Enhancing the Security Posture of IoT: Study of Remote Attestation at the Deep Edge

Chris Meier, Raj Pal

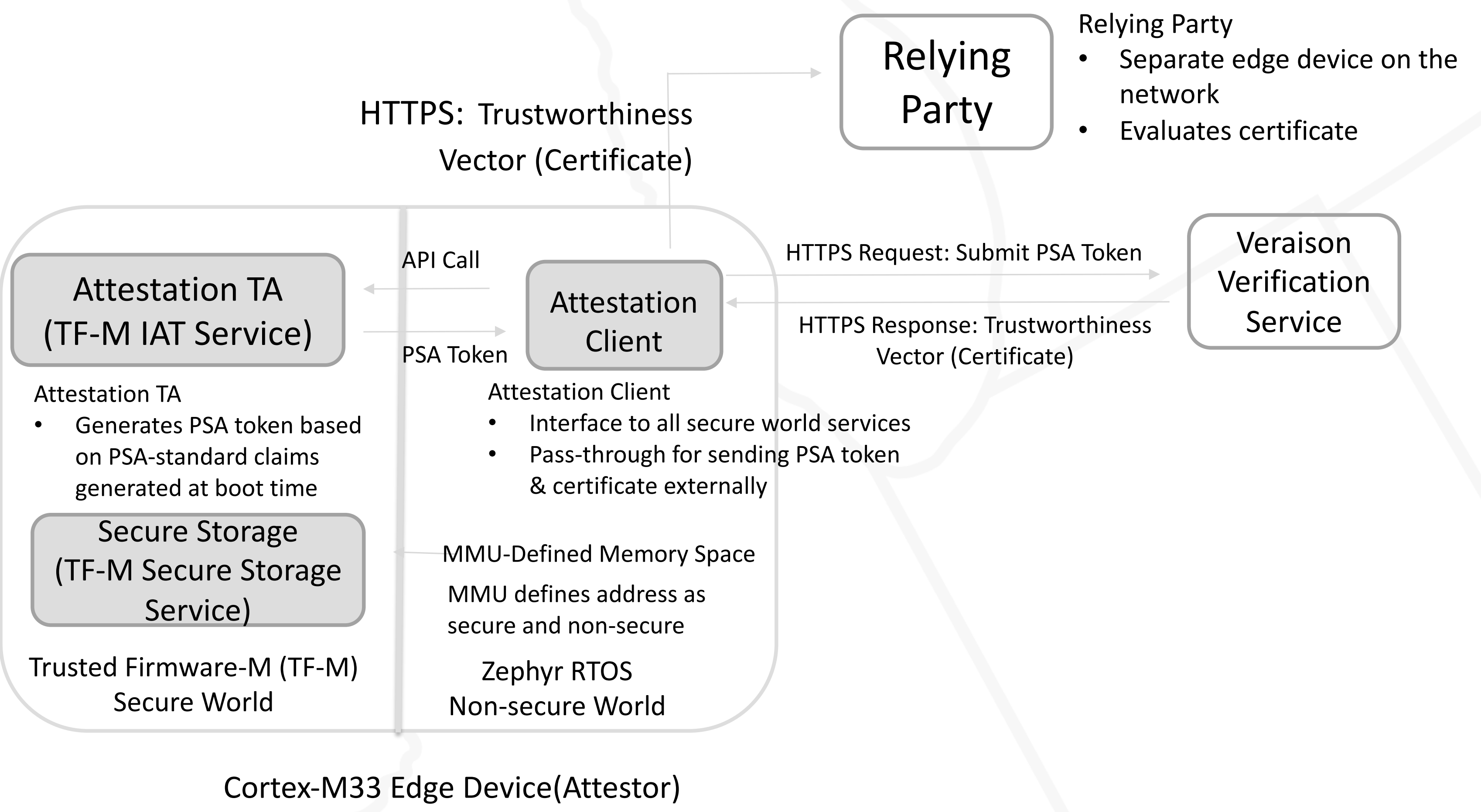
Laboratory for Advanced Cybersecurity Research, National Security Agency

Research Problem: Identify potentially malicious behavior in embedded IoT devices.

