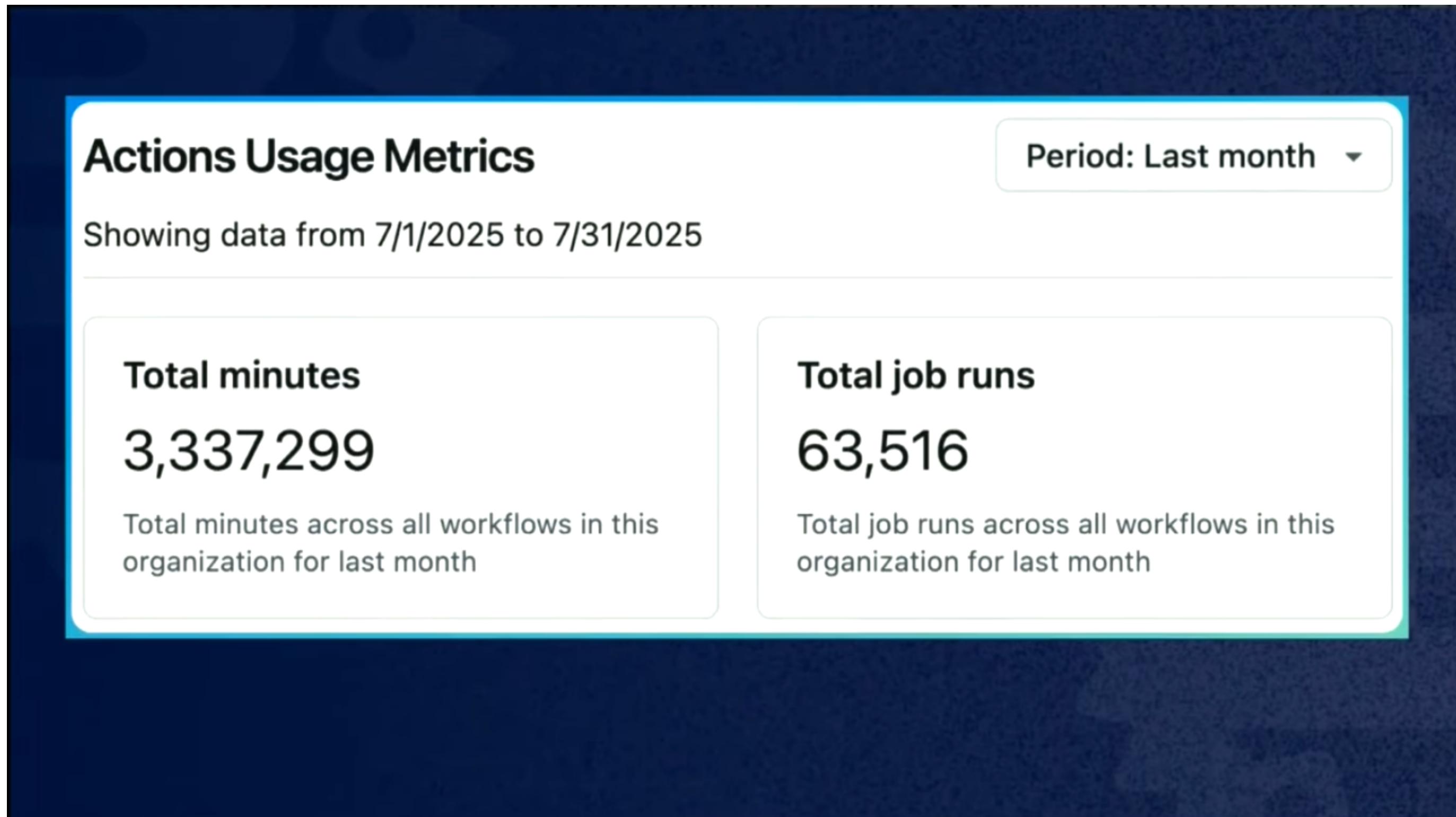


For each merged PR:

- ~3 hours walltime
- ~150 runner hours
- ~600 (!) CPU hours



CI costs





CI costs

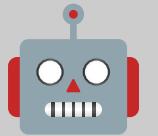
- \$140 per merged PR
- \$50 000 per month!



The screenshot shows the GitHub repository page for `rust-lang / rust`. The interface includes a sidebar with a menu icon, the GitHub logo, and the repository name. Below the repository name, there are four main navigation links: `<> Code`, `Issues` (with a count of `5k+`), `Pull requests` (with a count of `899`), and a link to the `Code` section. A horizontal progress bar is visible at the bottom of the page.

≡  rust-lang / rust

<> Code Issues 5k+ Pull requests 899



bors

Rollups



Rollup of 10 pull requests #148356

Merged

bors merged 166 commits into `rust-lang:master` from `matthiaskrgr:rollup-mh4l2yi` 16 hours ago

Conversation 10

Commits 166

Checks 11

Files changed 300+



matthiaskrgr commented yesterday • edited by rustbot

Member ...

