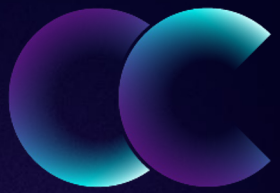
Confidential Compute Working Group (WG) Kickoff



CORECOLLECTIVE

Where the Arm software ecosystem connects

8th June 2026

Confidential Compute WG Agenda 8th June 2026

CORECOLLECTIVE
Where the Arm software ecosystem connects

- Attestation Endorsement API
- Arm CCA SW Updates
- Linaro CCA QEMU Support Update
- AOB



Attestation Endorsement API

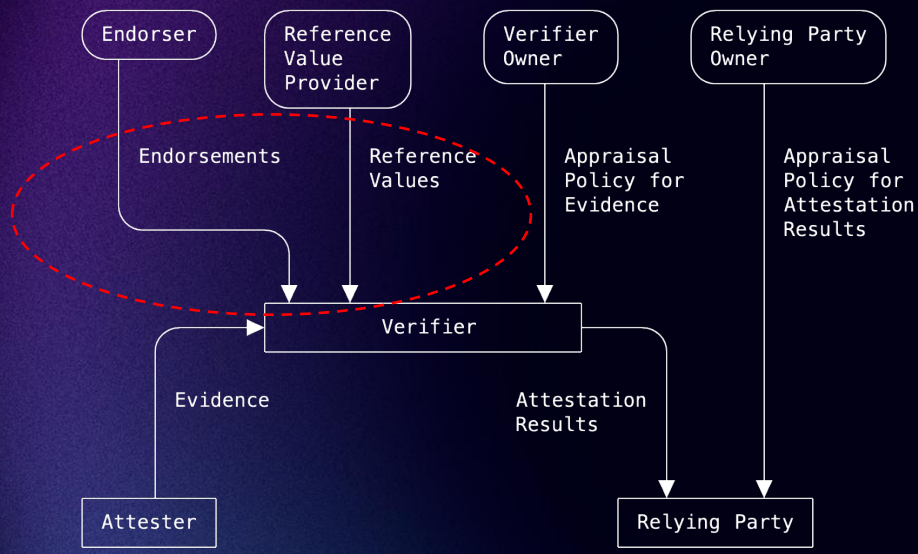
Collaboration in Standards and Open Source