Edge Device Attestation

Attestation Procedure

1. PSA-standard claims are generated by bootloader at boot and stored in secure memory
2. Attestation Client requests PSA token from attestation TA via API call
   1. Initially done on boot
   2. Future: Runtime
3. Attestation TA returns PSA token to Attestation Client and Attestation Client submits PSA token to Veraison verification service
4. Veraison verification service returns a certificate
5. Certificate is stored in secure storage on the attestor
6. Certificate is presented to relying party via HTTPS


Relying Party
HTTPS: Trustworthiness Vector (Certificate)
Attestation TA (TF-M IAT Service)
API Call
PSA Token
Attestation Client
HTTPS Request: Submit PSA Token
HTTPS Response: Trustworthiness Vector (Certificate)
Veraison Verification Service
Secure Storage (TF-M Secure Storage Service)
MMU-Defined Memory Space
Trusted Firmware-M (TF-M) Secure World
Zephyr RTOS Non-secure World

Relying Party
- Separate edge device on the network
- Evaluates certificate

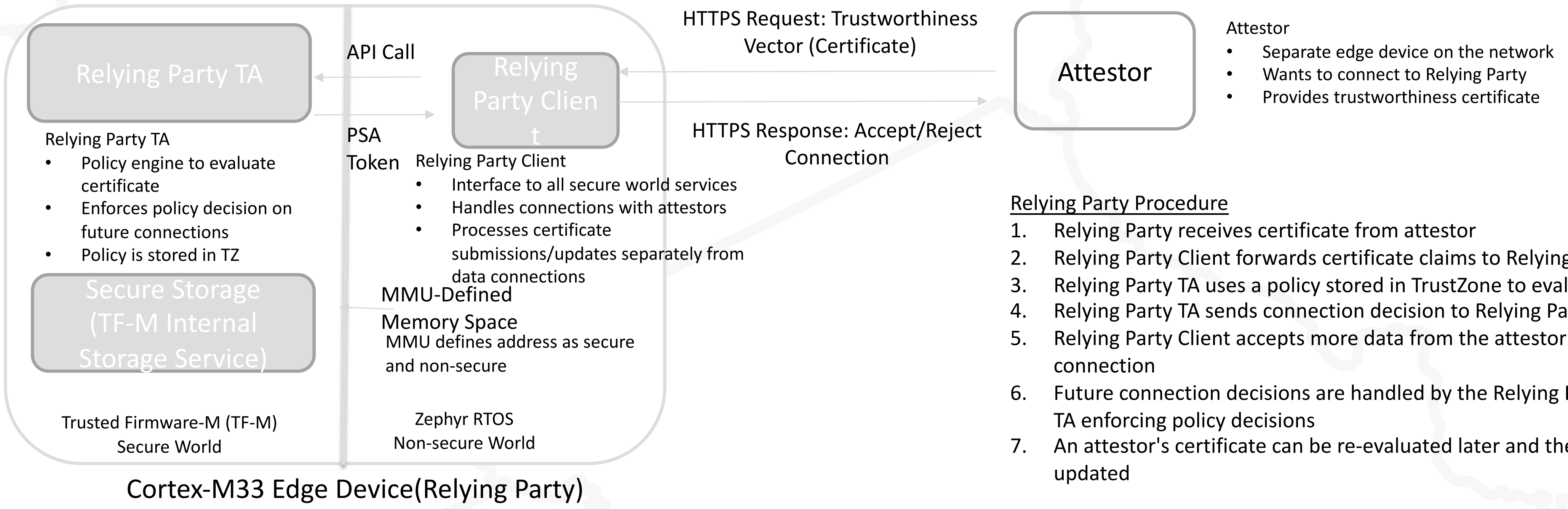
Attestation TA
- Generates PSA token based on PSA-standard claims generated at boot time

Attestation Client
- Interface to all secure world services
- Pass-through for sending PSA token & certificate externally

MMU defines address as secure and non-secure

Cortex-M33 Edge Device(Attestor)

Relying Party Appraisal


Relying Party TA
API Call
PSA Token
Relying Party Client
HTTPS Request: Trustworthiness Vector (Certificate)
HTTPS Response: Accept/Reject Connection
Attestor
Secure Storage (TF-M Internal Storage Service)
MMU-Defined Memory Space
Trusted Firmware-M (TF-M) Secure World
Zephyr RTOS Non-secure World

Relying Party TA
- Policy engine to evaluate certificate
- Enforces policy decision on future connections
- Policy is stored in TZ

Relying Party Client
- Interface to all secure world services
- Handles connections with attestors
- Processes certificate submissions/updates separately from data connections

MMU defines address as secure and non-secure

Attestor
- Separate edge device on the network
- Wants to connect to Relying Party
- Provides trustworthiness certificate

Cortex-M33 Edge Device(Relying Party)