Successful merges:

- ↳ [Tweak output of missing lifetime on associated type](#) #135602 (Tweak output of missing lifetime on associated type)
- ↳ [Implement pin-project in pattern matching for &pin mut|const T](#) #139751 (Implement pin-project in pattern matching for `&pin mut|const T`)
- ↳ [Update bundled musl to 1.2.5](#) #142682 (Update bundled musl to 1.2.5)
- ↳ [Simplify code to generate line numbers in highlight](#) #148171 (Simplify code to generate line numbers in highlight)
- ↳ [Unpin libc and rustix in compiler and rustbook](#) #148263 (Unpin `libc` and `rustix` in `compiler` and `rustbook`)
- ↳ [\[rustdoc search\] Include extern crates when filtering on import](#) #148301 ([rustdoc search] Include extern crates when filtering on `import`)
- ↳ [Don't require dlltool with the dummy backend on MinGW](#) #148330 (Don't require dlltool with the dummy backend on MinGW)
- ↳ [cleanup: upstream dropped amx-transpose functionality](#) #148338 (cleanup: upstream dropped amx-transpose functionality)
- ↳ [Clippy subtree update](#) #148340 (Clippy subtree update)
- ↳ [nonpoison::Condvar should take MutexGuard by reference](#) #148343 (`nonpoison::Condvar` should take `MutexGuard` by reference)



Empowering new contributors



Rust Compiler Development Guide

Getting Started

Thank you for your interest in contributing to Rust! There are many ways to contribute, and we appreciate all of them.

If this is your first time contributing, the [walkthrough](#) chapter can give you a good example of how a typical contribution would go.

This documentation is *not* intended to be comprehensive; it is meant to be a quick guide for the most useful things. For more information, [see this chapter on how to build and run the compiler](#).



```
/tm/rust [master]$ ./x setup
Building bootstrap
  Finished `dev` profile [unoptimized] target(s) in 0.04s
Welcome to the Rust project! What do you want to do with x.py?
a) library: Contribute to the standard library
b) compiler: Contribute to the compiler itself
c) tools: Contribute to tools which depend on the compiler, but
          don't have their own repository
d) dist: Install Rust from source
e) none: Do not modify `bootstrap.toml`
Please choose one (a/b/c/d/e): █
```



rustbot commented on Sep 25, 2024

Member

...

Thanks for the pull request, and welcome! The Rust team is excited to review your changes, and you should hear from [@chenyukang](#) (or someone else) some time within the next two weeks.

Please see the [contribution instructions](#) for more information. Namely, in order to ensure the minimum review times lag, PR authors and assigned reviewers should ensure that the review label (`s-waiting-on-review` and `s-waiting-on-author`) stays updated, invoking these commands when appropriate:

- `@rustbot author` : the review is finished, PR author should check the comments and take action accordingly
- `@rustbot review` : the author is ready for a review, this PR will be queued again in the reviewer's queue





rustbot commented last week

Member

...

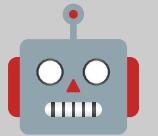
⚠ Warning ⚠

- The following commits have merge commits (commits with multiple parents) in your changes. We have a [no merge policy](#) so these commits will need to be removed for this pull request to be merged.
 - [85f3ca8](#)

You can start a rebase with the following commands:

```
$ # rebase  
$ git pull --rebase https://github.com/rust-lang/rust.git master  
$ git push --force-with-lease
```





bors



oli-obk commented yesterday

Member

...

@bors delegate=hkBst



bors commented yesterday

Member

...

👉 @hkBst, you can now approve this pull request!

If @oli-obk told you to " r=me " after making some further change, please make that change, then do @bors r=@oli-obk





Dev Desktops

The dev desktops provide maintainers and contributors to the Rust Project with free access to high-powered cloud compute. They are part of the [Cloud Compute Program](#) by the [Rust Foundation](#).

Machine	Architecture	Perf enabled	Location
dev-desktop-eu-1	aarch64	Yes	Germany
dev-desktop-us-1	aarch64	Yes	N. Virginia, US



Imposter Syndrome

Apr. 19, 2022 · Jane Lusby, Project Director of Collaboration on behalf of [Rust Foundation Project Directors](#)

Preface: This is in response to some feedback the project directors received from the Rust Foundation staff. Some of the contributors they'd talked to said they didn't feel justified in applying for Foundation grants even though they'd love the opportunity, because they don't feel qualified or deserving of them compared to the other amazing contributors they look up to within the Rust project. This was a little bit heart breaking to me, because I know exactly what that feeling is like¹, and I also know just how wrong they are.