Paul Howard, Senior Principal Software Architect, **Arm**

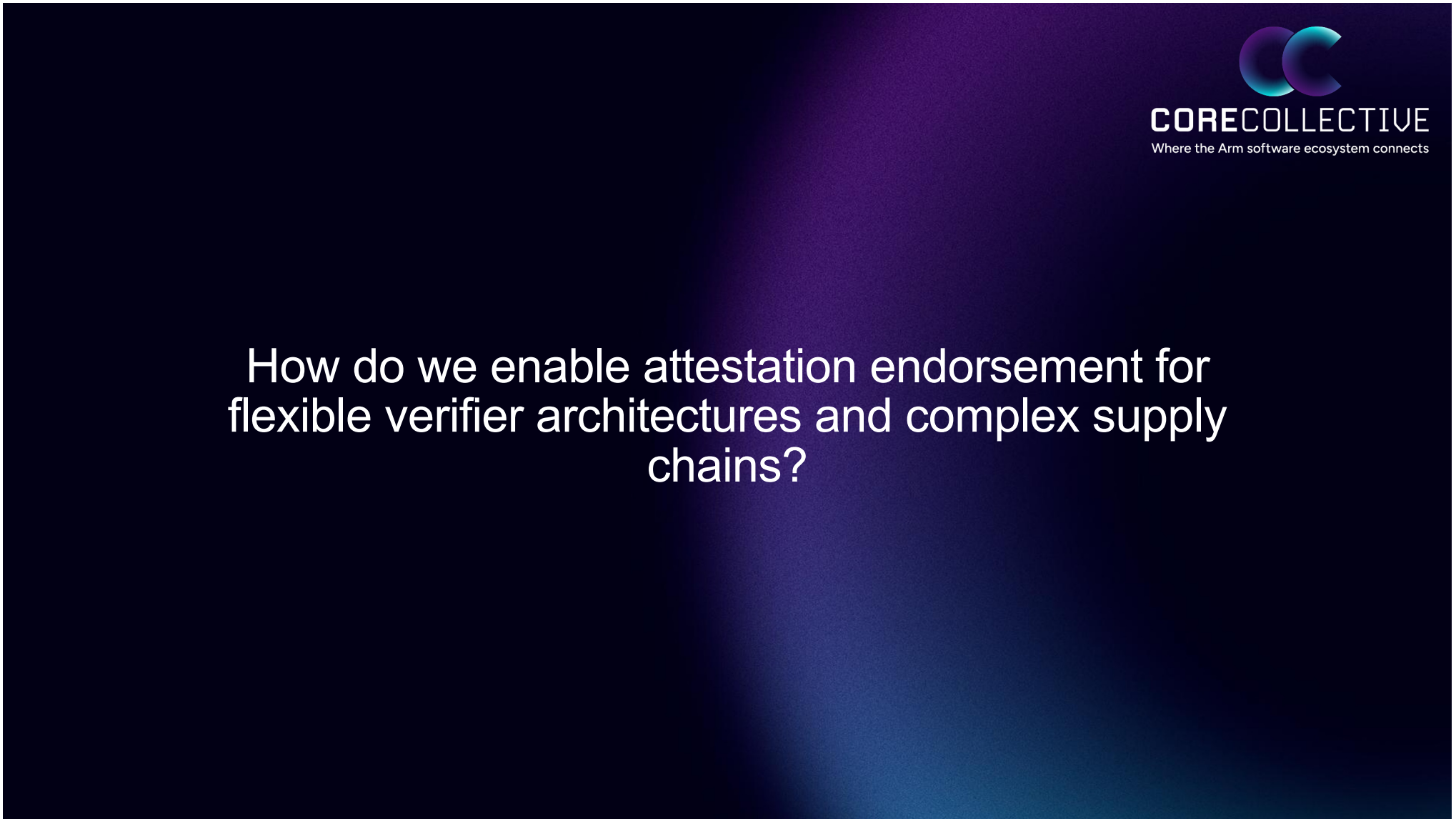
Context

- **Attestation:** the process of establishing trust in a computing device, platform, or environment, through the evaluation of cryptographically-protected evidence
- **Verification:** the process of examining the evidence and comparing it against the expected norms of a trusted supply chain
- **Endorsement:** the process by which the supply chain can publish authoritative statements that describe a trustworthy system

