linaro™

Arm Solutions at Lightspeed

Supporting Confidential Computing on Arm

With Open Source Software

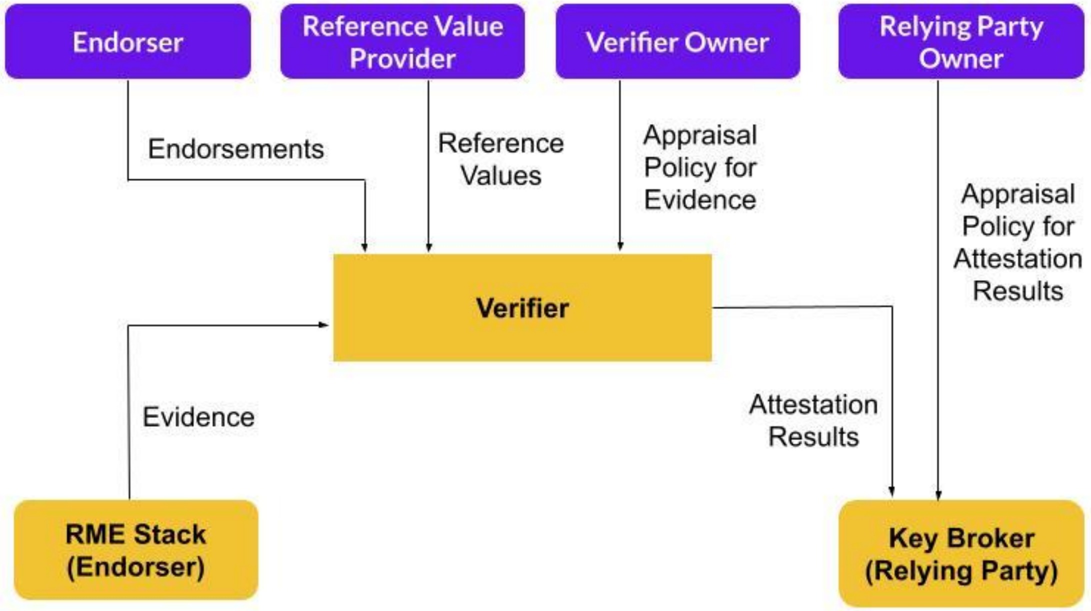
CCA End-to-End Scenario?

The content herein is a join effort between Linaro, Arm and the Linaro Data Center Group members

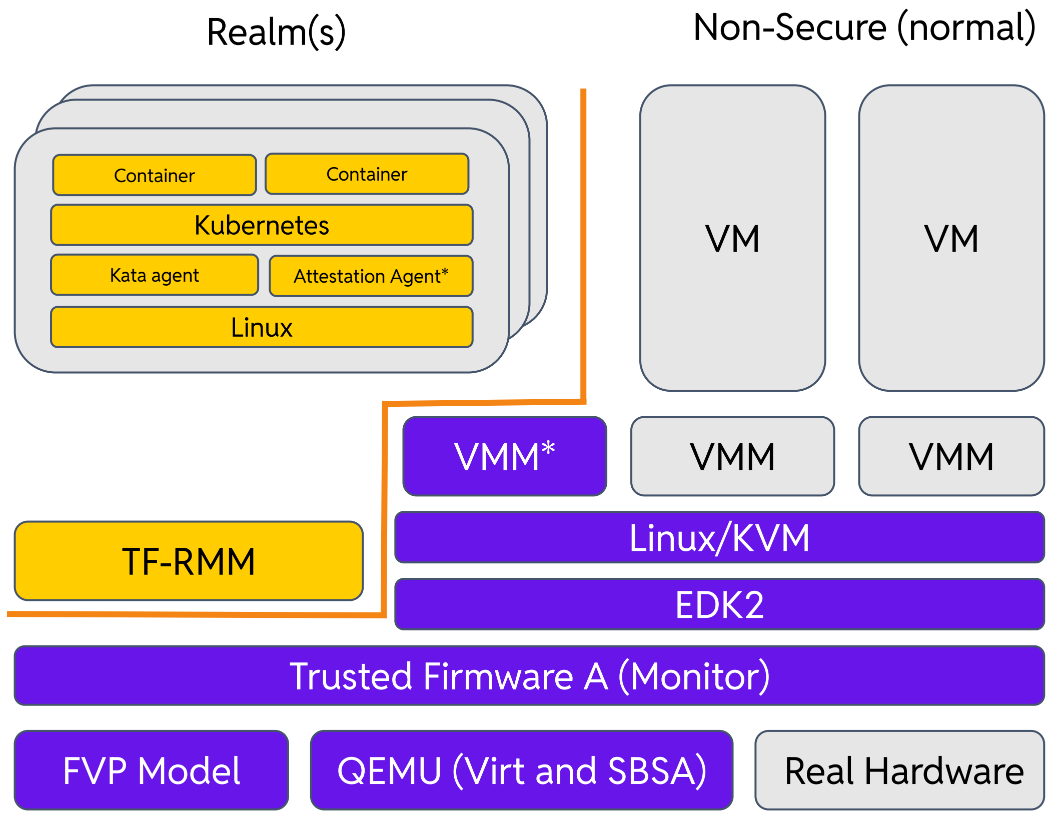
Goal → Support All Elements of a Confidential Computing Solution on Arm platforms

