

CoSERV

CONCISE SELECTOR FOR ENDORSEMENTS AND REFERENCE VALUES

IETF-125 DRAFT AND RUNNING CODE UPDATES

PAUL HOWARD

CoSERV Recap

A common query language for RATS artifacts, based on CBOR and CoRIM



Trust Anchors



Reference Values
Endorsed Values

Supporting flexible interactions and transports



HTTP REST

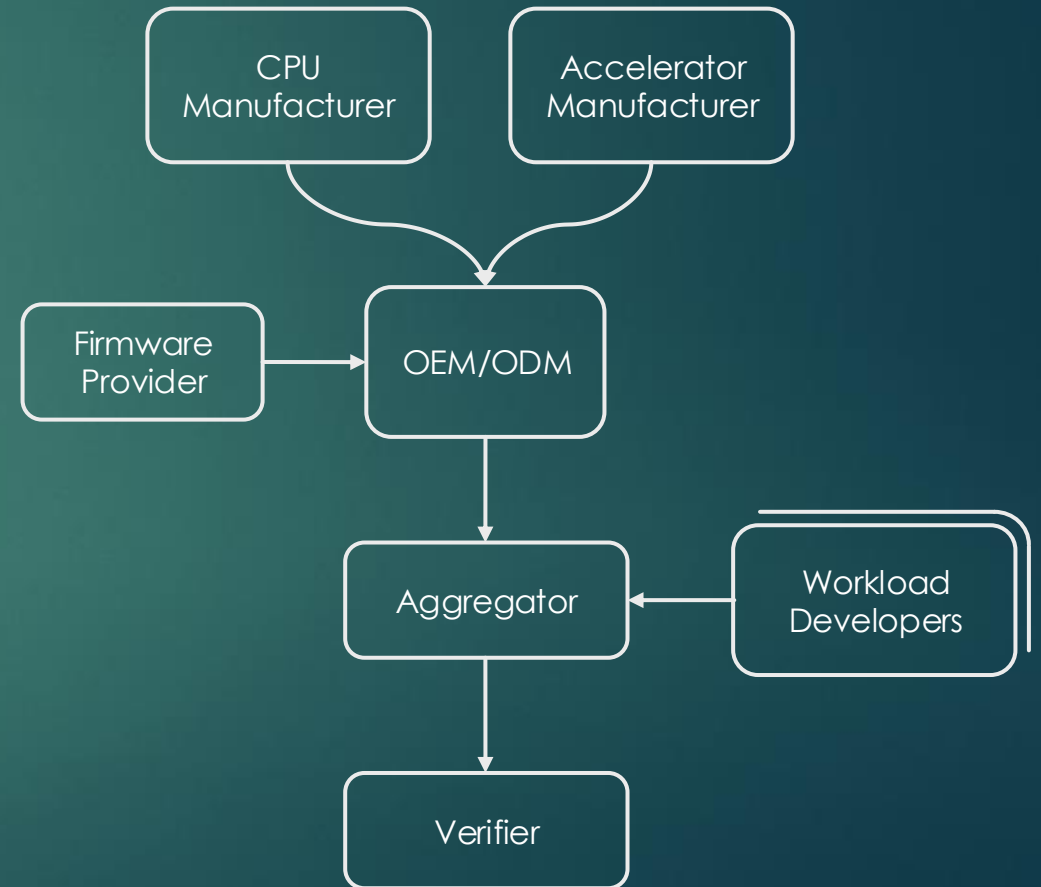


CoAP



One-shot or pub/sub

Supporting real-world supply chain complexity



I-D Updates and Roadmap

Updates Since IETF-124

- ▶ **Data Model Change**
 - ▶ Timestamp field removed from the Query object
 - ▶ Enables the intended caching model
- ▶ **Incorporating WG Feedback**
 - ▶ Updates to Security and Privacy Considerations for aggregators
 - ▶ New (informative) appendix on discovery of CoSERV endpoints
 - ▶ Addressed points from Carl Wallace's detailed review (thank you, Carl)
 - ▶ Closed 16 GitHub issues

Roadmap

- ▶ **Incorporating WG Feedback**
 - ▶ More reviews invited
 - ▶ Continue to address open GitHub issues
- ▶ **Identifying Gaps**
 - ▶ Prototyping-informed fixes and enhancements
 - ▶ Use case exploration (e.g. CoCo project)
 - ▶ More artifact types (e.g. topological info)?
- ▶ **WGLC Readiness**
 - ▶ CoRIM data model stabilization
 - ▶ COTS integration
 - ▶ Suggesting **HTTP directorate** review after next editorial pass