Thanks

Rust is a community project and is very thankful for the many community contributions it receives. [See this page for more information.](#)

All time ›

10210 people

421418 contributions

Master ›

100 people

459 contributions

Beta ›

341 people

4168 contributions

Rust 1.91.0 ›

382 people

4327 contributions

Rust 1.90.0 ›

553 people

7276 contributions

Rust 1.89.0 ›

525 people

7594 contributions

Rust 1.88.0 ›

366 people

4741 contributions

Rust 1.87.0 ›

390 people

4779 contributions



All-time Contributors

[Back to all releases](#)

We have had 9802 individuals contribute to all of Rust so far. Thank you so much!

Rank	Name	Contributions
1	bors	54726



Deployment



Release train

5. 11.

6. 11.

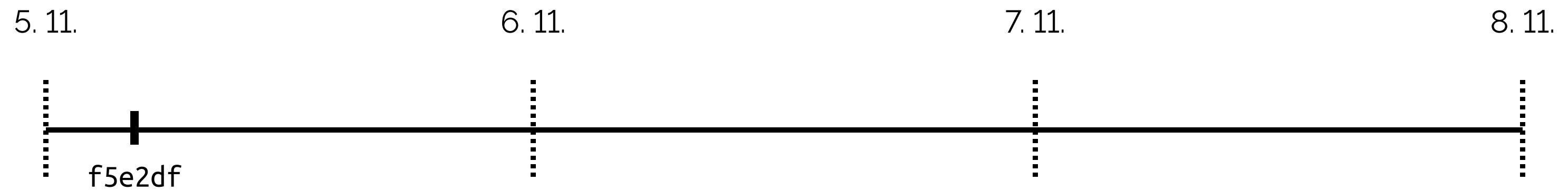
7. 11.

8. 11.