All the links to the projects and demonstration software are on the last slide

RATS Architecture



CCA-Aware Reference Software Stack



VMM:

QEMU
Kvmtool
Cloud Hypervisor

Attestation Agents:

Key Broker Service
Attestation tools

Verifier

Built with project Veraison

“Provides software components that can be used to build an Attestation Verification service”

Verifier can run in a local infrastructure

Linaro provides a cloud based instance publicly available

→ http://veraison.test.linaro.org:8443/.well-known/veraison/verification

→ Pre-populated with an attestation token that matches the TF-A

Key Broker Demonstration

Part of project Veraison

Built to exercise an end-to-end confidential computing scenario

Key broker server:

→ Runs in a local infrastructure

→ Pre-configured to use the public Linaro verifier

→ Can be configured to use a local verifier instance

Key broker application:

→ Already included in the rootfs of the CCA aware reference software stack

→ Can run without the stack with using a built-in RIM

Noteworthy Tools

CCA workload attestation:

→ Proof of concept for initial interaction with a verifier

→ Integrated to the reference stack rootfs

→ Useful to output the CCA attestation token to the command line

→ Enacts the “passport” model of RFC9334

CCA realm measurement tool:

→ Part of project Veraison

→ Computes the RIM and REM of a secure VM

RIM: Realm Initial Measurement

REM: Realm Extended Measurement

CCA End-to-End Scenario?

CCA aware reference software stack

- → Runs on QEMU (Virt Machine + SBSA) and FVP
- → Entirely composed of open source components
- → User space applications for appraisal of evidence