CORECOLLECTIVE
Where the Arm software ecosystem connects

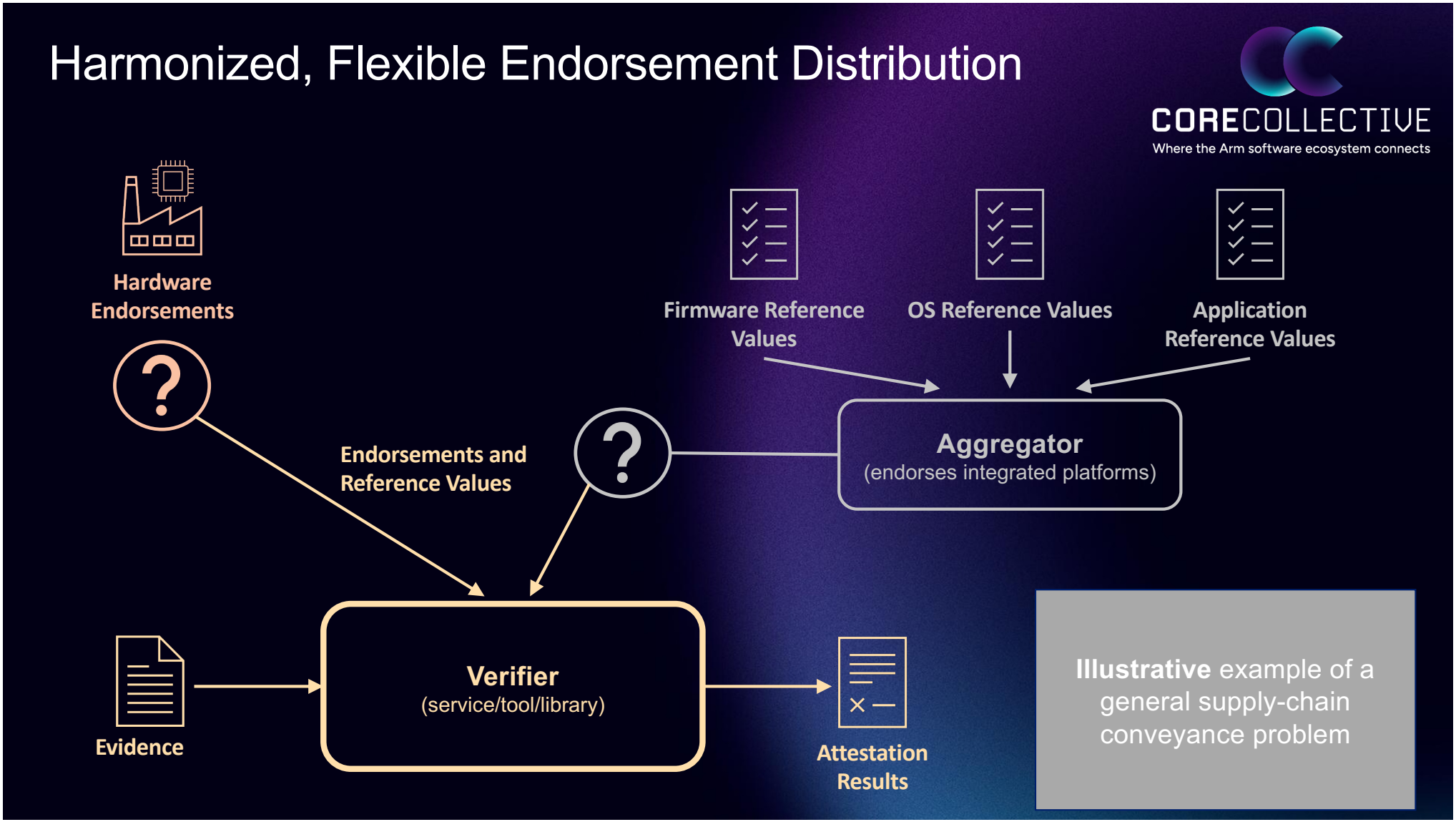


IETF RATS Architecture (RFC9334)


CORECOLLECTIVE
Where the Arm software ecosystem connects
How do we enable attestation endorsement for flexible verifier architectures and complex supply chains?

Harmonized, Flexible Endorsement Distribution

CORECOLLECTIVE
Where the Arm software ecosystem connects



A Common Interface: CoSERV



- **CoSERV** = **Co**ncise **S**elector for **E**ndorsements and **R**eference **V**alues
- Active (adopted) IETF draft within the RATS Working Group
- Closely related to CoRIM; directly inherits CoRIM data model
- Fulfils a function that CoRIM lacks: the ability to query for the required RIMs
- Encoded in CBOR for efficiency
- Minimal query language, specialised for endorsement and reference value artifacts
- Could operate over different transports
- HTTP REST API included in the specification
  - Simple request-response protocol
  - Cache-friendly: CoSERV queries are stable identifiers

A CoSERV Transaction

CORECOLLECTIVE
Where the Arm software ecosystem connects



Producer (e.g. Endorser or Reference Value Provider)

CoSERV Query

- Selects for **trust anchors**, **endorsed values** or **reference values**
- Selects by **environments** (instance/group/class) or by **RIM identifiers**
- Can request **aggregated** results
- Can include **stateful environments**
- Encodes deterministically: **a query is a stable identifier**