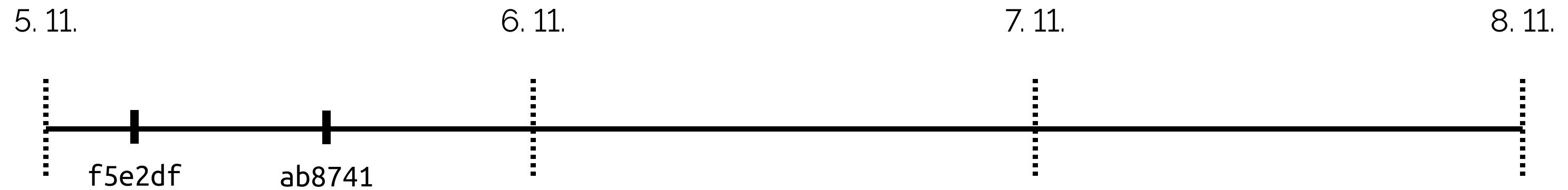


Release train



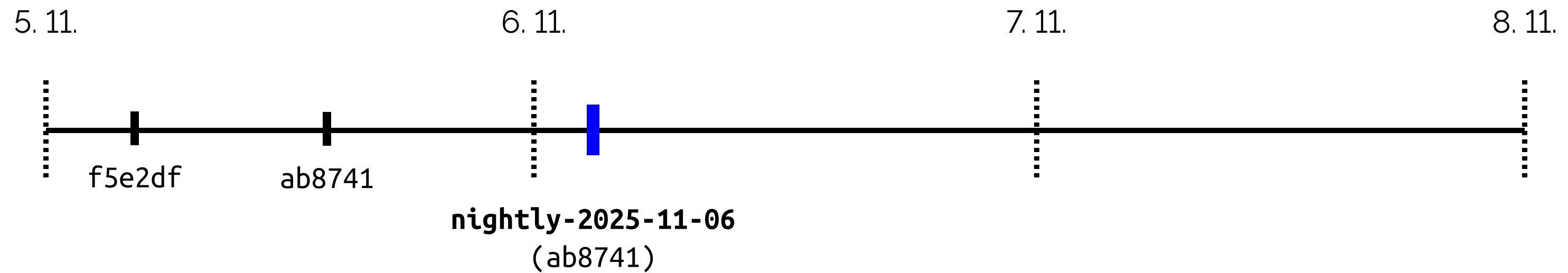


Release train





Release train



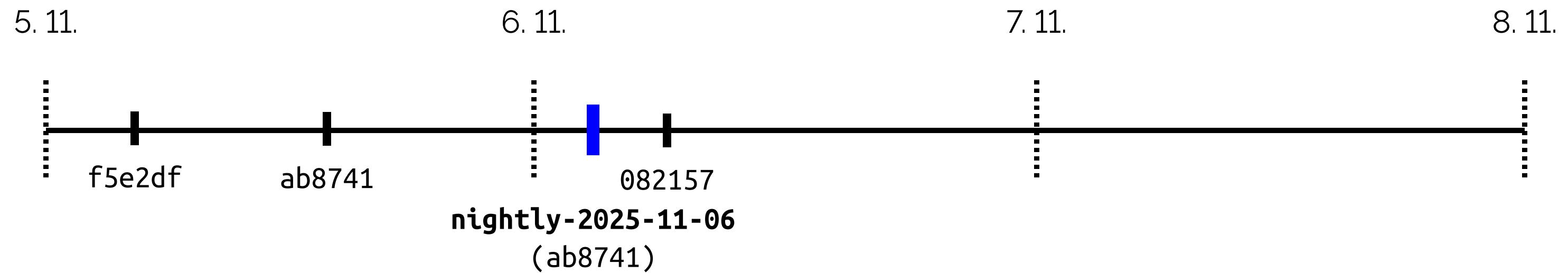


Release train

```
~$ rustup update nightly
info: syncing channel updates for 'nightly-x86_64-unknown-linux-gnu'
info: latest update on 2025-11-02, rust version 1.93.0-nightly (bd3ac0330 2025-11-01)
info: downloading component 'cargo'
  10.3 MiB / 10.3 MiB (100 %) 1.8 MiB/s in 6s
info: downloading component 'clippy'
  4.4 MiB / 4.4 MiB (100 %) 2.6 MiB/s in 1s
info: downloading component 'rust-docs'
  20.5 MiB / 20.5 MiB (100 %) 3.4 MiB/s in 6s
info: downloading component 'rust-std'
  27.9 MiB / 27.9 MiB (100 %) 3.6 MiB/s in 7s
info: downloading component 'rustc'
  74.3 MiB / 74.3 MiB (100 %) 3.6 MiB/s in 20s
```

