Rust Foundation

Technology Report

February-August 2024

CONTENTS

Foreword by Dr. Rebecca Rumbul,

Rust Foundation Executive Director & CEO

As the nonprofit steward of the Rust programming language, the Rust Foundation has a duty not only to support and champion the Rust community and Project, but also to contribute meaningful packages of work to Rust and the open source ecosystem, to enable Rust to be truly beneficial for everyone. I speak for the entire Rust Foundation team when I say that it is a privilege for us to witness and support the impressive growth, progress, and evolution of the Rust programming language over the past few years.

Although we are still a young foundation, there is much to be proud of in the three short years since our inception. The Rust Foundation Technology Team, led by Joel Marcey, is responsible for a variety of impactful and innovative programmes and countless collaborative hours working with the Rust Project. I'm thrilled to present a recap of our Tech Team's excellent work to date in 2024.

For those who have followed our communications closely, you will recall that the Rust Foundation published two Security Initiative Reports between 2023 and early 2024. These reports catalogued our contributions to Rust ecosystem security and contextualised the nature of our collaboration with specific Rust Project teams and working groups.

Since releasing our most recent Security Initiative report in February of this year, our team has grown and we have been able to allocate meaningful funding to create new technical initiatives and further expand upon existing efforts. For those reasons, it has become necessary to expand our team reporting to include updates on all of our technical programmes. I hope that our community is able to use the Technology Report series as a helpful tool to track our engineering activities and contributions to Rust over time.

Thank you to all of our supporters for enabling us to deliver impactful work in supporting the Rust programming language, and to our team here at the Rust Foundation in working tirelessly to fulfill our mission of serving the Rust language and community!

Rebecca Rumbul
Executive Director & CEO of the Rust Foundation

Executive Summary by Joel Marcey,
Rust Foundation Director of Technology

The past eight months have been particularly exciting on the Rust Foundation's Technology Team. The accomplishments detailed in the following pages serve as evidence that our organization is maturing, continuing to identify areas of positive impact, and maintaining successful programs established previously.

Recent Tech Highlights Include:

- Hiring and onboarding two new Tech Team members: Rust-C++ Interoperability Engineer Jon Bauman & Infrastructure Engineer Marco Ieni. (page 14)
- Securing $1M in funding from Rust Foundation Platinum Member Google for our Interop Initiative. (page 10)
- Forming the Safety-Critical Rust Consortium in concert with several other organizations and Rust Foundation members. (page 9)

Current Tech Team Priorities Include:

- Making further progress within the Security Initiative on supply-chain security, including crate provenance, malicious crate detection and crate signing.
- Defining C++/Rust Interop Initiative strategy and goals in close collaboration with the Rust Project.
- Continuing to support stable and efficient infrastructure, including backups of Rust releases and crates.
- Officially launching the Safety Critical Consortium at RustConf.
- Making substantial progress on the Rust specification.

Joel

Joel Marcey
Director of Technology at the Rust Foundation

Rust Foundation Technology Team

The programs detailed in the report were managed and made possible by the following Rust Foundation staff members...

Joel Marcey

Director of Technology

Jon Bauman

Rust/C++ Interoperability Engineer

Tobias Bieniek

Software Engineer (crates.io), crates.io Team Lead

Jan David Nose

Infrastructure Engineer, Infrastructure Team Lead

Marco Ieni

Infrastructure Engineer, Infrastructure Team Member

Walter Pearce

Security Engineer, RustSec Advisor

Adam Harvey

Software Engineer (Security), crates.io Team Member

Thank you to Adam, Jan David, Joel, Jon, Marco, Tobias, and Walter for their contributions to the safety, security, interoperability, performance, and good working order of the Rust programming language!

Acknowledgements

The Tech Team could not be successful without transparent and productive relationships with hard working members of the Rust Project, so the Rust Foundation would like to thank all the maintainers who have been willing to work and communicate openly with us about our technical endeavors. Many of these Rust maintainers have been donating their time and skills to the Rust programming language before the Rust Foundation existed and we would be remiss not to mention this alongside the report on the Foundation's technical activities.

Special thanks to the members of these Rust Project Teams and Working Groups:

Leadership Council | **Infrastructure Team** | **crates.io Team** | **Security Response Working Group** | **Secure Code Working Group**

Special Funding Update

In 2024, several generous donations enabled the Rust Foundation's Technology Team to carry out critical initiatives:

Google

$1M USD

In February 2024, the Rust Foundation announced a $1M donation from Platinum Member Google to fund our Rust/C++ Interop Initiative. Learn more about this project on page 10.

