Oracle Groundbreakers

Device Attestation using Java Card

Patrick Van Haver
Security Solutions Architect
Oracle - Java Platform Group

Nicolae Bors
Software Engineer
Oracle - Java Platform Group

Dec. 2020

ORACLE

Safe Harbor Statement

Agenda

Agenda

- Purpose of Device Attestations
- How Device Attestation works?
- Device Attestations using Java Card – Demo
- Conclusion

Purpose of Device Attestations

- Get reliable evidence on the characteristics and state of a device
  - Device identity and manufacturer,
  - Security state and capabilities,
  - Software versions installed,
  - Location
  - …
- Typically used
  - to detect rogue devices during on-boarding,
  - to perform remote monitoring and enforce security policies,
  - to manage device lifecycle, detect non-updated or tampered devices,
  - …

Entity Attestation Tokens

Entity Attestation Token IETF draft

Entity Attestation Protocol draft specification

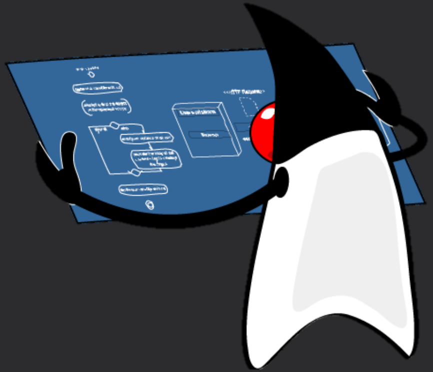
Requirements

- Self-contained (no dependency on protocol)
- Extensible list of claims
- Simple and compact encoding
- Support for integrity, authenticity and confidentiality
- Supports for multiple signing and encryption schemes

Token structure based on existing standards, and extended with specific claims

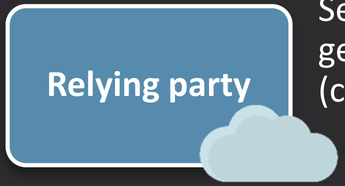
- Either JSON Web Token (JWT – RFC7519),
- Or CBOR Web Token (CWT – RFC8392), CBOR Object Signing & Encryption (COSE – RFC8152)

Agenda



How Device Attestation works?

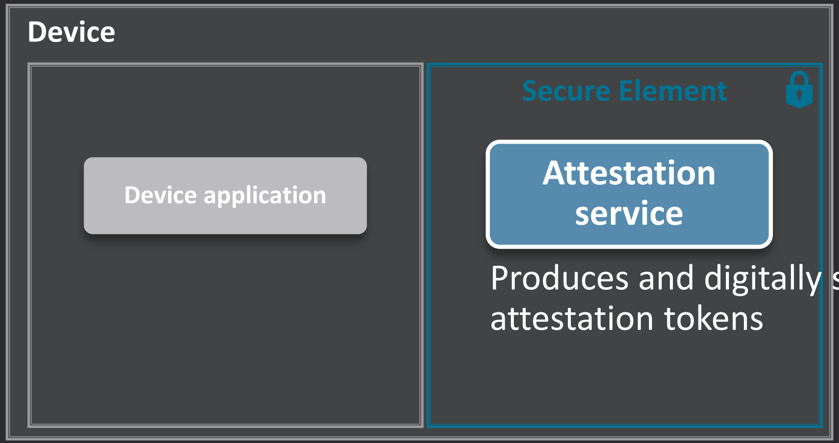
Actors



Service Provider who wants to get reliable information from a device (characteristics, state, …)