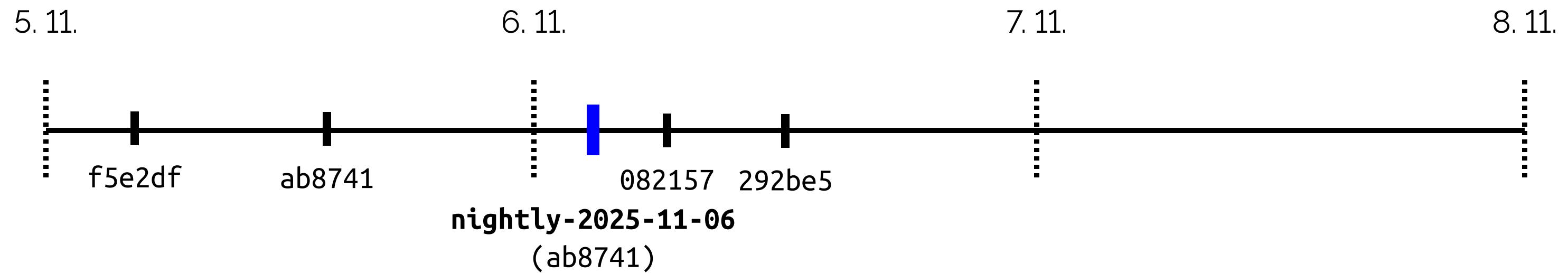


Release train



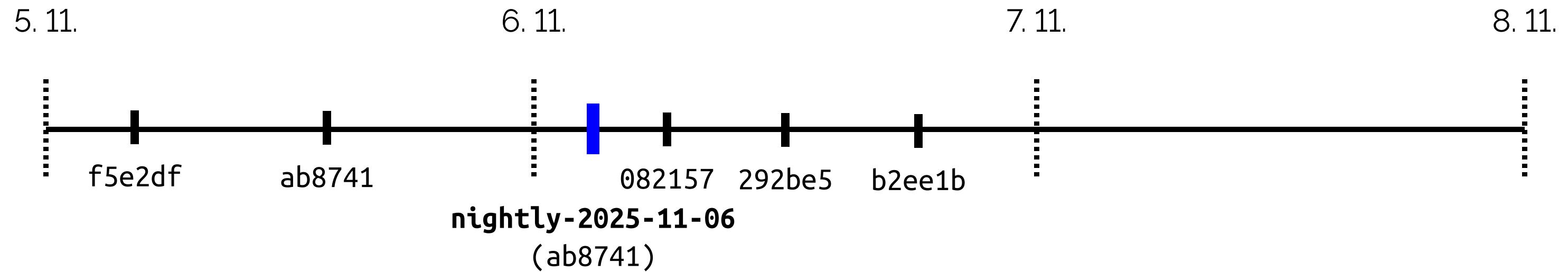


Release train



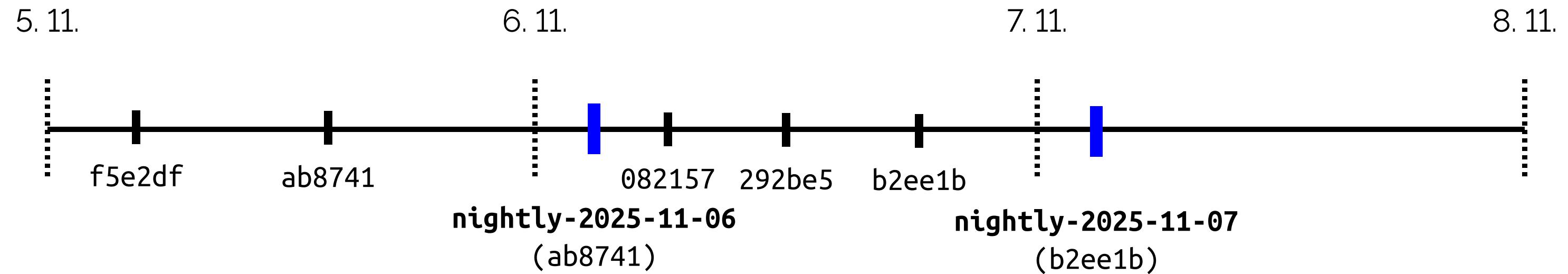


Release train





Release train





Release train

Nightly



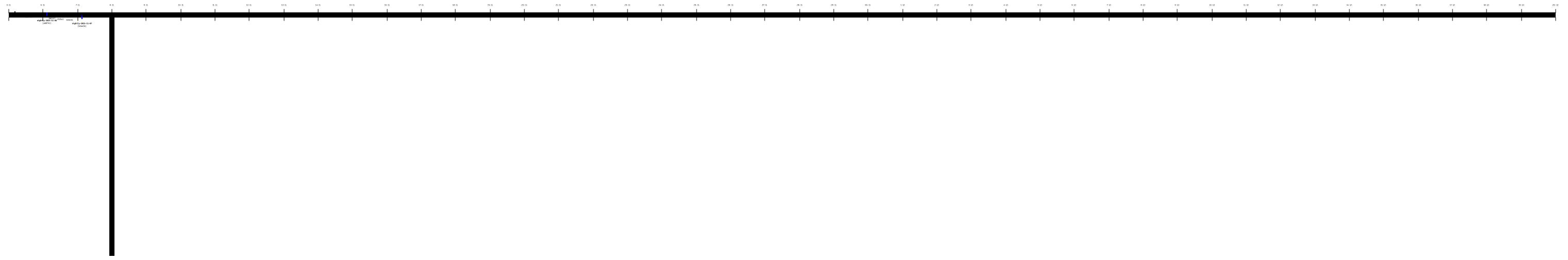
Beta

Stable



Release train

Nightly



Beta

beta-2025-11-05

Stable



Release train

Nightly



Beta

6 weeks

beta-2025-11-05

Stable



Release train

Nightly



6 weeks

beta-2025-11-05

Beta

1.91.0

Stable



Release train

Nightly

Beta

Stable



beta-2025-11-05

beta-2025-12-21

1.91.0



Release train

Nightly

Beta

Stable



6 weeks

6 weeks

beta-2025-11-05

beta-2025-12-21

1.91.0



Release train

Nightly

Beta

Stable



6 weeks

6 weeks

beta-2025-11-05

beta-2025-12-21

1.91.0

1.92.0



Release train benefits

- New stable release every 6 weeks, no matter what



Release train benefits

- New stable release every 6 weeks, no matter what
- No big deal if a feature misses the train



Release train benefits

- New stable release every 6 weeks, no matter what
- No big deal if a feature misses the train
- Reduces contributor stress



Rust keeps backward compatibility to 1.0 (2015)*

*modulo compiler and soundness bug fixes



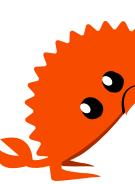
Crater

For every stable release (or on-demand):



Crater

For every stable release (or on-demand):
test ~all Rust code on GitHub and crates.io (~600k Rust crates)



m-ou-se commented on Feb 12

@craterbot run mode=build-and-test

😊

craterbot commented on Feb 12

💡 Experiment [pr-136932](#) created and queued.
🤖 Automatically detected try build [7af7790](#)
🔍 You can check out [the queue](#) and [this experiment's details](#).

ℹ️ Crater is a tool to run experiments across parts of the Rust ecosystem. [Learn more](#)

😊

craterbot added [S-waiting-on-crater](#) and removed [S-experimental](#) labels on Feb 12

craterbot commented on Feb 13

🎉 Experiment [pr-136932](#) is completed!
📊 172 regressed and 113 fixed (582049 total)
🔗 Open the full report.

⚠️ If you notice any spurious failure please add them to the denylist!