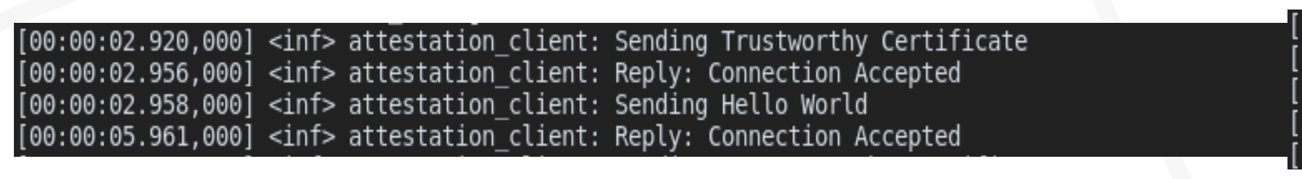
Relying Party Procedure

1. Relying Party receives certificate from attestor
2. Relying Party Client forwards certificate claims to Relying Party TA
3. Relying Party TA uses a policy stored in TrustZone to evaluate claims
4. Relying Party TA sends connection decision to Relying Party Client
5. Relying Party Client accepts more data from the attestor or closes connection
6. Future connection decisions are handled by the Relying Party TA enforcing policy decisions
7. An attestor's certificate can be re-evaluated later and the policy updated

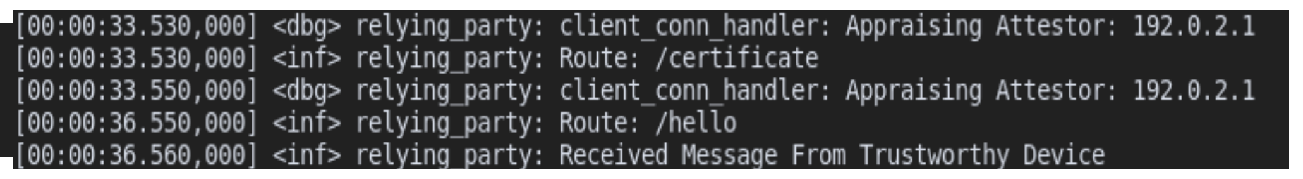
Attestor (Zephyr RTOS/TF-M)

Relying Party (Zephyr RTOS/TF-M)

Trustworthy Certificate

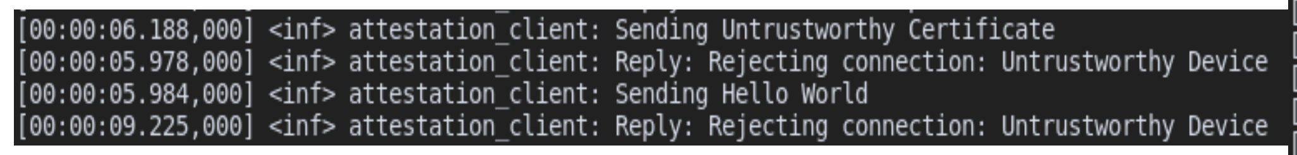


```
[00:00:02.920,000] <inf> attestation_client: Sending Trustworthy Certificate
[00:00:02.956,000] <inf> attestation_client: Reply: Connection Accepted
[00:00:02.958,000] <inf> attestation_client: Sending Hello World
[00:00:05.961,000] <inf> attestation_client: Reply: Connection Accepted
```

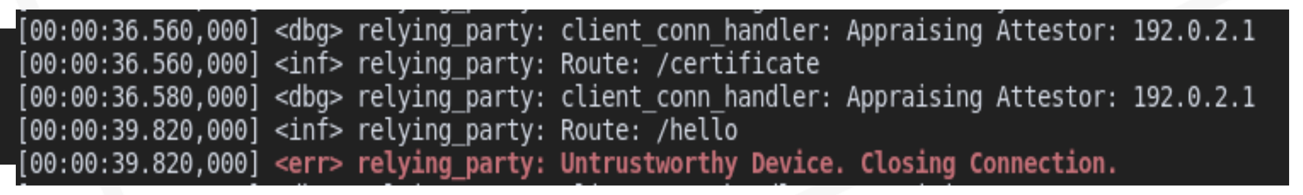


```
[00:00:33.530,000] <dbg> relying_party: client_conn_handler: Appraising Attestor: 192.0.2.1
[00:00:33.530,000] <inf> relying_party: Route: /certificate
[00:00:33.550,000] <dbg> relying_party: client_conn_handler: Appraising Attestor: 192.0.2.1
[00:00:36.550,000] <inf> relying_party: Route: /hello
[00:00:36.560,000] <inf> relying_party: Received Message From Trustworthy Device
```

Untrustworthy Certificate



```
[00:00:06.188,000] <inf> attestation_client: Sending Untrustworthy Certificate
[00:00:05.978,000] <inf> attestation_client: Reply: Rejecting connection: Untrustworthy Device
[00:00:05.984,000] <inf> attestation_client: Sending Hello World
[00:00:09.225,000] <inf> attestation_client: Reply: Rejecting connection: Untrustworthy Device
```