Microsoft

$1M USD

In May 2024, the Rust Foundation announced a $1M unrestricted donation from Platinum Member, Microsoft. These funds enabled us to hire our second Infrastructure Engineer, Marco Ieni, and are being allocated to support the Community Grants Program and Rust Project and language priorities via the Foundation.

Alpha-Omega

$460k USD

In November 2023, OpenSSF's Alpha-Omega Project announced a second year of supporting our Security Initiative. To date, they have contributed $920k towards this important work. Our Security Initiative accomplishments between February and August 2024 were made possible by Alpha-Omega's most recent $460k contribution.

LAMBDA CLASS

$100k USD

In August 2024, the Rust Foundation received a $100K donation from LambdaClass ("Lambda") — a deep tech venture studio and investment firm with expertise in a variety of industries, including gaming, machine learning, cryptography, database design, distributed systems, programming languages, compiler development, high-frequency trading, real-time bidding, embedded devices, aerospace, and biotech. This donation was made by Lambda to support the ongoing development and maintenance of the Rust programming language. We are grateful for their support of our mission.

Further reading:

https://foundation.rust-lang.org/news/google-contributes-1m-to-rust-foundation-to-support-c-rust-interop-initiative/
https://foundation.rust-lang.org/news/1m-microsoft-donation-to-fund-key-rust-foundation-project-priorities/
https://foundation.rust-lang.org/news/alpha-omega-to-continue-support-of-rust-foundation-security-initiative-in-2024/ https://foundation.rust-lang.org/news/lambdaclass-donates-100k-to-the-rust-foundation/

Crates.io Contributions

February-August 2024 Update

The Rust Foundation contributes maintenance and feature development for crates.io: the official Rust package registry. Below is an update on the Rust Foundation Tech Team's recent work in this domain.

Our recent crates.io contributions were carried out and supported by the following Rust Foundation team members in particular:

- **Tobias Bieniek** – Rust Foundation Software Engineer, Co-Lead of the Rust Project crates.io Team, and longtime Rust Project contributor.

- **Adam Harvey** – Rust Foundation Software Engineer, crates.io Team member

Thanks to the entire Rust Project crates.io team for their ongoing collaboration and contributions to these efforts.

Sponsorship

The Rust Foundation's recent contributions to crates.io would not have been possible without the generous financial support of the Rust Foundation Security Initiative page 11 by the Alpha-Omega Project and Rust Foundation Platinum Member, AWS.

Recent Highlights

Admin Functionality

In March, Software Engineer Adam Harvey continued to work on adding crates.io admin functionality, ultimately creating and merging a PR that added a "sudo-mode" for admins logged into crates.io. Actions that require admin privileges are now disabled by default and must be explicitly granted. Privileges are short-lived and not synced across multiple browser sessions.

Associated PR: https://github.com/rust-lang/crates.io/pull/8210

Crate Download Changes

Counting crate downloads based on CDN log files has been running in-production since Q2, indirectly allowing cargo to download crates directly from static.crates.io.

Thanks to the log-based download counting change, crates downloads will keep working even if the crates.io API is experiencing issues. The reliability and speed of crates downloads has greatly improved.

The Rust Project crates.io team and our team believe these changes are making crates.io more reliable and scalable overall.

Further Reading: https://blog.rust-lang.org/2024/03/11/crates-io-download-changes.html

Test Suite Migration

In April, the crates.io test suite was migrated to async tests to make it easier to use async-only libraries in the test suite.

Associated PR: https://github.com/rust-lang/crates.io/pull/8498

Database Improvements

Also in April, the crates.io team began to investigate the remaining API calls, particularly some performance issues, and began working on potential solutions. A scaling issue related to the increased number of packages on crates.io was discovered. As a first step to address this, Tobias introduced a default_versions database table. Next, a new archive version download background job was created and old, unneeded data was archived, ultimately allowing the crates.io team to shrink the crates.io database significantly and improve performance.

Associated PRs: https://github.com/rust-lang/crates.io/pull/8484

https://github.com/rust-lang/crates.io/pull/8596

Tobias also helped migrate the crates.io database servers to a new provider. Previously, some parts of the website were taking multiple seconds to load, but since the change, the server typically responds in less than a second.

Further Reading: https://blog.rust-lang.org/2024/07/29/crates-io-development-update.html#database-performance-optimizations

Cargo install

Tobias and the crates.io team implemented a change to make it easier for users to install crates that have "binary targets". If a crate has binary targets, the names of the binaries will now be saved in the crates.io database and then conveniently displayed on the crate's page – unless a user wants targets excluded.