A Verifier running in the cloud

- → Based on project Veraison
- → Publicly available for test purposes

Key Broker

- → Runs on your local machine
- → Integrated with the stack and verifier

All Open Source

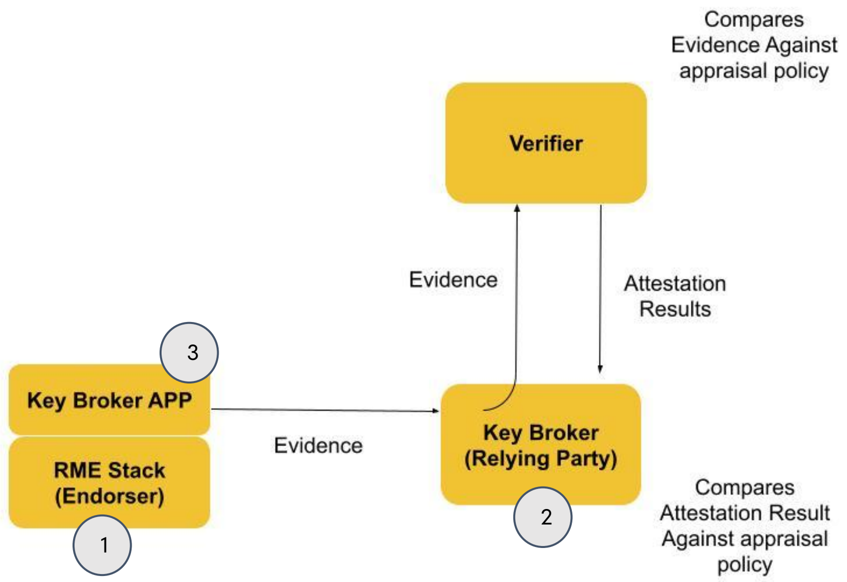
No Black Boxes

No Magic Binary Blobs

Putting It All Together

1) Start a Realm VM
2) Start the key broker service
3) Ask for a secret payload

RATS Architecture - Background-Check Model

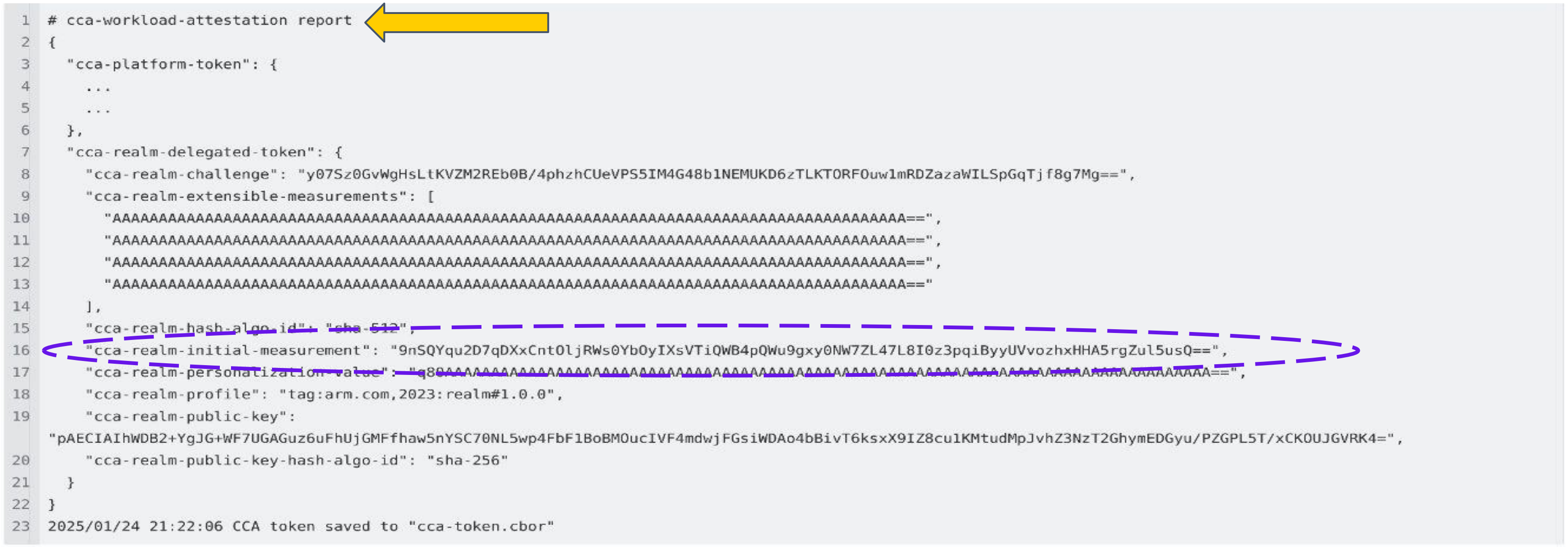

Compares Evidence Against appraisal policy
Verifier
Evidence
Attestation Results
3
Key Broker APP
RME Stack (Endorser)
1
Evidence
Key Broker (Relying Party)
2
Compares Attestation Result Against appraisal policy

Putting It All Together - Step 1

Acquire, build and run the CCA reference stack

→ All instructions are here

Once in a Realm VM, extract the CCA attestation token and look for the RIM:



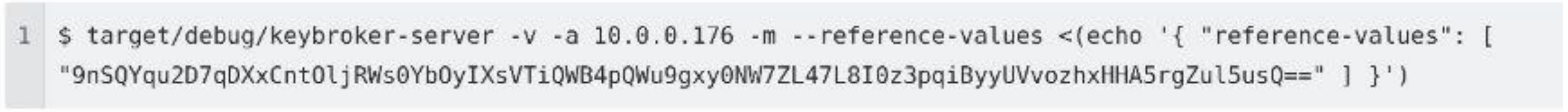
```
# cca-workload-attestation report
{
  "cca-platform-token": {
    ...
    ...
  },
  "cca-realm-delegated-token": {
    "cca-realm-challenge": "y07Sz0GvWgHsLtKVZM2REb0B/4phzhCUeVPS5IM4G48b1NEMUKD6zTLKTORFOuw1mRDZazaWILSpGqTjf8g7Mg==",
    "cca-realm-extensible-measurements": [
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==",
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==",
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==",
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=="
    ],
    "cca-realm-hash-algo-id": "sha-512",
    "cca-realm-initial-measurement": "9nSQYqu2D7qDXxCntOljRWs0YbOyIXsVTiQWB4pQWu9gxy0NW7ZL47L8I0z3pqiByyUVvozhxHHA5rgZul5usQ==",
    "cca-realm-personalization-value": "q80AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==",
    "cca-realm-profile": "tag:arm.com,2023:realm#1.0.0",
    "cca-realm-public-key":
"pAECIAIhWDB2+YgJG+WF7UGAGuz6uFhUjGMFfhaw5nYSC70NL5wp4FbF1BoBMOucIVF4mdwjFGsiWDAo4bBivT6ksxX9IZ8cu1KMtudMpJvhZ3NzT2GhymEDGyu/PZGPL5T/xCKOUJGVRK4=",
    "cca-realm-public-key-hash-algo-id": "sha-256"
  }
}
2025/01/24 21:22:06 CCA token saved to "cca-token.cbor"
```

Putting It All Together - Step 2

Acquire and build the key broker server

→ All instructions are here

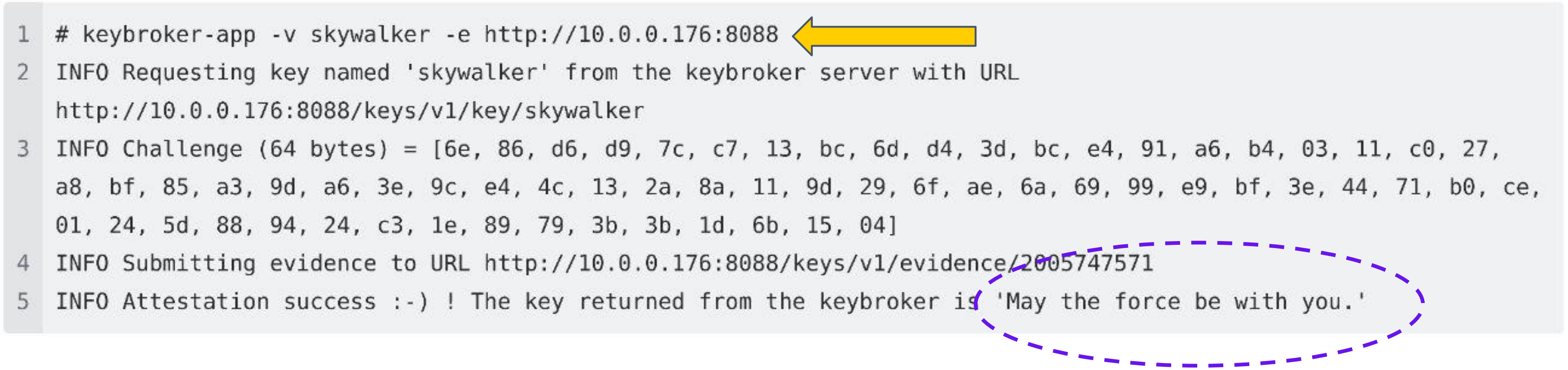
Start the key broker service locally with the Realm's RIM:



```
$ target/debug/keybroker-server -v -a 10.0.0.176 -m --reference-values <(echo '{ "reference-values": [
"9nSQYqu2D7qDXxCntOljRWs0YbOyIXsVTiQWB4pQWu9gxy0NW7ZL47L8I0z3pqiByyUVvozhxHHA5rgZul5usQ==" ] }')
```