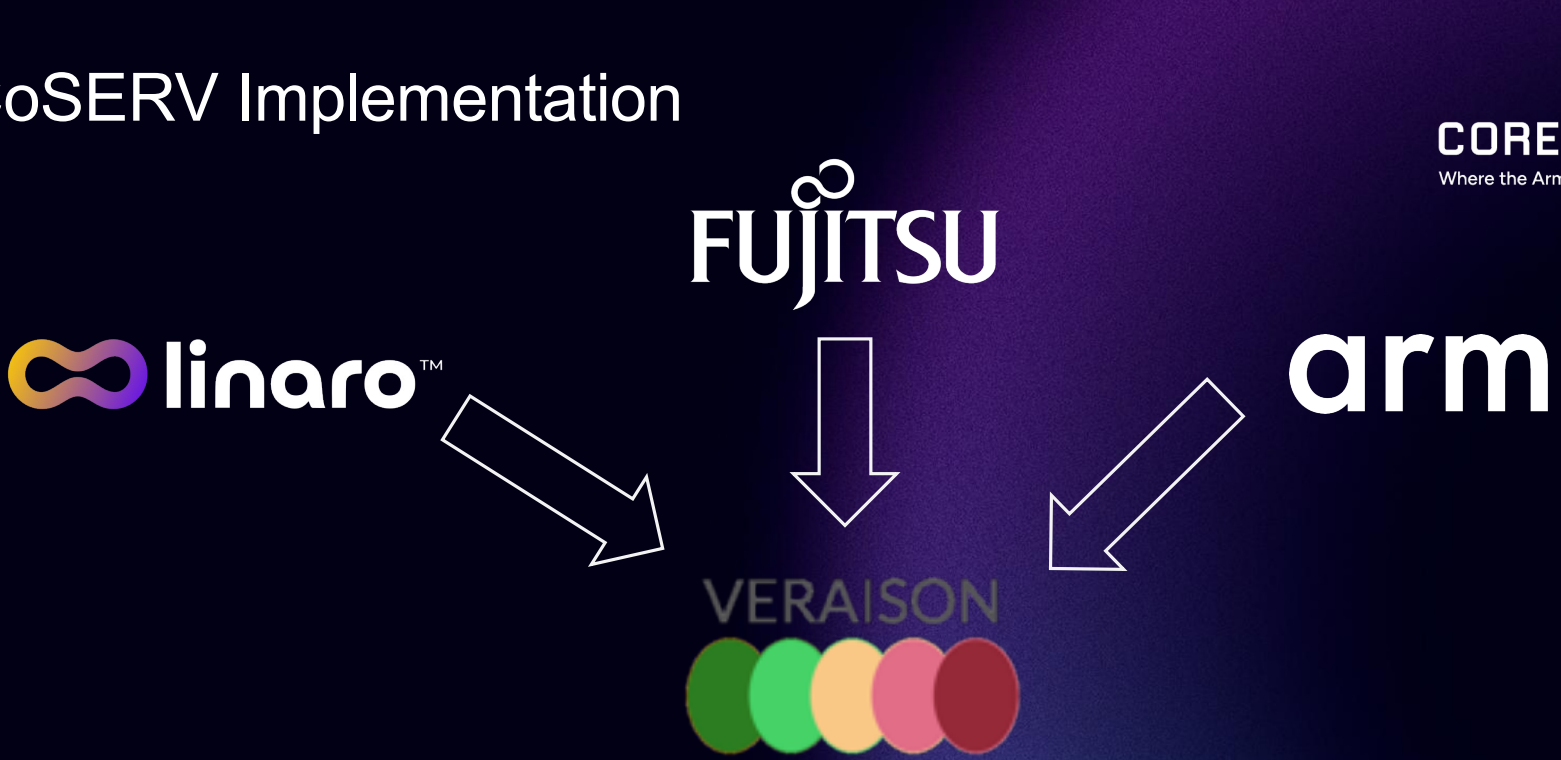
CoSERV Result Set

- Optionally **signed** by the producer
- Embeds the original query (for integrity checks)
- Conveys **artifacts** to satisfy the query
- **Source artifacts** can be embedded verbatim along with original signatures
- **Collected artifacts** can provide a convenient aggregated view over the supply chain

Consumer (e.g. Attestation Verifier)



CoSERV Implementation



The **Veraison** project is the ideal place to show how the CoSERV standard can be implemented and used in practice