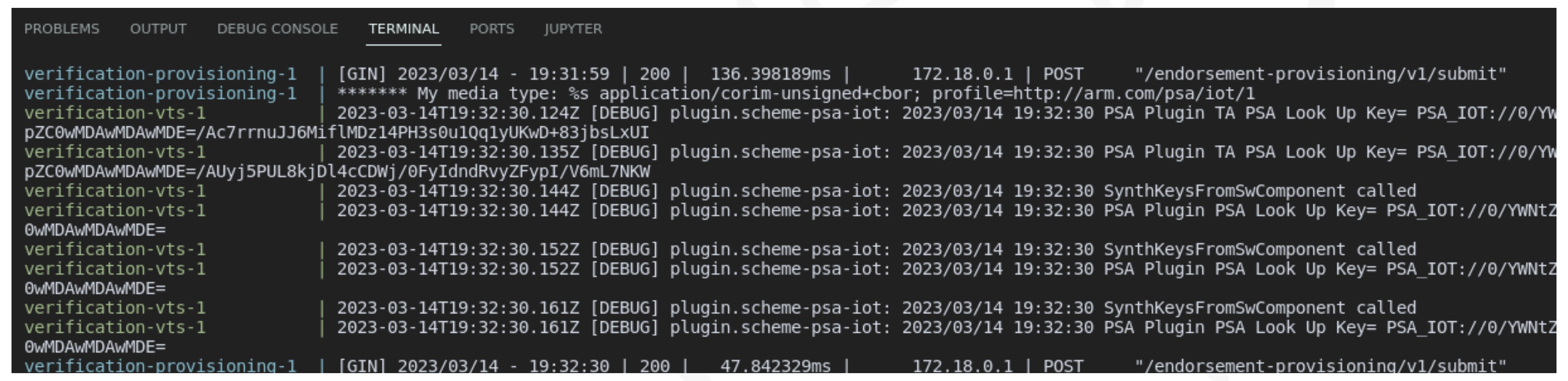
```
[00:00:36.560,000] <dbg> relying_party: client_conn_handler: Appraising Attestor: 192.0.2.1
[00:00:36.560,000] <inf> relying_party: Route: /certificate
[00:00:36.580,000] <dbg> relying_party: client_conn_handler: Appraising Attestor: 192.0.2.1
[00:00:39.820,000] <inf> relying_party: Route: /hello
[00:00:39.820,000] <err> relying_party: Untrustworthy Device. Closing Connection.
```

Verifier

Verification Procedure

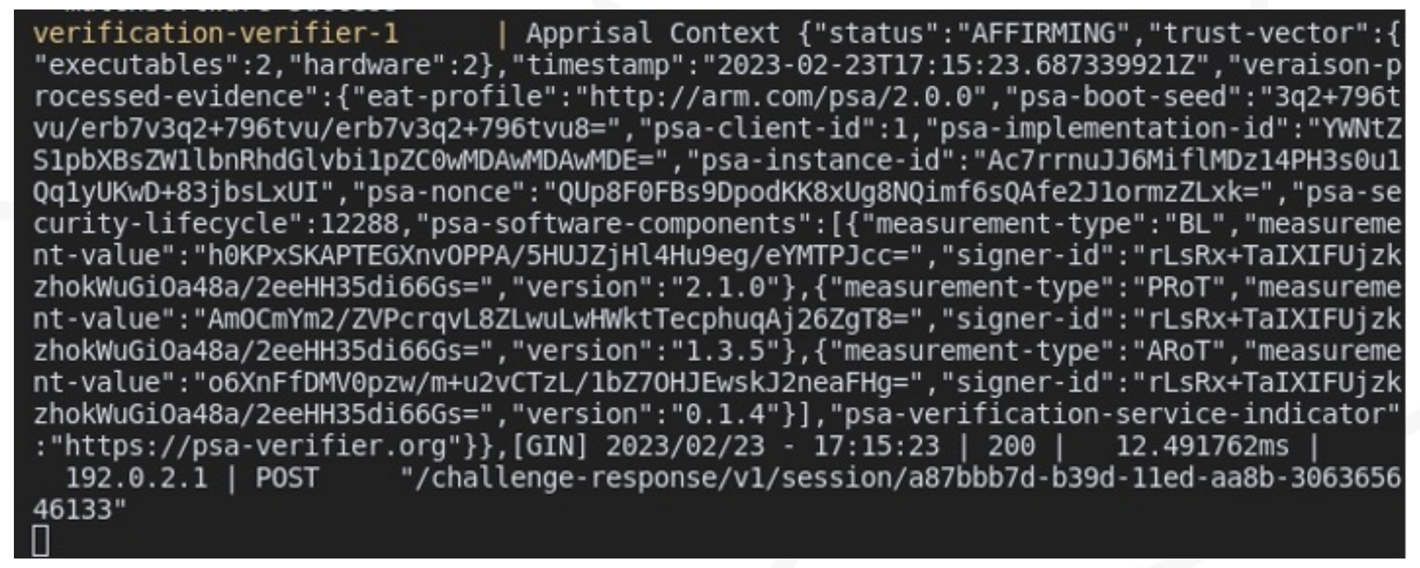
1. Endorsements (golden values) and trust anchors (used to verify PSA token signature) are loaded via the provisioning service at creation time
2. Attestor submits PSA token to verification service
3. Verification passes PSA token to PSA plugin in Veraison Trusted Service
4. PSA plugin parses claims in token, verifies token signature with trust anchors, compares claims against endorsements, and evaluates comparisons using the policy engine to generate a certificate
5. Certificate is passed to the verification service which forwards it to the attestor