Verification service

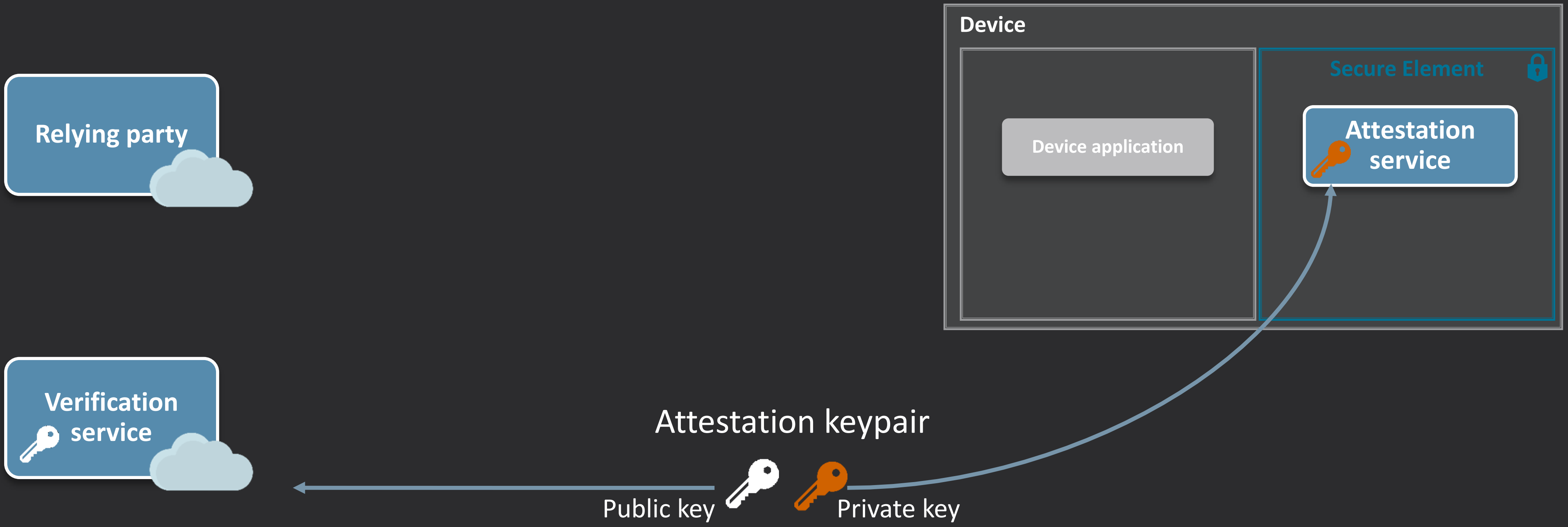
Service to verify authenticity of attestation tokens