ℹ️ Crater is a tool to run experiments across parts of the Rust ecosystem. [Learn more](#)

😊



Summary Full report Downloads

[552a959051cebf8f88a8f558399baf733bec9ce0](#)

[7af779037716ae4125ceabb429791b4cf5dd0a43](#)

error (164352)

spurious-fixed (1690)

spurious-regressed (2751)

regressed: dependencies (2 root crates, 172 regressed crates in total)

regressed: root results (2 different results, 172 regressed crates in total)

regressed build failed (unknown) (18)

Crate	Previous run	Current run
Gabriel2409.advent_of_code_2023.54636eb8552ed0426837780350bc715ed2e28e3b	● test failed (unknown)	● build failed (unknown)
dswd.vpncloud.rs.bef99162fe0d66e2c9fbf6421a69ca8ec2ac8f55	● test failed (unknown)	● build failed (unknown)
ecumene.discord_spin.fd3f5c18c734a704ce1473c8f89f9551542056ca	● test passed	● build failed (unknown)
ecumene.spin_pg.594a522b1d7895c1573c33bb2d6f48ea215d76f4	● test passed	● build failed (unknown)
einoj.ratsim.92738d4f590b915f8c8ec4c8213ad4758a06e374	● test passed	● build failed (unknown)
frol.rust-benchmark-option-replace_with-rfc.a79f0a3c095e82cb31849a723bc96b064ef944dc	● test passed	● build failed (unknown)
gtrafimenkov.git-rcrypt.924a89b1537d038aa9705bc8b9b8ad2a3b7431de	● test passed	● build failed (unknown)
gugz0r.rouillefwd.f09def8d3e92ff13888439d4e4e121894141bda4	● test passed	● build failed (unknown)



Stability without stagnation

- New Rust edition every 3 years



Stability without stagnation

- New Rust edition every 3 years
- **Opt-in** backwards incompatible changes



Remove object lifetime cast #1

Open

BoxyUwU wants to merge 1 commit into `Thombrom:master` from `BoxyUwU:patch-2`

Conversation 0

Commits 1

Checks 0

Files changed 1



BoxyUwU commented on Mar 13

...

Hi there o/

I'm a member of the Rust Language Types Team; as part of stabilizing the `arbitrary_self_types` and `derive_coerce_pointee` features we may need to change what raw pointer casts are legal ([rust-lang/rust#136702](#)). Specifically, casting `*const dyn Trait + 'a` to `*const dyn Trait + 'b` where it is not able to be proven that `'a` outlives `'b` may become an error.

Without going into too much detail, the justification for this change be that casting trait object's lifetime bound to a lifetime that lives for a longer time could invalidate the VTable of the trait object, allowing for dispatching to methods that should not be callable. See this example from the linked issue:



[BoxyUwU](#) mentioned this in 12 pull requests on Mar 13

- ↳ Remove object lifetime cast [metrics-rs/metrics#564](#)
- ↳ Remove object lifetime cast [Xudong-Huang/may#113](#)
- ↳ Remove object lifetime cast [Thombrom/snek#1](#)
- ↳ Remove object lifetime cast [Wind-Corporation/trapiron#5](#)
- ↳ Remove object lifetime cast [WorldSEnder/scoped_worker_thread#1](#)
- ↳ Remove object lifetime cast [maroider/async_scoped_task#1](#)
- ↳ Remove object lifetime cast [orengine/orengine#33](#)
- ↳ Remove object lifetime and auto trait cast [gipsyh/PPSMC#1](#)
- ↳ Remove object lifetime casts [vitorhnn/bfr#1](#)
- ↳ Remove object lifetime cast [luksan/lox#1](#)
- ↳ Remove object lifetime casts [tyilo/multi-vec#1](#)
- ↳ Remove object lifetime cast [belalang-project/belalang#6](#)

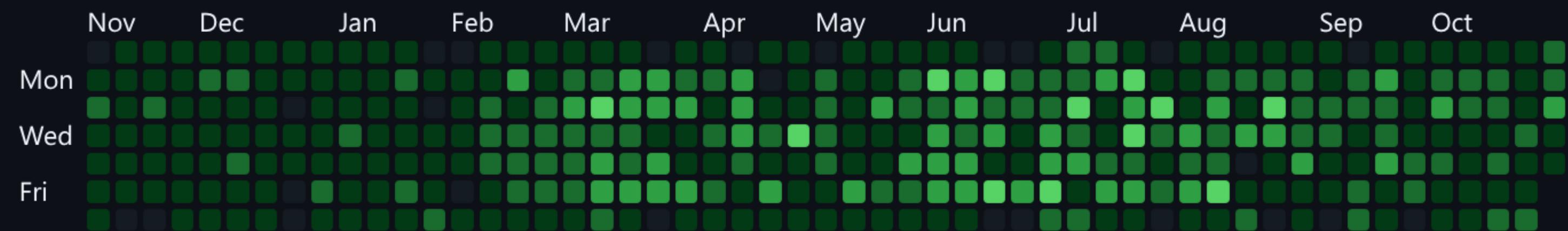


Maintenance



6,857 contributions in the last year

Contribution settings ▾



Learn how we count contributions

Less More



Daniel Stenberg

Maintaining is doing many things

Security work

PR reviews

Debugging

Write documentation

Release management

User support

PR merging

Event planning

Website admin

Blogging & information

CI maintenance

Getting stickers

Mailing list admin

People management

Find sponsors

Doing talks

Feature development

FrOSCon



Daniel Stenberg @ FrOSCon



Maintaining stuff is hard!