Summary

As the official package registry for the Rust programming language, crates.io plays a critical role in the Rust ecosystem. Aside from being the place Rust users share their own libraries and packages, crates.io makes it easier for other developers to find tools that solve specific issues they are encountering and expand upon their own work.

Given this significance, the Rust Foundation is proud to play a role in funding and supporting improvements to make crates.io more secure, organized, and performant. Thank you again to everyone who played a role in the milestones listed above.

Safety-Critical Rust Consortium

In June, we announced a new group dedicated to the responsible use of Rust in safety-critical software, hosted by the Rust Foundation. This news was shared in partnership with founding participants and Rust Foundation Members AdaCore, Arm, Ferrous Systems, HighTec EDV-Systeme GmbH, Lynx Software Technologies, OxidOS, TECHFUND, and Veecle, as well as TrustInSoft and Woven by Toyota.

The primary objective of this group is to support the responsible use of the Rust programming language in safety-critical software — systems whose failure can impact human life or cause severe environmental or property harm.

Following the news announcement, we received an astounding amount of interest from industry press outlets as well as organizations seeking to participate. Media highlights include:

SD Times

- [Safety-Critical Rust Consortium forms at the Rust Foundation]

THE NEW STACK

- [Rust: The Future of Fail-Safe Software Development]

siliconANGLE

- [Rust Foundation and industry leaders launch Safety-Critical Rust Consortium]

Safety-Critical Rust Consortium Membership is open to Rust Foundation member organizations and other invitees, such as industry, academic, and legal experts. The founding members have an initial meeting planned for September 2024 in Montreal, Canada where they will discuss the proper process to onboard new member organizations and the creation of a public charter and goals. The Safety-Critical Rust Consortium will liaise with the Rust Project through Rust Foundation Project Directors and members of Rust Project teams.

We look forward to keeping the public up to date on progress related to this important stream of work.

Further Reading: https://foundation.rust-lang.org/news/announcing-the-safety-critical-rust-consortium/

Rust-C++ Interoperability Initiative

Improving the state of interoperability between the Rust and C++ programming languages.

In February, the Rust Foundation announced a new effort to improve the state of interoperability between the Rust and C++ programming languages, made possible through a generous contribution from Platinum Member, Google. Since then, we have made meaningful initial progress thanks to Rust-C++ Interop Engineer Jon Bauman and collaboration with the Rust Project and key stakeholders.

Support & Sponsorship

The Interop Initiative was made possible through a contribution of $1M from Rust Foundation Platinum Member, Google.

> [Google is] supporting the Rust Foundation's Interop Initiative because greater interoperability with C++ will be key to Rust's adoption and allow for more organizations and communities to benefit from memory-safe systems.
>
> - ***Lars Bergstrom,*** Rust Foundation Board Chair and Member Director for Google

Progress

In June, 2024, Jon Bauman joined the Rust Foundation as our Rust-C++ Interop Engineer. Jon began laying the groundwork for the structure and vision of the Interop Initiative through important conversations with members of the Rust Project and existing players in Rust interoperability. An open listening session for open source, Rust Project, and Rust Foundation members is currently being discussed.

Upcoming Areas of Focus

A public-facing summary and outline of this initial research is being developed. This document will include a problem statement about the state of Rust-C++ interoperability, outline impacted groups and sectors, and list short and long term goals for the initiative.

Further Reading:

- https://foundation.rust-lang.org/news/google-contributes-1m-to-rust-foundation-to-support-c-rust-interop-initiative/
- https://foundation.rust-lang.org/news/welcoming-rust-c-interoperability-engineer-jon-bauman-to-the-rust-foundation-team/

Security Initiative

The Security Initiative is a fantastic example of what we can achieve with the Rust Foundation's expert stewardship. In September 2022, the Rust Foundation announced its commitment to fulfilling Rust language security commitments through a Security Initiative. Over the past 2+ years, we have used this program to work with the Rust Project to help empower contributors to participate in a secure and scalable manner, eliminate security burdens for Rust maintainers, and educate the public about security within the Rust ecosystem.

Support & Sponsorship

The Security Initiative would not exist without financial and in-kind support from several generous organizations. For the February-August period, we would particularly like to thank AWS and OpenSSF's Alpha-Omega Project for their financial contributions.

Progress

Project Collaboration

As described in our July 2023 and February 2024 Security Initiative Reports, the Security Initiative would not be possible without ongoing collaboration with the Rust Project, particularly via the Security Response Working Group and Secure Code Working Group. In June, Security Engineer Walter Pearce joined the RustSec Advisory team.