Verifier



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    JUPYTER

verification-provisioning-1  | [GIN] 2023/03/14 - 19:31:59 | 200 |  136.398189ms |      172.18.0.1 | POST     "/endorsement-provisioning/v1/submit"
verification-provisioning-1  | ******* My media type: %s application/corim-unsigned+cbor; profile=http://arm.com/psa/iot/1
verification-vts-1           | 2023-03-14T19:32:30.124Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 PSA Plugin TA PSA Look Up Key= PSA_IOT://0/YW
pZC0wMDAwMDAwMDE=/Ac7rrnuJJ6MiflMDz14PH3s0u1Qq1yUKwD+83jbsLxUI
verification-vts-1           | 2023-03-14T19:32:30.135Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 PSA Plugin TA PSA Look Up Key= PSA_IOT://0/YW
pZC0wMDAwMDAwMDE=/AUyj5PUL8kjDl4cCDWj/0FyIdndRvyZFypI/V6mL7NKW
verification-vts-1           | 2023-03-14T19:32:30.144Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 SynthKeysFromSwComponent called
verification-vts-1           | 2023-03-14T19:32:30.144Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 PSA Plugin PSA Look Up Key= PSA_IOT://0/YWNtZ
0wMDAwMDAwMDE=
verification-vts-1           | 2023-03-14T19:32:30.152Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 SynthKeysFromSwComponent called
verification-vts-1           | 2023-03-14T19:32:30.152Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 PSA Plugin PSA Look Up Key= PSA_IOT://0/YWNtZ
0wMDAwMDAwMDE=
verification-vts-1           | 2023-03-14T19:32:30.161Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 SynthKeysFromSwComponent called
verification-vts-1           | 2023-03-14T19:32:30.161Z [DEBUG] plugin.scheme-psa-iot: 2023/03/14 19:32:30 PSA Plugin PSA Look Up Key= PSA_IOT://0/YWNtZ
0wMDAwMDAwMDE=
verification-provisioning-1  | [GIN] 2023/03/14 - 19:32:30 | 200 |   47.842329ms |      172.18.0.1 | POST     "/endorsement-provisioning/v1/submit"
```

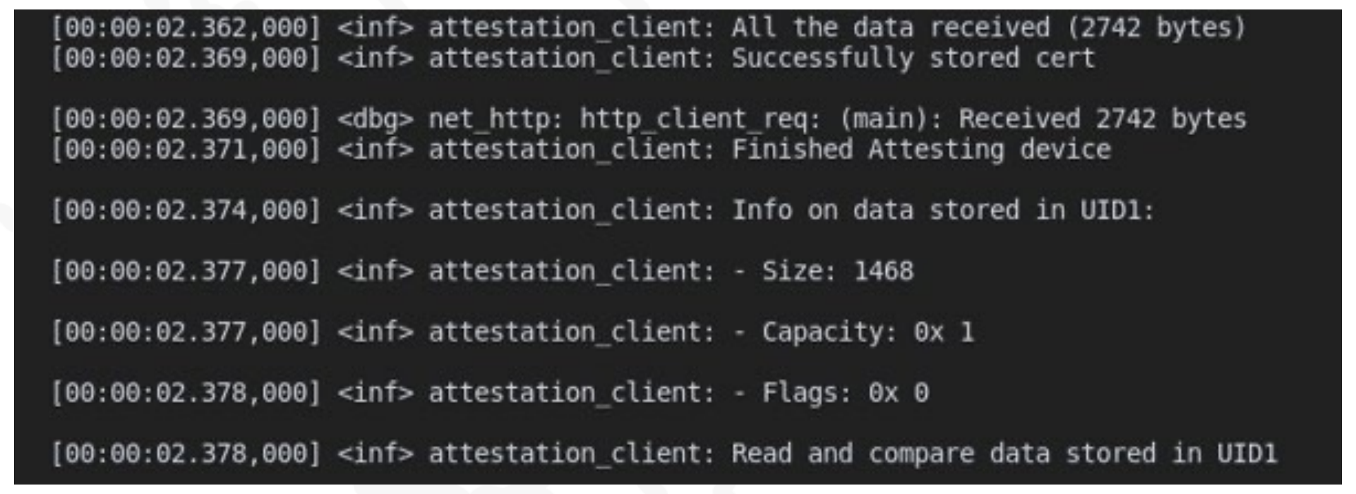
Trust Anchor & Endorsement Provisioning