CoSERV Implementation: Libraries





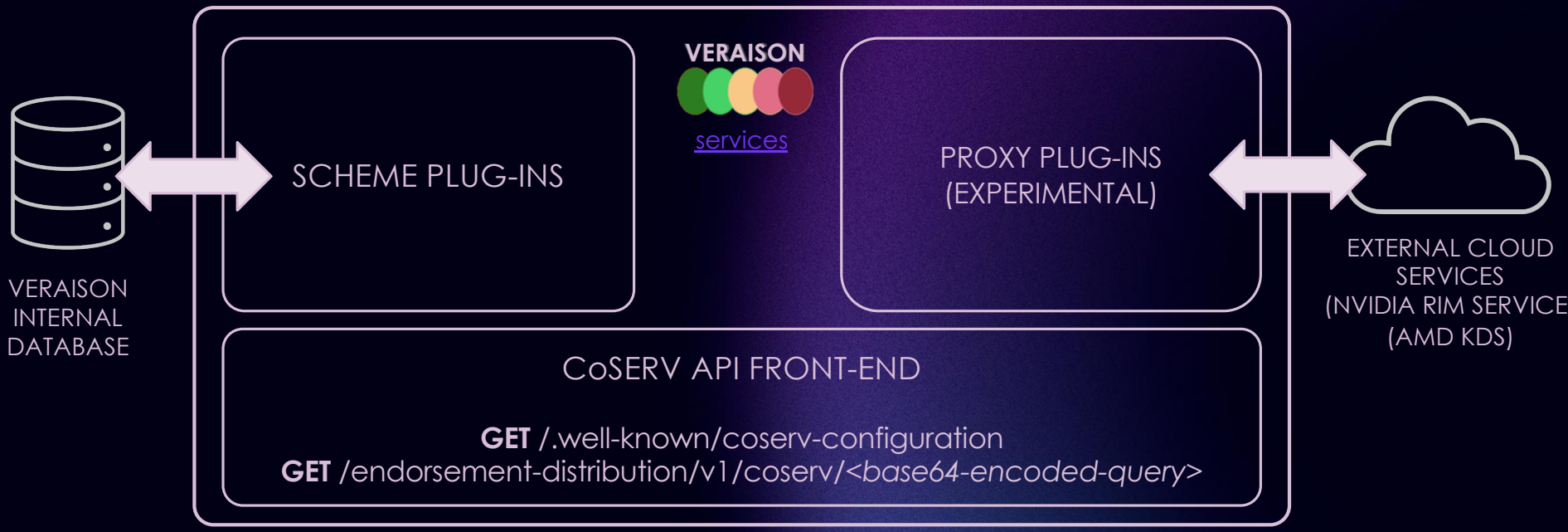
- [x] Data model and CBOR serialization/de-serialization, plus COSE sign/verify support: https://github.com/veraison/corim

- [x] Data model and CBOR serialization/de-serialization, plus COSE sign/verify support: https://github.com/veraison/coserv-rs

- [x] Client-side support for running and caching CoSERV queries: https://github.com/veraison/rust-apiclient

CoSERV Implementation: Services

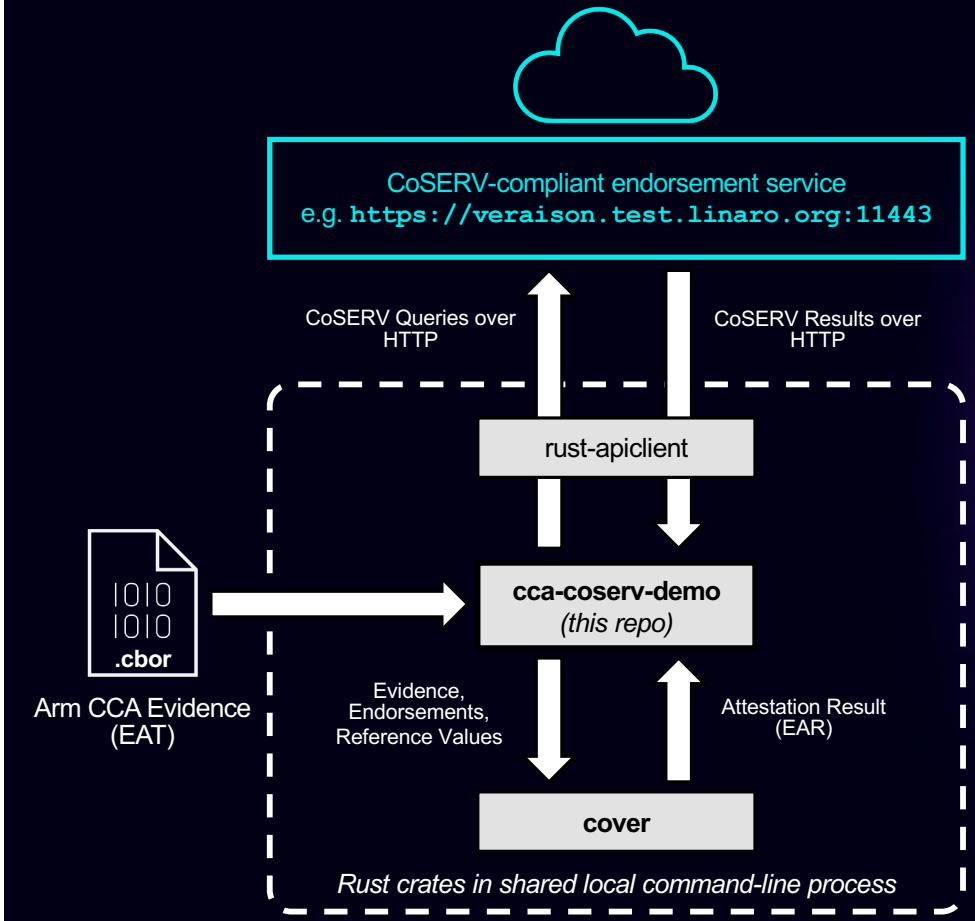
CORECOLLECTIVE
Where the Arm software ecosystem connects