Produces and digitally sign attestation tokens

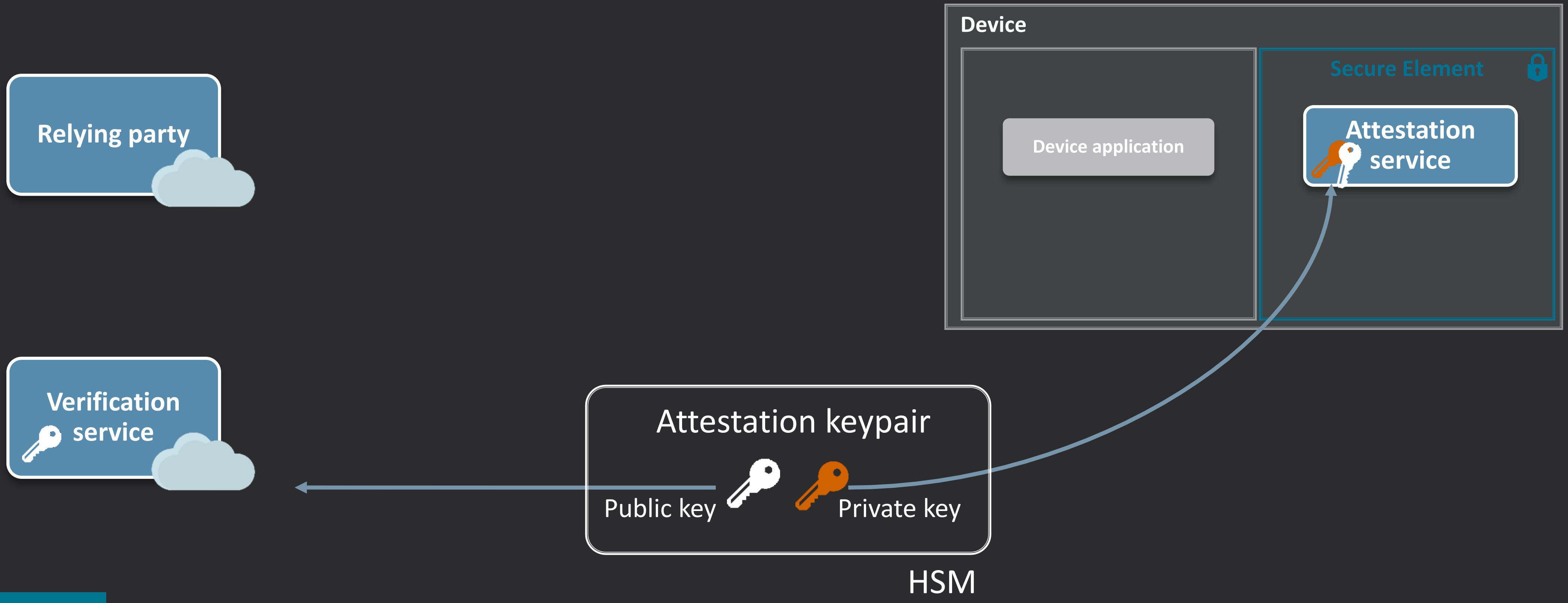
How Device Attestation works?

Key Provisioning



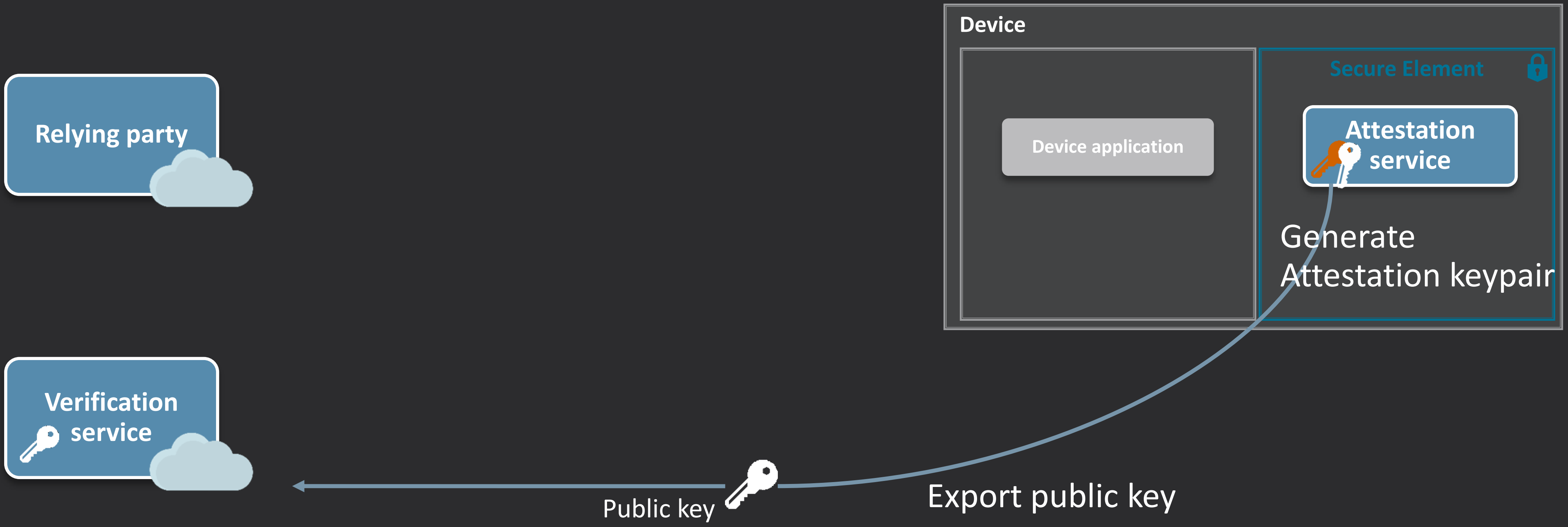
How Device Attestation works?