```
verification-verifier-1      | Apprisal Context {"status":"AFFIRMING","trust-vector":{
"executables":2,"hardware":2},"timestamp":"2023-02-23T17:15:23.687339921Z","veraison-p
rocessed-evidence":{"eat-profile":"http://arm.com/psa/2.0.0","psa-boot-seed":"3q2+796t
vu/erb7v3q2+796tvu/erb7v3q2+796tvu8=","psa-client-id":1,"psa-implementation-id":"YWNtZ
S1pbXBsZW1lbnRhdGlvbi1pZC0wMDAwMDAwMDE=","psa-instance-id":"Ac7rrnuJJ6MiflMDz14PH3s0u1
Qq1yUKwD+83jbsLxUI","psa-nonce":"QUp8F0FBs9DpodKK8xUg8NQimf6sQAfe2J1ormzZLxk=","psa-se
curity-lifecycle":12288,"psa-software-components":[{"measurement-type":"BL","measureme
nt-value":"h0KPxSKAPTEGXnvOPPA/5HUJZjHl4Hu9eg/eYMTPJcc=","signer-id":"rLsRx+TaIXIFUjzk
zhokWuGiOa48a/2eeHH35di66Gs=","version":"2.1.0"},{"measurement-type":"PRoT","measureme
nt-value":"AmOCmYm2/ZVPcrqvL8ZLwuLwHWktTecphuqAj26ZgT8=","signer-id":"rLsRx+TaIXIFUjzk
zhokWuGiOa48a/2eeHH35di66Gs=","version":"1.3.5"},{"measurement-type":"ARoT","measureme
nt-value":"o6XnFfDMV0pzw/m+u2vCTzL/1bZ7OHJEwskJ2neaFHg=","signer-id":"rLsRx+TaIXIFUjzk
zhokWuGiOa48a/2eeHH35di66Gs=","version":"0.1.4"}],"psa-verification-service-indicator"
:"https://psa-verifier.org"}},[GIN] 2023/02/23 - 17:15:23 | 200 |   12.491762ms |
  192.0.2.1 | POST     "/challenge-response/v1/session/a87bbb7d-b39d-11ed-aa8b-3063656
46133"
```

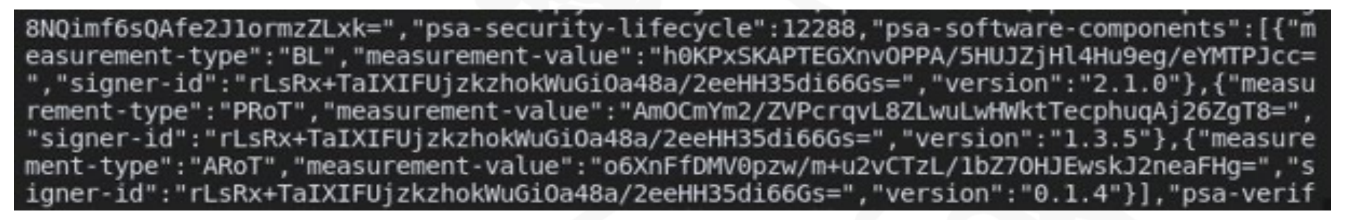
Trustworthiness Certificate

Attestor (Zephyr RTOS/TF-M)



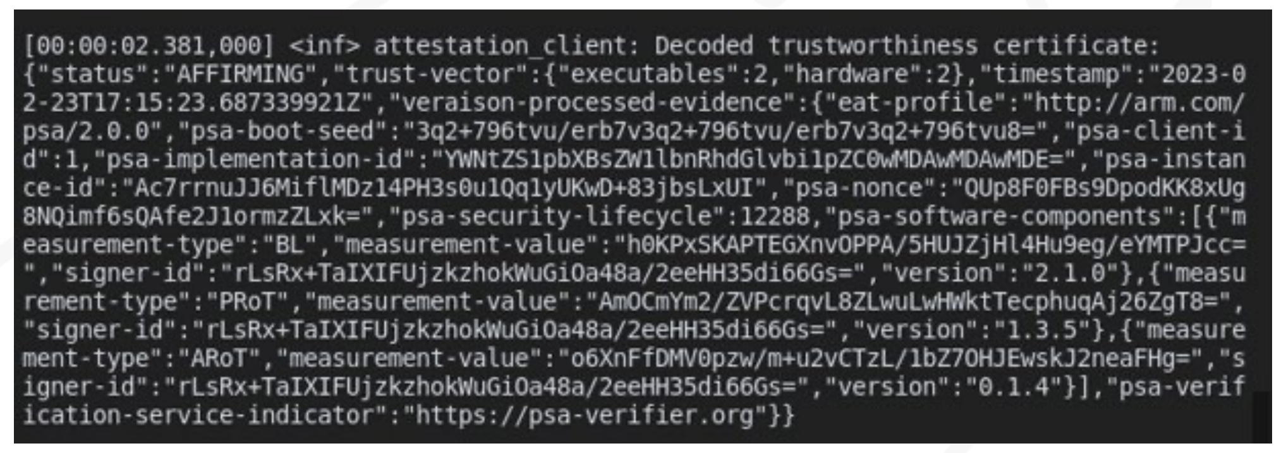
```
[00:00:02.362,000] <inf> attestation_client: All the data received (2742 bytes)
[00:00:02.369,000] <inf> attestation_client: Successfully stored cert