Example live deployment available for Arm CCA pre-silicon environments:
`https://veraison.test.linaro.org:11443`

End-to-End Demonstrator (Arm CCA)

CORECOLLECTIVE
Where the Arm software ecosystem connects



- Fully encapsulated demo at: https://github.com/veraison/cca-coserv-demo
- Presented at IETF-125 Hackathon
- Supports verification and appraisal of Arm CCA evidence, but would extend to other schemes
- Showcases server-side support in Veraison services
- Showcases Rust client-side support, including caching (can be run "unplugged" while cached results are valid)
- Uses the new **CoRIM verifier** from the Veraison project: https://github.com/veraison/cover
- Demonstrates CoRIM/CoSERV compatibility and combined value
- All functionality in mainline branches

Thanks



Thomas Fossati and the team at **Linaro**



Shefali Kamal, **Dhanus Lal** and the team at **Fujitsu**



Sergei Trofimov, **Yogesh Deshpande** and the team at **Arm**



Carl Wallace, **Giri Mandyam**, **Henk Birkholz**, **Ding Ma**

CORECOLLECTIVE
Where the Arm software ecosystem connects

Resources



- **Draft**: https://datatracker.ietf.org/doc/draft-ietf-rats-coserv/06/



- **Draft repo** (for issues and PRs): https://github.com/ietf-rats-wg/draft-ietf-rats-coserv



- **End-to-end demo**: https://github.com/veraison/cca-coserv-demo

CORECOLLECTIVE
Where the Arm software ecosystem connects


CORECOLLECTIVE
Where the Arm software ecosystem connects
CCA SW Updates

CCA software 1.0 status

CORECOLLECTIVE
Where the Arm software ecosystem connects

- RMM spec 1.0-REL0 released Sep 2024
  - Enables ”CPU-only” realms
  - Minimum Viable Product (MVP) but missing some features needed by lead partners
- TF-A / TF-RMM (tag rmm-spec-v1.0-rel0) upstream since Oct 2024
- Guest Linux patches merged to v6.13 in Jan 2025
  - Patches to support guest on host with large page support on list since Dec 2025
- Host KVM patches have been regularly pushed to the list for more than a year
  - Last 1.0-based v12 Host KVM patches on list Dec 2025 (addressed maintainer UABI feedback)
- PR to merge EDK2 realm guest firmware patches raised Mar 2026
  - Now split into several smaller PRs to facilitate upstreaming (see top-level PR for status)