Key Provisioning – using HSM



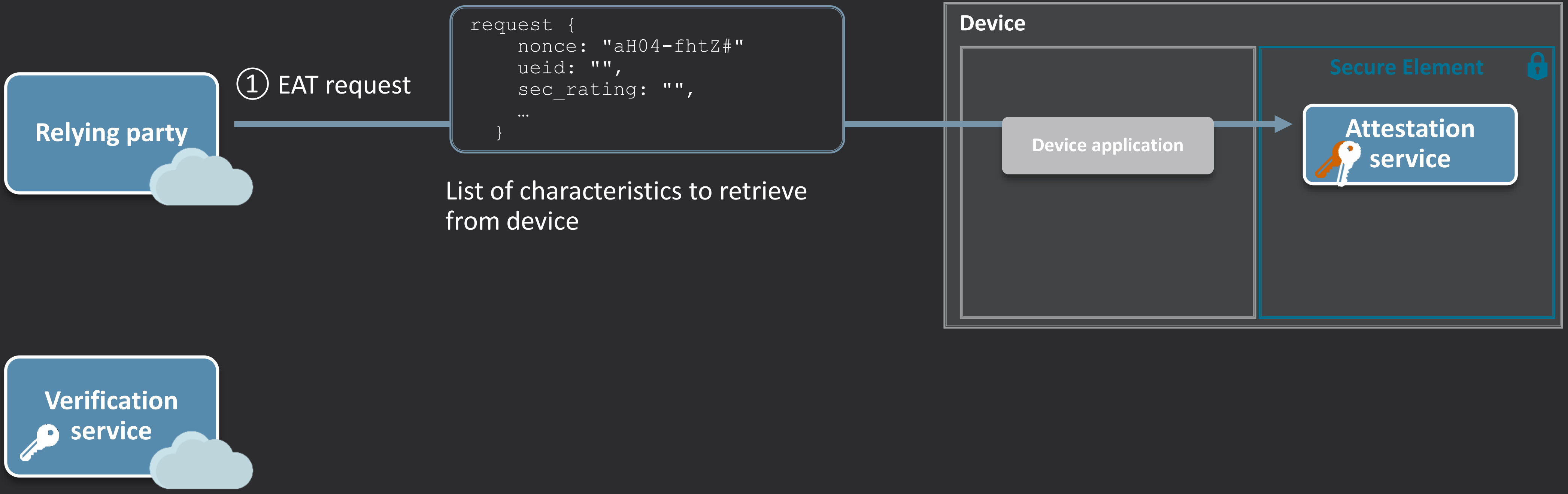
How Device Attestation works?

Key Provisioning – using on-board key-generation



How Device Attestation works?