[00:00:02.369,000] <dbg> net_http: http_client_req: (main): Received 2742 bytes
[00:00:02.371,000] <inf> attestation_client: Finished Attesting device

[00:00:02.374,000] <inf> attestation_client: Info on data stored in UID1:

[00:00:02.377,000] <inf> attestation_client: - Size: 1468

[00:00:02.377,000] <inf> attestation_client: - Capacity: 0x 1

[00:00:02.378,000] <inf> attestation_client: - Flags: 0x 0

[00:00:02.378,000] <inf> attestation_client: Read and compare data stored in UID1
```

Certificate stored in TrustZone



```
8NQimf6sQAfe2J1ormzZLxk=","psa-security-lifecycle":12288,"psa-software-components":[{"m
easurement-type":"BL","measurement-value":"h0KPxSKAPTEGXnvOPPA/5HUJZjHl4Hu9eg/eYMTPJcc=
","signer-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"2.1.0"},{"measu
rement-type":"PRoT","measurement-value":"AmOCmYm2/ZVPcrqvL8ZLwuLwHWktTecphuqAj26ZgT8=",
"signer-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"1.3.5"},{"measure
ment-type":"ARoT","measurement-value":"o6XnFfDMV0pzw/m+u2vCTzL/1bZ7OHJEwskJ2neaFHg=","s
igner-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"0.1.4"}],"psa-verif
```

Software Measurements