CCA software 2.0 status



- Baseline support:
  - Initial 2.0 TF-RMM support merged March 2026 and released in v0.9 May 2026
  - 2.0 v14 Host KVM patches on list May 2026 (depends on PoC TF-RMM branch)
- Foundation DA:
  - 2.0-BET1 stack
    - TF-RMM patches merged May 2026; use this branch to be sure of a working stack; integrated build/test instructions here
    - v4 Guest Linux, Host KVM, IDE setup (Host), IOMMU-FD UABI for DA (Host) on list April 2026
  - 1.1-ALP17 stack
    - (Superseded) TF-RMM patches merged Dec 2025; use this branch to be sure of a working stack; integrated build/test instructions here
    - v3 Guest Linux, Host KVM, IDE setup (Host), IOMMU-FD UABI for DA (Host) on list March 2026
- Planes:
  - RMM spec 1.1 ALP14 TF-RMM patches merged Sept 2025
  - Matching v1 RFC Host KVM also available
  - Will need to be reworked/rebased on 2.0 base stack
- MEC:
  - TF-A / TF-RMM patches merged Sept 2025
  - Host KVM, kvmtool prototype branches available since Jan 2025
  - Obsoleted by 2.0 spec (VMID/MECID allocation will now be handled within RMM) – TF-A / TF-RMM will need to be reworked
- LFA:
  - LFA of TF-RMM: TF-A / TF-RMM patches merged incrementally between June 2025 and Jan 2026
    - v2 Linux patches patches available since Mar 2026
  - LFA of BL31: TF-A public branch available since Mar 2026 (see here for usage info)
  - LFA of Secure Partitions (SPs): TF-A patches merged Mar 2026 (see here for usage info)
- Boot sync public branches available since Sept 2025
  - Enabling "encrypted disk boot" and "measured boot" use-cases

CCA Software Deliverables ROADMAP



Done

- **RMM 1.0** | TF-RMM (UQ) | Guest (UQ)
- **DA Foundation** | RMM 1.1 ALP17 | TF-RMM (UQ) & Kernel (PB)
- **Planes** | RMM 1.1 ALP14 | TF-RMM (UQ) & KVM Host (PB) | P0-Lite PoC
- **MEC** | RMM 1.1 | TF-RMM (UQ) & KVM Host (PB)
- **LFA** | TF-RMM (UQ) | Kernel Host (PB) | SP Hafnium (UQ)
- **RMM 2.0** |Initial Branch | TF-RMM & Kernel Host
- **EDK2** Support (PB)
- **Large page support** | Guest on a host (RFC)
- **FIRME TF-A & TF-RMM** | Static L1 GPT | GPI_set Single Page Support

Q2 2026

- **RMM 1.0** | EDK2 Guest (UQ)
- **RMM 2.0 Base Support**| TF-RMM & Kernel Host (PB)
- **RMM 2.0 DA Foundation** | Initial Branch | Kernel Host
- **Planes** | P0-Lite PoC
- **LFA** | BL31 LFA Itself (UQ)
- **Large page support** | Guest on a host (UQ)

Q3 2026

- **RMM 2.0 Base Support** | TF-RMM (PB)
- **RMM 2.0 DA Foundation** | Kernel Host (PB) | Kernel Guest (UQ) | TF-RMM PSMMU (UQ)
- **RMM 2.0 MEC** | TF-RMM (UQ) | Kernel Host Initial Alignment
- **RMM 2.0 Planes** | TF-RMM (UQ) | Kernel Host Initial Alignment | P0-Lite PoC
- **LFA** | Kernel Support (UQ)
- **EDK2 Boot Sync** | Encrypted Disk Boot (UQ)
- **FIRME TF-A & TF-RMM (PB)**| Dynamic GPT | GPI_Set | Long Running MM | IDE | MEC

Q4 2026

- **RMM 2.0 Base Support** | TF-RMM (UQ) | Kernel Host (UQ)
- **RMM 2.0 DA Foundation** | Kernel Host (PB) | PSMMU (PoL)
- **RMM 2.0 MEC** | Kernel Host Initial Alignment
- **RMM 2.0 Planes** | Kernel Host (PB) | P0-Lite PoC
- **EDK2 Boot Sync** | Encrypted Disk Boot (UQ)
- **FIRME TF-A & TF-RMM (PB)**| Dynamic GPT | GPI_Set | Long Running MM | IDE | MEC

Q1 2027

- **RMM 2.0 DA Foundation** | Kernel Host (UQ)
- **RMM 2.0 Planes** | Kernel Host (UQ)