Attestation Request

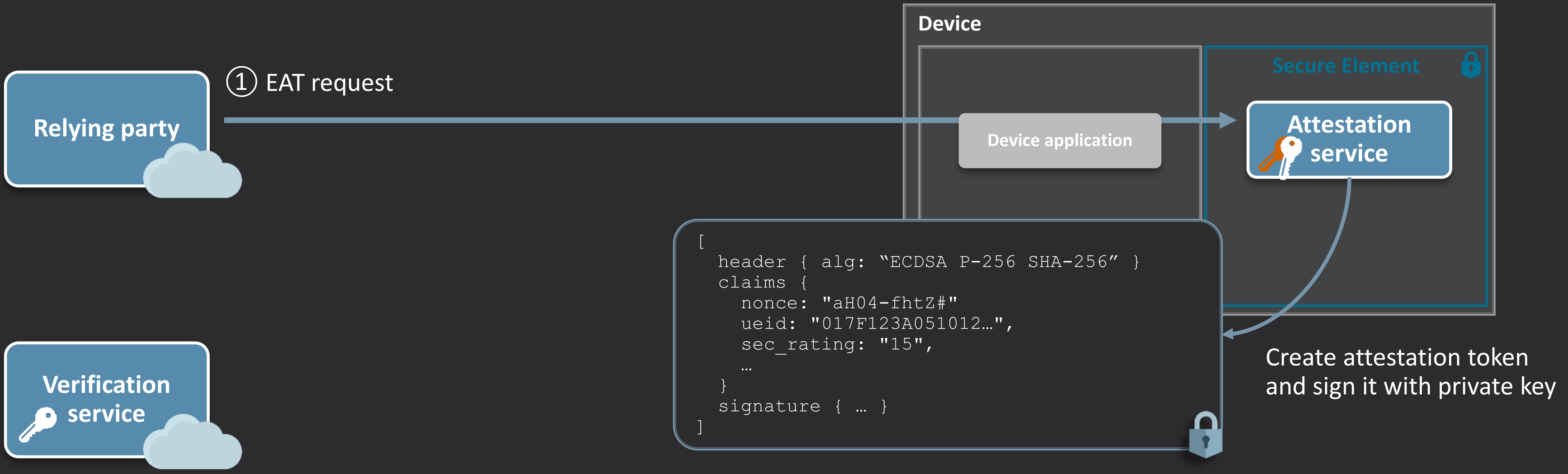


```
request {
    nonce: "aH04-fhtZ#"
    ueid: "",
    sec_rating: "",
    …
  }
```

List of characteristics to retrieve from device

How Device Attestation works?

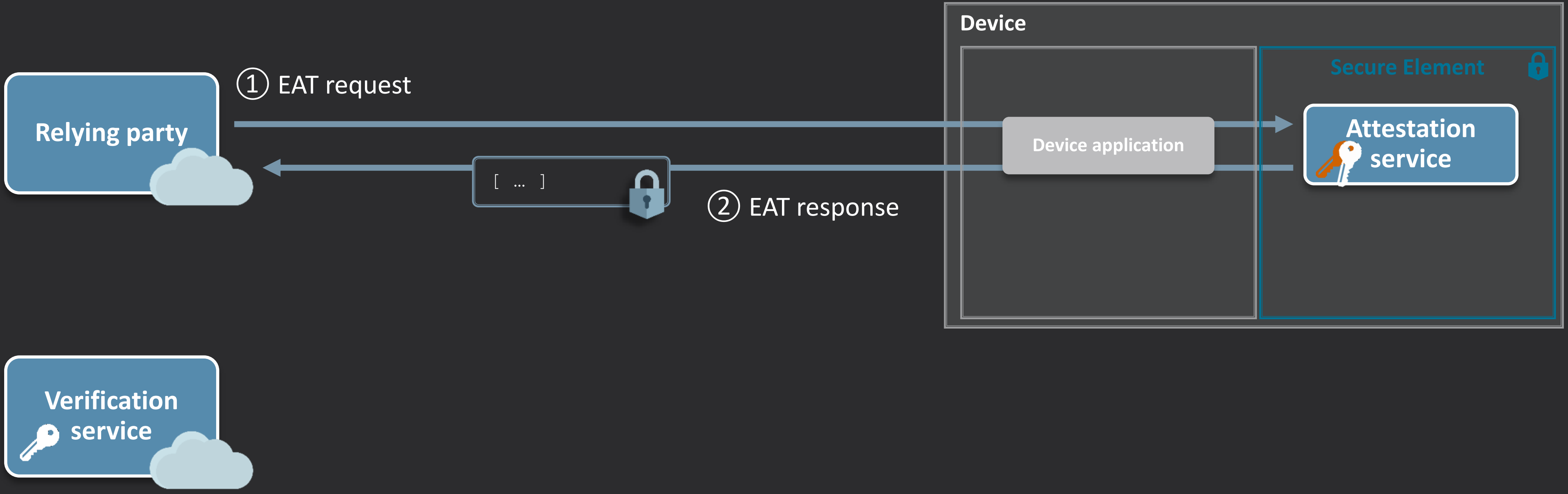
Attestation Token generated by Secure Element



```
[
  header { alg: "ECDSA P-256 SHA-256" }
  claims {
    nonce: "aH04-fhtZ#"
    ueid: "017F123A051012…",
    sec_rating: "15",
    …
  }
  signature { … }
]
```