```
[00:00:02.381,000] <inf> attestation_client: Decoded trustworthiness certificate:
{"status":"AFFIRMING","trust-vector":{"executables":2,"hardware":2},"timestamp":"2023-0
2-23T17:15:23.687339921Z","veraison-processed-evidence":{"eat-profile":"http://arm.com/
psa/2.0.0","psa-boot-seed":"3q2+796tvu/erb7v3q2+796tvu/erb7v3q2+796tvu8=","psa-client-i
d":1,"psa-implementation-id":"YWNtZS1pbXBsZW1lbnRhdGlvbi1pZC0wMDAwMDAwMDE=","psa-instan
ce-id":"Ac7rrnuJJ6MiflMDz14PH3s0u1Qq1yUKwD+83jbsLxUI","psa-nonce":"QUp8F0FBs9DpodKK8xUg
8NQimf6sQAfe2J1ormzZLxk=","psa-security-lifecycle":12288,"psa-software-components":[{"m
easurement-type":"BL","measurement-value":"h0KPxSKAPTEGXnvOPPA/5HUJZjHl4Hu9eg/eYMTPJcc=
","signer-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"2.1.0"},{"measu
rement-type":"PRoT","measurement-value":"AmOCmYm2/ZVPcrqvL8ZLwuLwHWktTecphuqAj26ZgT8=",
"signer-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"1.3.5"},{"measure
ment-type":"ARoT","measurement-value":"o6XnFfDMV0pzw/m+u2vCTzL/1bZ7OHJEwskJ2neaFHg=","s
igner-id":"rLsRx+TaIXIFUjzkzhokWuGiOa48a/2eeHH35di66Gs=","version":"0.1.4"}],"psa-verif
ication-service-indicator":"https://psa-verifier.org"}}
```

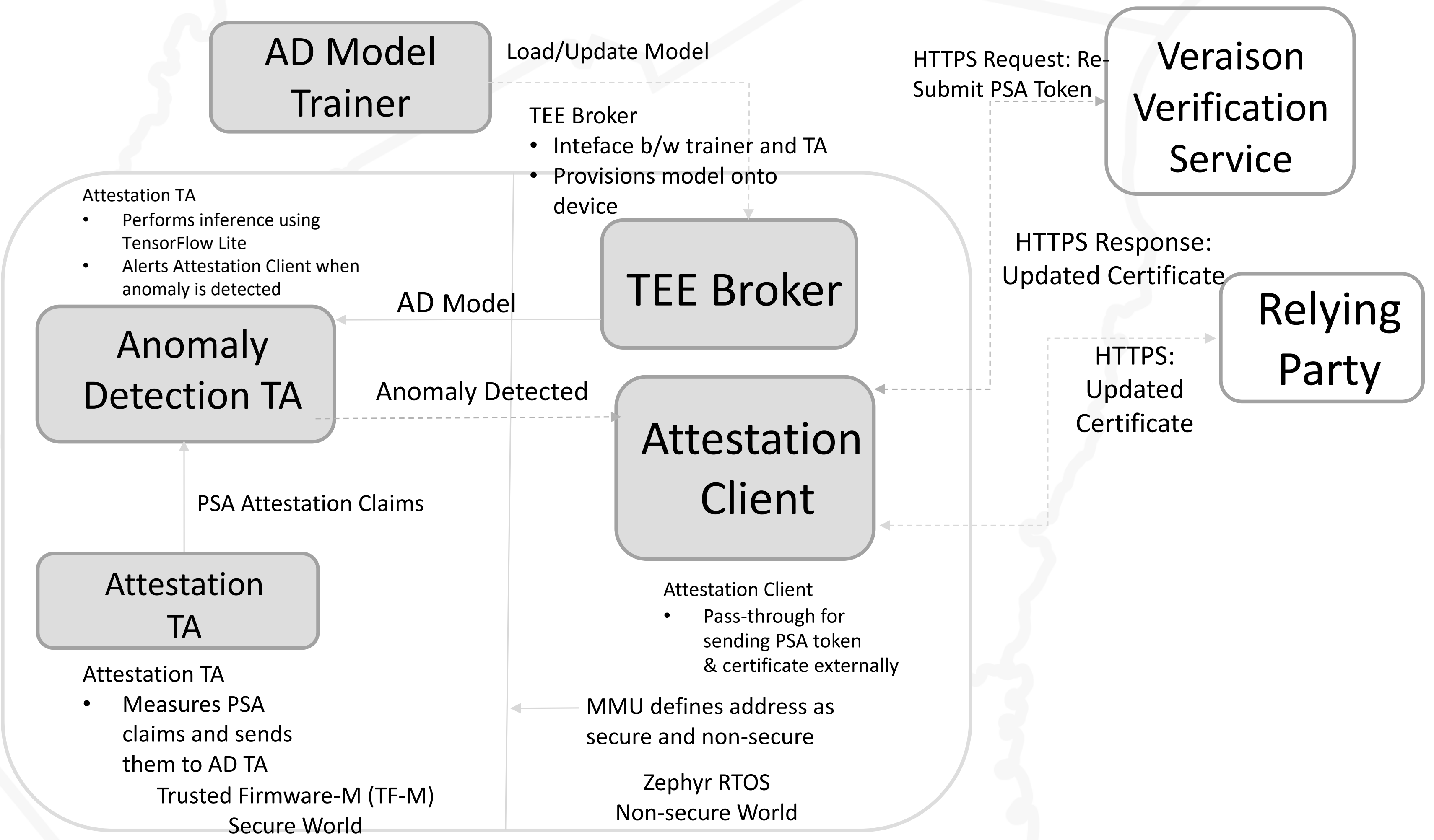
Trustworthiness Certificate

Future Work: Anomaly Detection

draft-ietf-teep-architecture-18

Anomaly Detection Procedure

1. Reinforcement Learning Service trains anomaly detection model and sends it to the attestor, which stores using secure storage
2. Anomaly detection TA (AD) loads the model and begins performing the **inference**
3. Claims are measured periodically sent to the AD from the Attestation TA which are used to infer trustworthiness
4. When an anomaly is detected, the attestor resubmits evidence to the Veraison verification service and receives an updated certificate (categorizing it as trustworthy or untrustworthy)
5. Updated certificate is distributed to other relaying parties


AD Model Trainer
Load/Update Model
TEE Broker
AD Model
Anomaly Detection TA
Anomaly Detected
Attestation Client
PSA Attestation Claims
Attestation TA
HTTPS Request: Re-Submit PSA Token
Veraison Verification Service
HTTPS Response: Updated Certificate
HTTPS: Updated Certificate
Relying Party
Trusted Firmware-M (TF-M) Secure World
Zephyr RTOS Non-secure World

TEE Broker
- Inteface b/w trainer and TA
- Provisions model onto device

Attestation TA
- Performs inference using TensorFlow Lite
- Alerts Attestation Client when anomaly is detected

Attestation TA
- Measures PSA claims and sends them to AD TA

Attestation Client
- Pass-through for sending PSA token & certificate externally

MMU defines address as secure and non-secure

10TH ANNUAL
HOT TOPICS in the SCIENCE OF SECURITY
APRIL 3 - 5, 2023 | *Virtually hosted by* The National Security Agency
hotsos.org