Running Code Updates: ACK



Thomas Fossati and the team at **Linaro**



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

Running Code Updates: Libraries

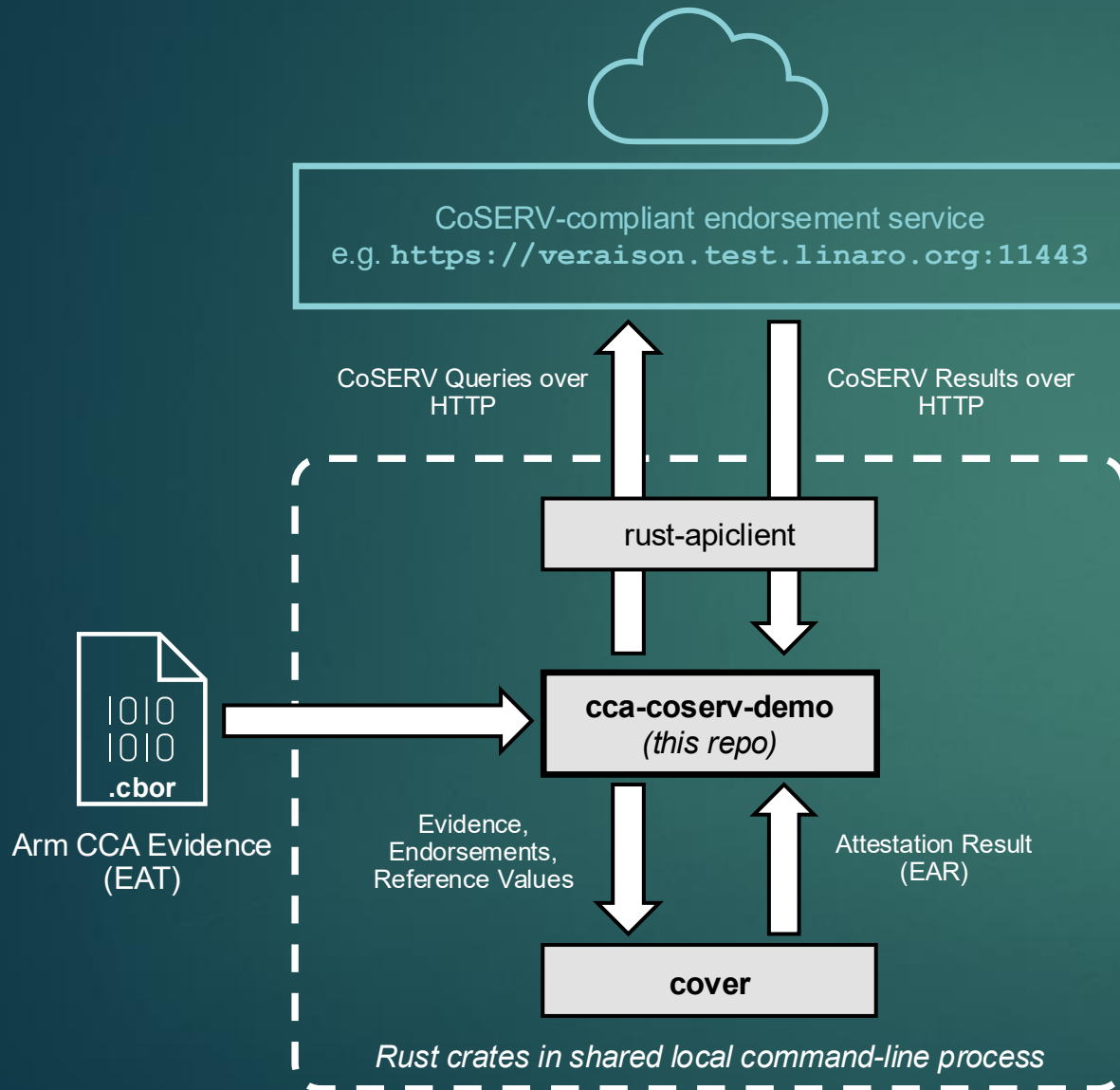


- ✓ Data model and CBOR serialization/de-serialization, plus COSE sign/verify support:
<https://github.com/veraison/corim>

- ✓ Data model and CBOR serialization/de-serialization, plus COSE sign/verify support:
<https://github.com/veraison/coserv-rs>

- ✓ Client-side support for running and caching CoSERV queries:
<https://github.com/veraison/rust-apiclient>

Running Code Updates: End-to-End



- ▶ 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

Resources



- ▶ **Draft:** <https://datatracker.ietf.org/doc/draft-ietf-rats-coserv/05/>



- ▶ **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>