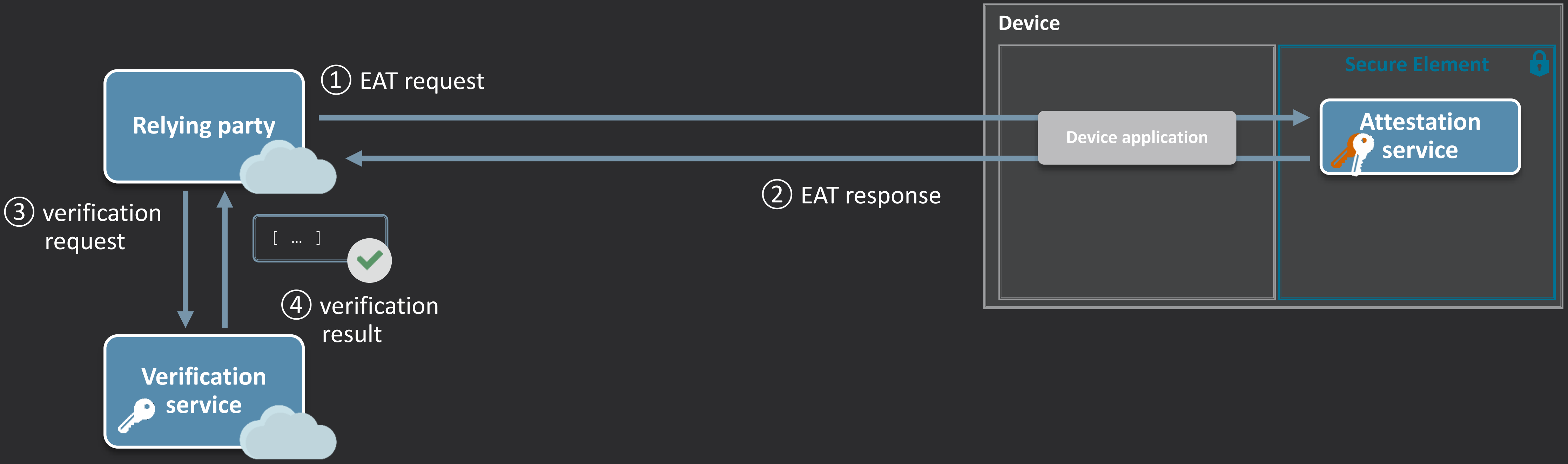
How Device Attestation works?

Attestation Response



How Device Attestation works?