Maintaining stuff is hard!

- ~300 repositories managed by the Rust Project



Maintaining stuff is hard!

- ~300 repositories managed by the Rust Project
- Unglamorous (but crucial!) work



Rule of two



"ALWAYS
TWO THERE
ARE,
A ~~MASTER~~
AND AN
~~APPRENTICE.~~"

Maintainer
Implementer



Announcing the Rust Foundation Maintainers Fund

November 4, 2025 / Rust Foundation Team

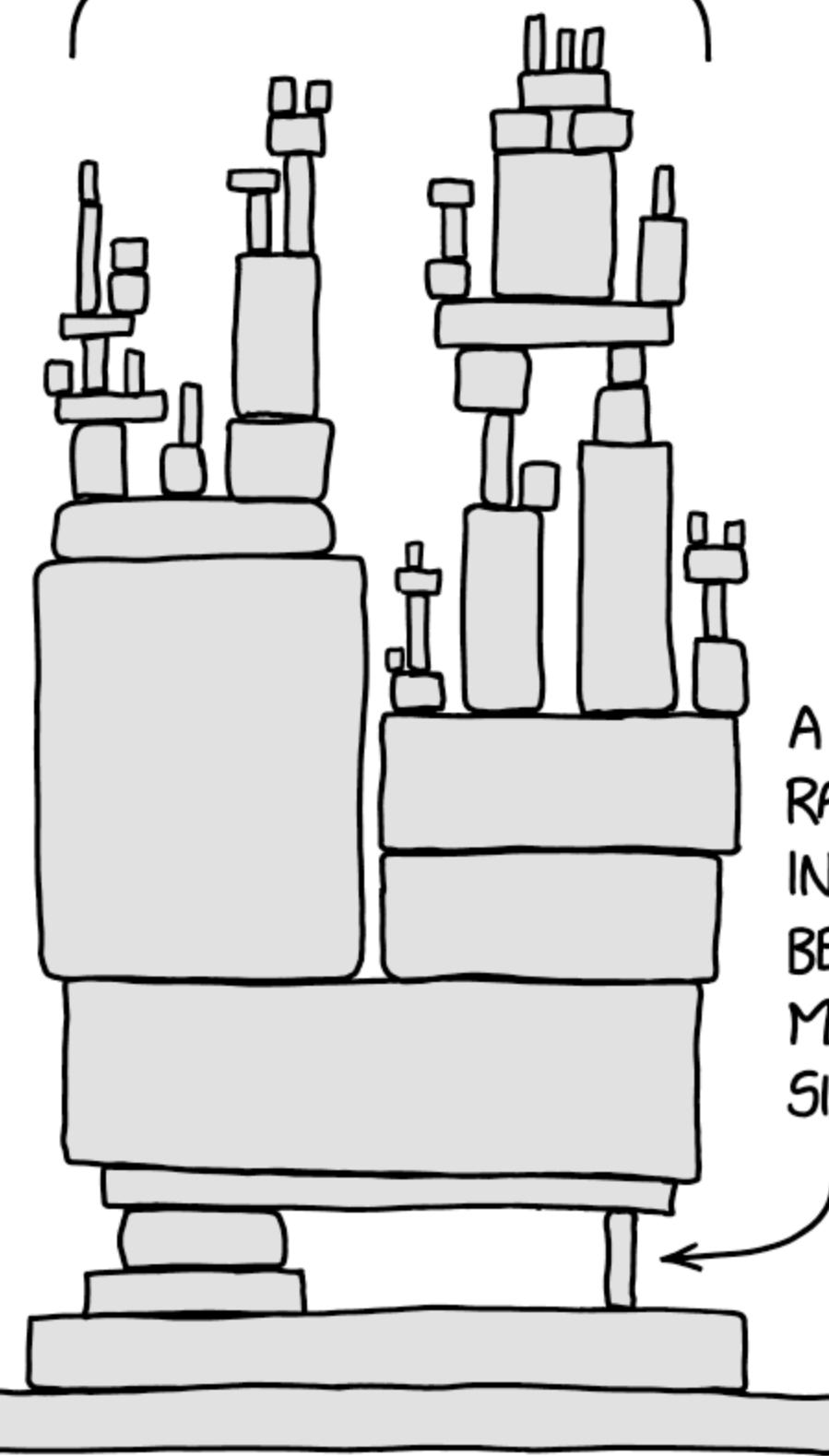
Rust Maintainers Fund

Support Rust's maintainers

RustNL is seeking sponsors to establish a Rust Maintainers Fund in 2026.