Future 2027+

- **RMM 2.0 DA Foundation** | Kernel Host (UQ)
- **RMM 2.0 Planes** | Kernel Host (UQ)
- **LFA** | Full System Update
- Firmware Activity Log
- **Large page support** | TF-RMM
- **RMM 2.x Coherent DA** | P2P | RCIeP | PRI | S1 SMMU
- RMM 2.x CMEM CXL
- RMM 2.x Live Migration

Roadmap updated: 27 May 2026

- UQ - Estimate for Upstream Queue
- PB - Refers to Public Branch
- PoL – Patches on List
- RFC – Patches Ready for Comment


CORECOLLECTIVE
Where the Arm software ecosystem connects
Linaro CCA QEMU Updates

Any other business?

CORECOLLECTIVE
Where the Arm software ecosystem connects

- Agenda for next meeting
  - CCA Learning Path Roadmap
  - Any other topics?

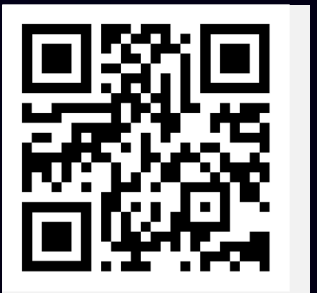

CORECOLLECTIVE
Where the Arm software ecosystem connects
Appendix

Joining the Confidential Compute Working Group

CORECOLLECTIVE
Where the Arm software ecosystem connects

- Join CoreCollective
  - Company: sign and submit Membership Agreement https://corecollective.dev/join/

- After your company has signed the Membership Agreement, please email confidential-compute+subscribe@corecollective.dev from your company email domain
  - Or ”Ask to Join” group from https://groups.google.com/u/0/a/corecollective.dev/g/confidential-compute



CoreCollective.dev

Note: Subscriptions are restricted to company email domains of current members

CoreCollective Terminology



- **Working Group (WG):** A Working Group (WG) is a free-to-join, technology-focused forum for member companies. Its purpose is to target, evangelise, and advance Arm technology within domain-specific verticals (such as Automotive, AI, or Edge Compute) in the Open Source community. Working Groups are typically either standards-based (with no engineering involved) or Initiative-based.
- **WG Initiative:** A WG Initiative is a long-term engineering development effort contained within and guided by a Working Group. Each WG Initiative is funded by Sponsorship from one or more member companies and is executed by an Open Source Solution Provider (OSSP). A WG Initiative may comprise multiple workstreams, but its higher-level focus will reflect a sustained level of effort in Open Source development.
- **Open Source Solution Provider (OSSP):** An Open Source Solution Provider (OSSP) is a specialised engineering firm that bridges commercial requirements and the Open Source ecosystem. OSSPs provide expert development, integration, and long-term maintenance of software stacks (such as the Linux kernel or AI frameworks). They focus on "upstream" contributions to ensure hardware enablement and long-term architectural sustainability for WG Initiatives.
- **Workstream:** a component of work within a WG Initiative. Example: The Virtualisation WG Initiative may include multiple workstreams, such as QEMU, VirtIO, and even a workstream to fund a maintainer for an open-source project.

CoreCollective Terminology continued



- **WG Chair:** Each Working Group is led by a WG Chair, an influential technical expert. The WG Chair is responsible for leading members toward a solution-focused WG Initiative that addresses ecosystem gaps of interest to the broader community. The WG Chair is supported by a CoreCollective Community Manager to guide them efficiently through CoreCollective processes, leading to the deployment of member-desired Initiatives.
- **Community Manager:** For each Working Group, a Community Manager will be assigned to facilitate all WG administrative tasks as defined in the Working Group Life Cycle of the Charter. They will support the WG Chair in driving Initiatives, scheduling/funding support, reporting, and OSSP vendor selection.
- **Technical Advisory Committee (TAC):** The Technical Advisory Committee (TAC) oversees the Working Group. It approves the addition, removal and advancement of Working Groups through the WG Life Cycle.
- **TAC Chair:** The Technical Advisory Committee (TAC) Chair is the individual ultimately responsible for all CoreCollective Collaborative efforts. This person also evangelises CoreCollective membership and is key to determining when to initiate new WGs.