Public Key Infrastructure Model

In late February, Security Engineer Walter Pearce published the first in a series of requests-for-comment (RFCs) for a "Public Key Infrastructure" (PKI) model for the Rust language; including the design and implementation for a PKI CA and a resilient Quorum model for the project to implement, and next steps for signing across the Project. Crate and release signing will follow in a subsequent separate RFC. Project-wide discussion within this RFC is ongoing, but as of July, language updates suggested by members of the Project were nearly ready for posting.

Security Auditing

Supply Chain Security

Following the XZ backdoor vulnerability, the Security Initiative redoubled its focus on supply chain security. Adam has been working on provenance-tracking, verifying that a given crate is actually associated with the repository it claims to be. In addition to catching innocuous mistakes in the crate metadata, this will limit fraudulent crate creators trying to hide their malicious crates behind seemingly valid repos, when, in fact, the code for these crates is actually in unknown repos. Once caught, these crates can be quarantined and deleted using recently implemented admin functionality.

As of July, Adam has analyzed the top 5,000 crates by download count and has found no obvious red flags. Verification now infers commits based on tag data if the crate file was packaged without VCS information. This results in more mismatches, albeit innocuous ones that build artifacts, swap files, etc.

Walter has continued improving our ecosystem scanning capabilities to detect and analyze suspicious crates published on crates.io (Project Sandpit). This work has now been deployed in a production environment performing real-time analysis of crates as they are published. Painter has also been established in the same pipeline, allowing for real-time updating of the graph database of the ecosystem.

Crates.io Token Security

In June, Walter began using a local copy of crates.io to running gitleaks regex scripts in order to detect exposed credentials. Nine exposed tokens were found and removed through collaboration with Adam on a security incident process. The authors were notified.

Threat-Modeling

In March, the last of our four planned threat models under the Security Initiative was published:

- Crates ecosystem
- Rust Infrastructure
- crates.io
- Rust Project

Threat-modeling enables the Rust Foundation and Rust Project to better understand the risks identified by our security auditing. While developing all four threat models linked above, we have consulted with the Rust Project's crates.io Team, Infrastructure Team, Security Response Working Group, Secure Code Working Group, and specific external stakeholders. We appreciate their collaboration.

Tooling & Research

Painter

Painter is the first open source tool formally released by the Rust Foundation. It is an implementation of methodologies to build a graph database of dependencies and invocations between all crates within the crates.io ecosystem.

Typomania

Typomania is the Rust Foundation's second formal open source project. It ports the excellent typogard (originally by a team led by Matthew Taylor at the University of Kansas) to Rust. Rather than being hard coded to a specific registry, the Typomania crate provides the same set of primitives that typogard uses in order to detect potential typosquatting as a reusable library that can be adapted to any registry by implementing the traits provided in this crate.

A substantial Painter code push was made by Walter in May. Those running Painter will now be able to obtain unsafe statistics, better call graph pruning, FFI boundary mapping and support for the latest version of Rust. In June, Walter added a RustSec CVE categorization to Painter, allowing it to output which CVE types are the most prevalent in the Rust community and to aid future vulnerability response efforts.

Walter is leading the porting of our Security Initiative infrastructure from local machines into the cloud, specifically Google Cloud Platform (GCP). There is now an active crates mirror where security scans are regularly run with tools like Painter and Typomania. The results of these scans are stored in a Neo4j database and Elasticsearch can be used to find specific results.

In May, there were substantial engineering efforts on Painter and crates.io and significant work to prevent fraudulent/malicious crates. An interesting blog post on the state of unsafe in the Rust ecosystem was also published.

Upcoming Areas of Focus

- Complete crates.io provenance scans, and identify the best way to surface this information to all Rustaceans.
- Investigate the best way to federate distributed code review-publishing (using tooling such as cargo-vet).
- Further investment in crates.io admin functionality.
- Investigate the possibility of replacing build.rs scripts with a unified framework, increasing standardization and making them easier to flag for review (in the same manner as many organizations currently review all unsafe blocks in dependency graphs).
- Begin the RFC process for a second pass at crate signing & PKI utilizing The Update Framework (aka TUF).
- Share real-time results of crate security scanning for broader Rust Project participation.
- Investigate integration of ecosystem graph database (Painter) into other projects in the ecosystem (cargo-audit, cargo-vet, RustSec, CVEs).
- Investigate changes and updates to crater for integration into Painter and upstreaming of improvements.
- Investigate key data points from the OpenSSF Package Analysis Project for integration into our pipelines.