Putting It All Together - Step 3

Back in the Realm VM, ask for secret payload “skywalker” from the key broker:

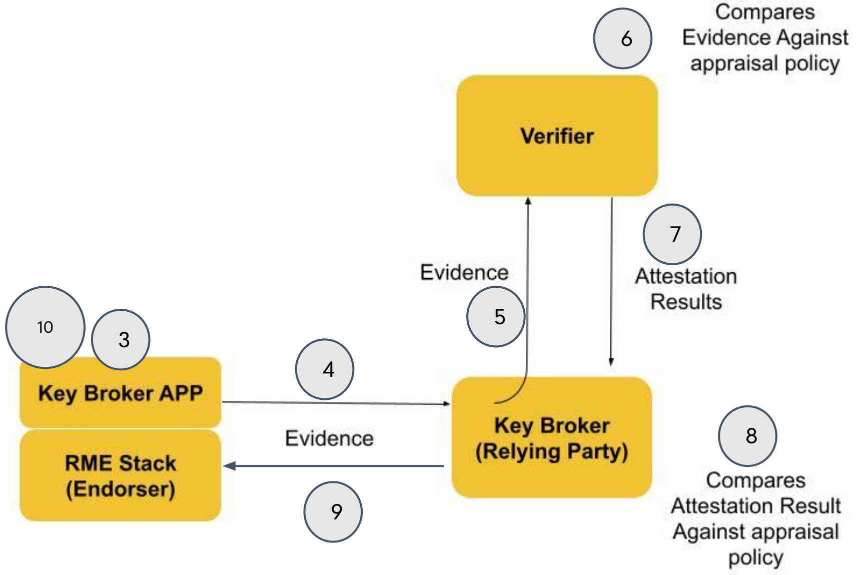


```
# keybroker-app -v skywalker -e http://10.0.0.176:8088
INFO Requesting key named 'skywalker' from the keybroker server with URL
http://10.0.0.176:8088/keys/v1/key/skywalker
INFO Challenge (64 bytes) = [6e, 86, d6, d9, 7c, c7, 13, bc, 6d, d4, 3d, bc, e4, 91, a6, b4, 03, 11, c0, 27,
a8, bf, 85, a3, 9d, a6, 3e, 9c, e4, 4c, 13, 2a, 8a, 11, 9d, 29, 6f, ae, 6a, 69, 99, e9, bf, 3e, 44, 71, b0, ce,
01, 24, 5d, 88, 94, 24, c3, 1e, 89, 79, 3b, 3b, 1d, 6b, 15, 04]
INFO Submitting evidence to URL http://10.0.0.176:8088/keys/v1/evidence/2005747571
INFO Attestation success :-) ! The key returned from the keybroker is 'May the force be with you.'
```

Understanding What Happened

3. Key broker App retrieves the CCA attestation token from “/sys/kernel/config/tsm”
4. The attestation token and a wrapping key are sent out to the Key Broker
5. The attestation token is forwarded to the verifier
6. The verifier verifies the platform token against appraisal policies
7. The verifier sends an attestation results
8. The key broker verifies the Realm token that contains the RIM against appraisal policies
9. Payload “skywalker” is encrypted with the wrapping key and sent back to the requester
10. The payload is decrypted and the content revealed

RATS Architecture - Background-Check Model


6
Compares Evidence Against appraisal policy
Verifier
Evidence
5
7
Attestation Results
10
3
Key Broker APP
4
RME Stack (Endorser)
Evidence
Key Broker (Relying Party)
8
Compares Attestation Result Against appraisal policy
9

Links

CCA-Aware reference stack: https://tinyurl.com/2anaptkn

Project Veraison: https://github.com/veraison

Remote Attestation Procedures (RATS) Architecture: https://www.ietf.org/rfc/rfc9334.html

Key Broker Demonstration: https://github.com/veraison/keybroker-demo

CCA workload attestation PoC: https://tinyurl.com/25oba4cq

CCA realm measurement tool: https://github.com/veraison/cca-realm-measurements

Too Much Information, Too Little Time

Questions?