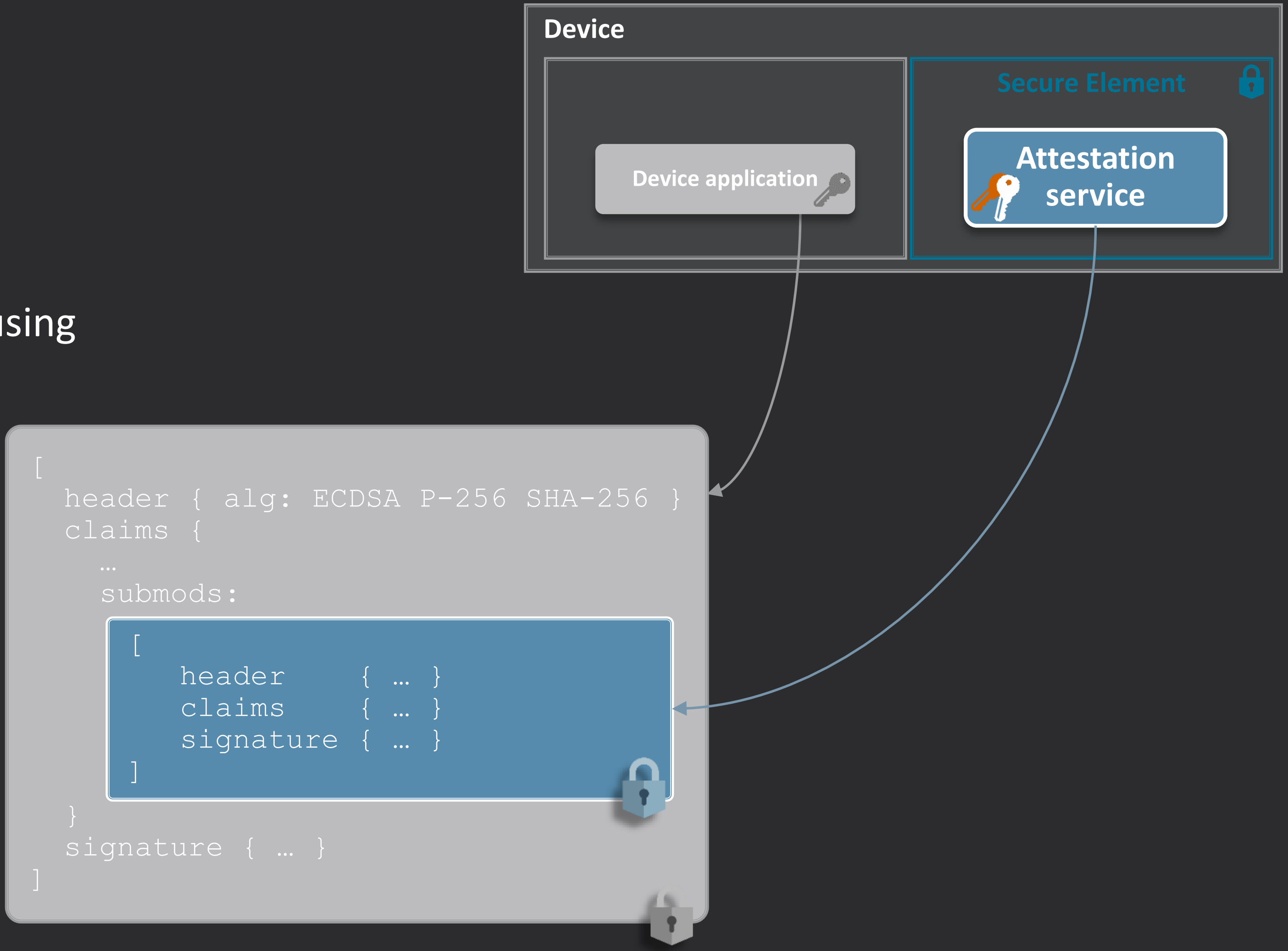
Attestation Verification



How Device Attestation works?

