# Introducing Rusted Firmware-A

a.k.a. RF-A

Sandrine Afsa
18th September 2025

# Agenda

Introducing the RF-A & Arm Firmware Crates projects

V0.1 Contents

Roadmap

Request for feedback

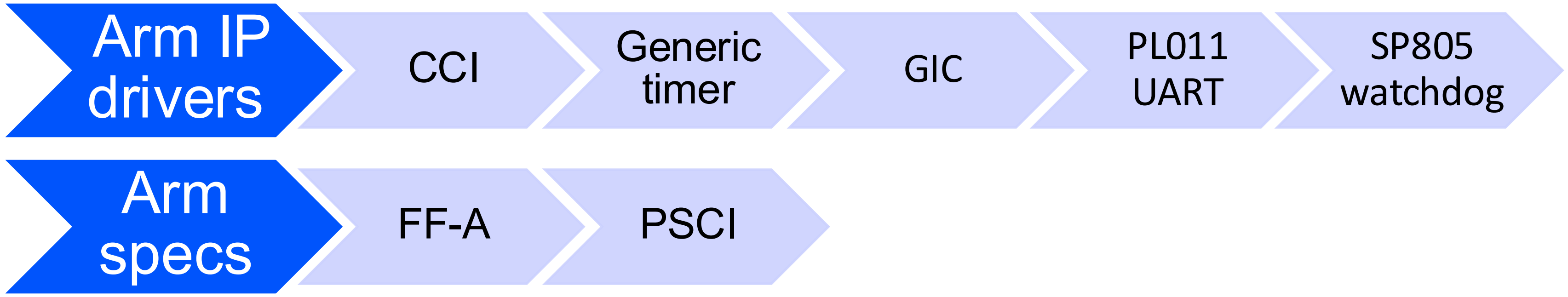
# What is Rusted Firmware-A? (a.k.a. RF-A)

- An experimental Rust implementation of EL3 runtime firmware (BL31).
- Developed collaboratively by Arm and Google.
- v0.1 version announced on 18th August.
- Open-source, governed by tf.org, BSD-3 license.
- Influenced by TF-A implementation but not the same!
  - Fresh design and implementation in idiomatic Rust.
  - Better modularity (see next slide).
  - Built for current and future hardware, drops legacy and underused features.
    - Assumptions: AArch64 only, GICv3+, DSU, hardware-assisted coherency, ...

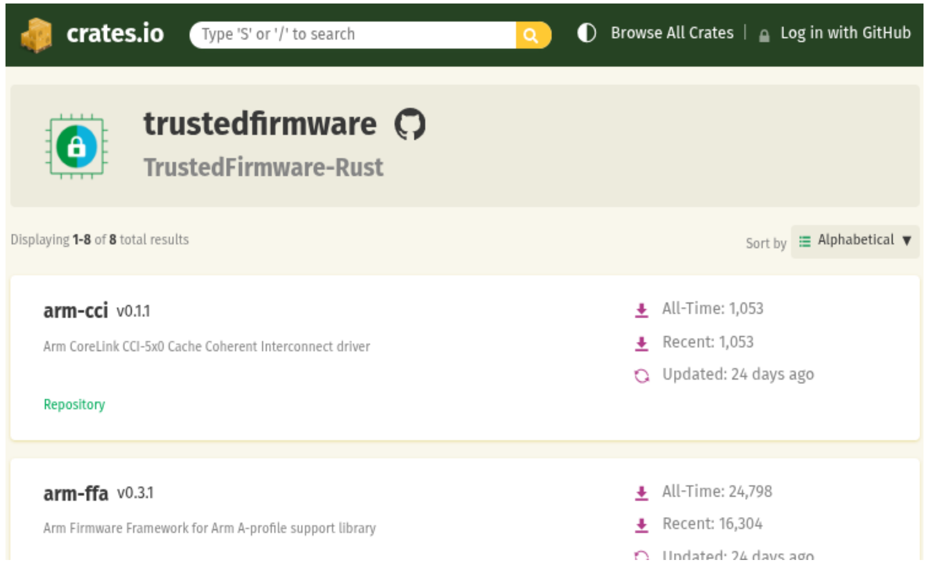
- Why **`Rust`**?
  - High-level language expressiveness and abstractions (type system, error handling, …).
  - Easy build system integration and dependency (crate) management.
  - Ensure memory safety.
  - Catch more bugs at compile time (borrow-checker).
  - Align with modern security guidance from regulators and security standards bodies.

# Arm Firmware Crates

- A collection of reusable crates for firmware development on Arm systems.



- More to come...
- Look them up on [crates.io](https://crates.io)!



trustedfirmware
TrustedFirmware-Rust

Displaying 1-8 of 8 total results

Sort by Alphabetical

arm-cci v0.1.1
Arm CoreLink CCI-5x0 Cache Coherent Interconnect driver
Repository
All-Time: 1,053
Recent: 1,053
Updated: 24 days ago

arm-ffa v0.3.1
Arm Firmware Framework for Arm A-profile support library
All-Time: 24,798
Recent: 16,304
Updated: 24 days ago

## V0.1 Contents

### Supported platforms

- Arm Base FVP.
- QEMU.

### Features (partial implementation)

- System initialization: UART, MMU, GIC, ...
- Boot process coordination: Secure World, Normal World.
- SMC runtime services framework.
- Manage transitions between secure and normal worlds.
- FF-A v1.2 compliant SPMD.
- Power management (PSCI).

### Tooling

- Dependency auditing through `cargo vet`.
- Code linter through `cargo clippy`.
- A test framework in Rust.
- OpenCI integration: unittests, build tests, boot tests (FVP only).

# What's Next?

- Full SMP support.
- Arm architectural extensions.
  - Armv9.2 Realm Management Extension (RME) support.
  - Armv8.3 Pointer Authentication (PAC) + Armv8.5 Branch Target Identification (BTI).
- Running TF-A test suite (TFTF).
- Full software stack demonstration (Hafnium, secure partitions, Linux).
- Firmware handoff support for easy, standardized integration with previous firmware.
- Hardware errata management framework.
- More tooling: Code coverage, memory consumption monitoring.

- And much more!
- See https://github.com/RustedFirmware-A/rusted-firmware-a/issues

## What is the Impact on TF-A?

**None at this stage!**

- RF-A is a prototype right now.
- It's a long way ahead before it will be production ready!
- TF-A will continue to be supported and maintained, including LTS versions.

- For future products, we encourage you to think about RF-A adoption.

## Feedback

- We welcome feedback!
  - Please reach out through the project's mailing list or Discord channel #rusted-firmware-a.
  - Report issues and feature wishes onto the issues tracker.

- Right now, patch contributions from Arm and Google are prioritized.
  - Other contributions will be taken on a best-effort basis.
  - This will change as the project matures.

# References

| | |
|---|---|
| Project announcement blog post | |
| Source code | RF-A git repository (v0.1.0 tag)<br>Arm Firmware Crates git repositories |
| RF-A OpenCI | |
| RF-A issues tracker | Crates issues tracker at https://github.com/ArmFirmwareCrates/***crate-name***/issues |
| Communication channels for developers | Mailing list<br>Tf.org Discord channel #rusted-firmware-a |
| Google's Rust training course | Useful to get up to speed with the Rust language |

Merci
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Thank You
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు
Köszönöm