Infrastructure Support

One of the key pillars of the Rust Foundation's mission is to provide robust infrastructure support to the Rust Project and community, particularly as the popularity and use of Rust continues to rise. In doing so, we help ensure that the infrastructure of Rust and related projects is reliable, efficient, productive, and secure.

This mission-critical stream of work is made possible through the dedication of our team's infrastructure engineers, their collaboration with members of the Rust Project, and generous infrastructure donations from the organizations listed below.

Support & Sponsorship

The following organizations have generously donated in-kind infrastructure services to the Rust Foundation. Our Technology Team is grateful for the benefits they provide to the Foundation and Rust ecosystem:

Progress

In July, we announced the arrival of Marco Ieni – our second Infrastructure Engineer, reporting to Jan David. Marco is responsible for the operations of all current and future infrastructure, managing existing tools and new developments while working directly with the Rust Project Infrastructure team in their daily activities. Marco has already accomplished some great work including setting up a CDN for crates.io's download archives. In addition to many other benefits, this helps the crates.io team make download statistics publicly available.

As mentioned in the Security Initiative Section, we published our Rust Infrastructure threat model earlier this year. It lists Rust Infrastructure actors, assets, threat scenarios and security recommendations. This work will allow us to help develop more secure infrastructure systems and reduce Rust infrastructure risk exposure.

Thank you to Platinum Member Microsoft for extending the expiration date of our Microsoft Azure credits for Rust to May 31, 2025 and for granting a generous sum of additional credits, which we expect will sustain us through this period based on the Rust Project's historical usage trends.

Finally, during the period of this report, Fastly generously provided the Rust Project bandwidth and additional services for Rust releases and crates through their Fast Forward program. In conjunction with AWS, this agreement provides the Foundation and Project peace of mind knowing that we have stable bandwidth infrastructure.

The Rust Foundation is grateful to our infrastructure donors for providing such an important service to the Rust language!

Upcoming Areas of Focus

- **Catalog and patch all servers:** The infrastructure team operates a few physical and virtual servers. These need to be cataloged, patched, and potentially upgraded to establish a process that can then be automated.

- **Back up Rust releases and crates:** Work to set up automated out-of-band backups of Rust releases and crates.

Further Reading: https://foundation.rust-lang.org/news/welcoming-infrastructure-engineer-marco-ieni-to-the-rust-foundation-team/

Jan David Nose's talk on Rust infrastructure at Rust Nation UK 2024: https://www.youtube.com/watch?v=GnLZMJ2r7sk

Rust Language Specification

In 2022, the Rust Project elected to start working on an official “specification” for the Rust language. As stated in a 2023 blog for Inside Rust, “An ideal specification (1.) defines prescriptive bounds on the semantics of a given Rust program for current and future Rust versions, and (2.) provides descriptive details of the semantics that are specific to the Rust version coupled with that instance of the specification.” The purpose of the forthcoming Rust language specification is to provide an authoritative resource to define valid Rust programs and how they should behave. Joel is currently serving as editor of the specification.

The specification has taken a considerable amount of time to get off the ground and the initial phases have lasted longer than expected. However, we are pleased to share that a clearer plan has been established and the Rust Reference is being converted for use as a proper specification. The Foundation has hired a specification consultant to help with that work and expectations are for substantial progress to be made by the end of the year.

You can follow public Specification Team discussions from the February-August period on Zulip: https://rust-lang.zulipchat.com/#narrow/stream/t-spec

Further Reading: https://github.com/rust-lang/team/pull/1458

https://rust-lang.github.io/rfcs/3355-rust-spec.html

https://blog.rust-lang.org/inside-rust/2023/11/15/spec-vision.html

Please join us in congratulating the Rust Foundation Technology Team, collaborators within the Rust Project, and organizational supporters – their hard work over the past six months allowed us to make significant progress across many packages of technical work. Thank you to our generous corporate donors for equipping the Rust Foundation with the funding and tooling needed to carry out our mission.

Finally, thank you for reading the Rust Foundation's first report covering recent focus areas of the entire tech team. We look forward to issuing future installments detailing these and other technology priorities in the coming months.

Interested in supporting our work?

The Rust Foundation is an independent nonprofit organization dedicated to stewarding the Rust programming language and community and making meaningful investments in the Rust Project. Through our work, we help ensure the safety, security, performance, growth, and global availability of Rust.

Interested in supporting one of the technical programs you read about in this report through membership, a donation, or collaboration? Reach out to us via email at contact@rustfoundation.org.

Rust
Foundation

rustfoundation.org