ALL MODERN DIGITAL
INFRASTRUCTURE



A PROJECT SOME
RANDOM PERSON
IN NEBRASKA HAS
BEEN THANKLESSLY
MAINTAINING
SINCE 2003



Open-source maintenance





Open-source maintenance





Open-source maintenance





Open-source maintenance





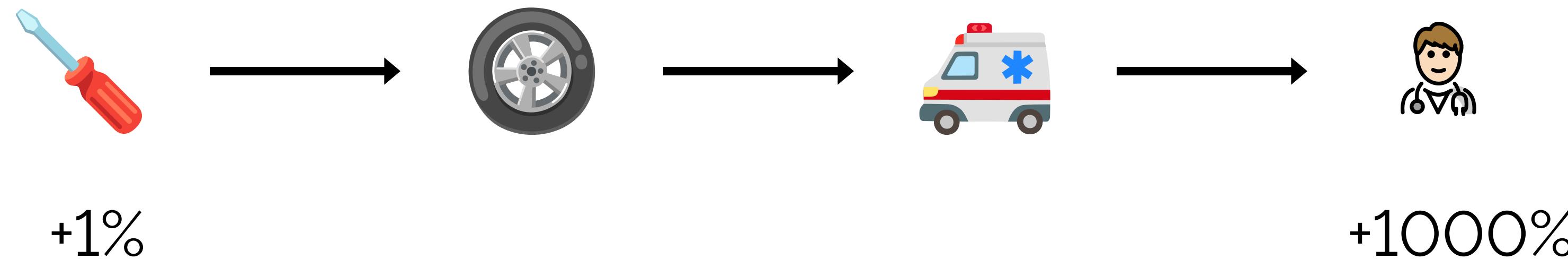
Open-source maintenance



+1%



Open-source maintenance



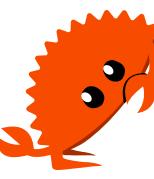


Do you depend on open-source?



Do you depend on open-source?

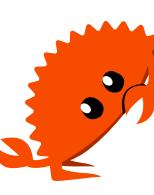
(yes, you do :-))



Do you depend on open-source?

(yes, you do :-))

Try to give back!



Every contribution counts



Every contribution counts

- Create an issue



Every contribution counts

- Create an issue
- Upstream fixes/features



Every contribution counts

- Create an issue
- Upstream fixes/features
- Improve documentation



Every contribution counts

- Create an issue
- Upstream fixes/features
- Improve documentation
- Discuss things, write blog posts



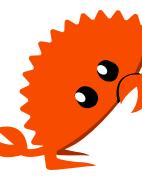
Every contribution counts

- Create an issue
- Upstream fixes/features
- Improve documentation
- Discuss things, write blog posts
- Give credit



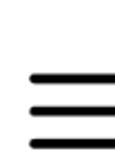
Every contribution counts

- Create an issue
- Upstream fixes/features
- Improve documentation
- Discuss things, write blog posts
- Give credit
- Be kind!



Every contribution counts

- Create an issue
- Upstream fixes/features
- Improve documentation
- Discuss things, write blog posts
- Give credit
- Be kind!
- Sponsor maintainers



thanks.dev

For sponsors

**sponsor the open source
projects you depend on.**



Become a sponsor to lcnr



lcnr

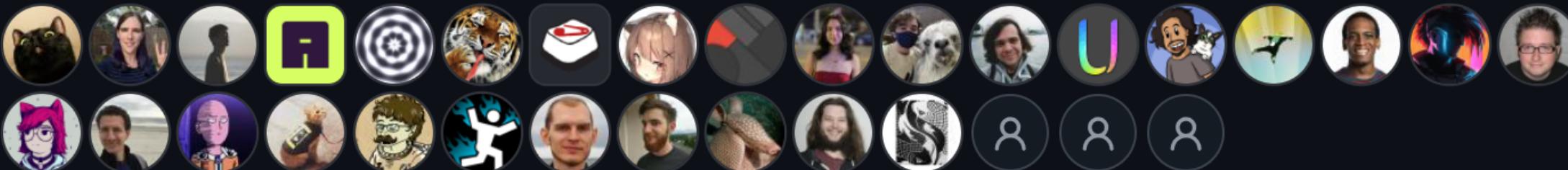
Germany

Hey 

I am working on the Rust programming language and am co-lead of the [types team](#), a member of the [compiler team](#), and team lead of the [trait system refactor initiative](#) and the [const generics project group](#).

You may also know me from smaller contributions to other projects, from social media, and from blog posts.

Current sponsors 32





is open-source = ❤



is open-source = ❤

rust-lang.zulipchat.com



is open-source = ❤

rust-lang.zulipchat.com

rustlang.cz



Thank you for your attention!

Slides are available here:



Slides were programmed using github.com/spirali/elsie

Several emojis were used from the Noto Emoji pack