More complex scenarios

- Nested Entity Attestation Tokens
  - To get information from multiple modules within the device
  - Each signed by the corresponding module, using its own key
- Privacy, Confidentiality
  - Each EAT can also have its claims encrypted to ensure confidentiality

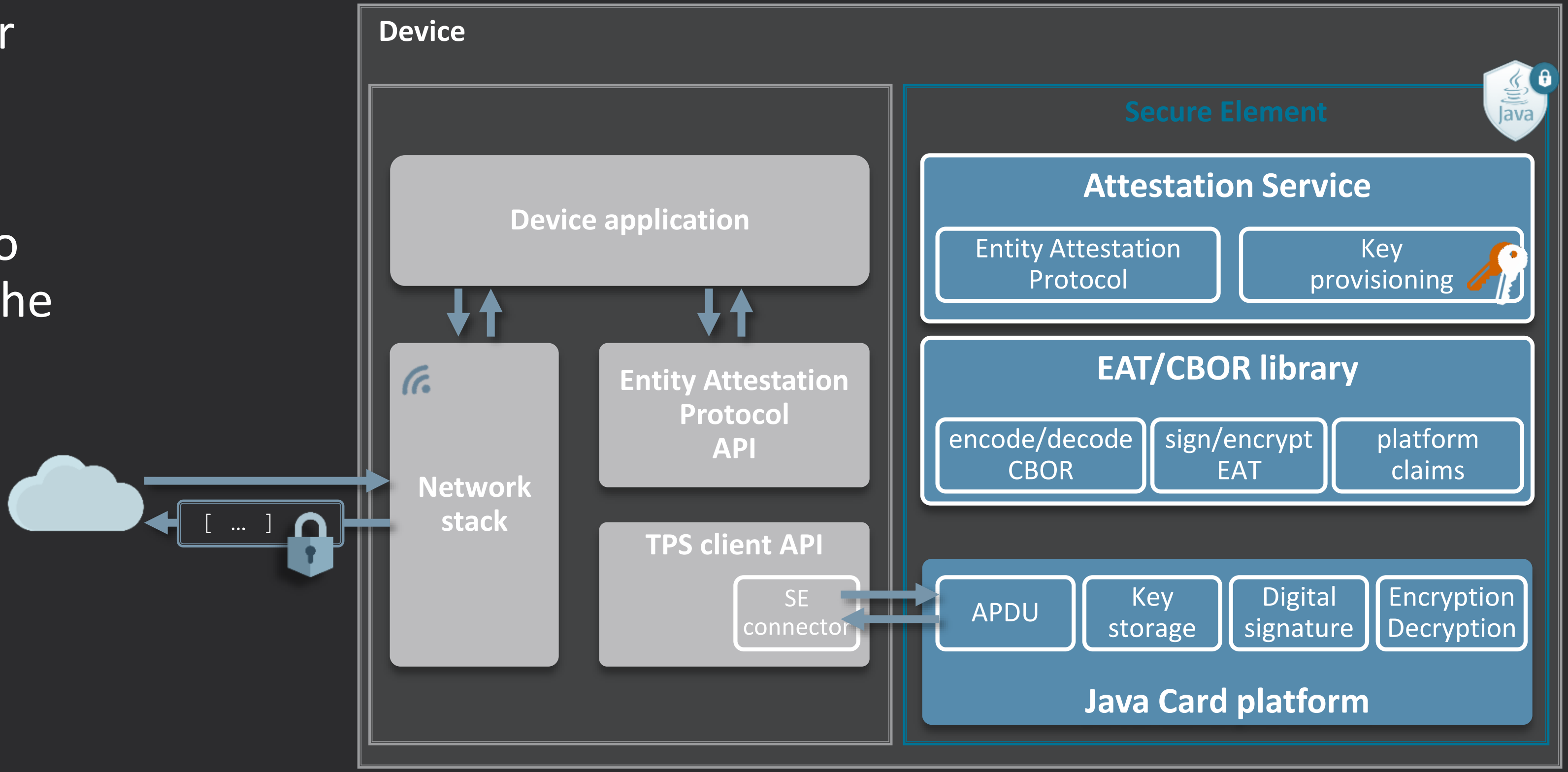


```
[
  header { alg: ECDSA P-256 SHA-256 }
  claims {
    …
    submods:
      [
         header    { … }
         claims    { … }
         signature { … }
      ]
  }
  signature { … }
]
```

Agenda

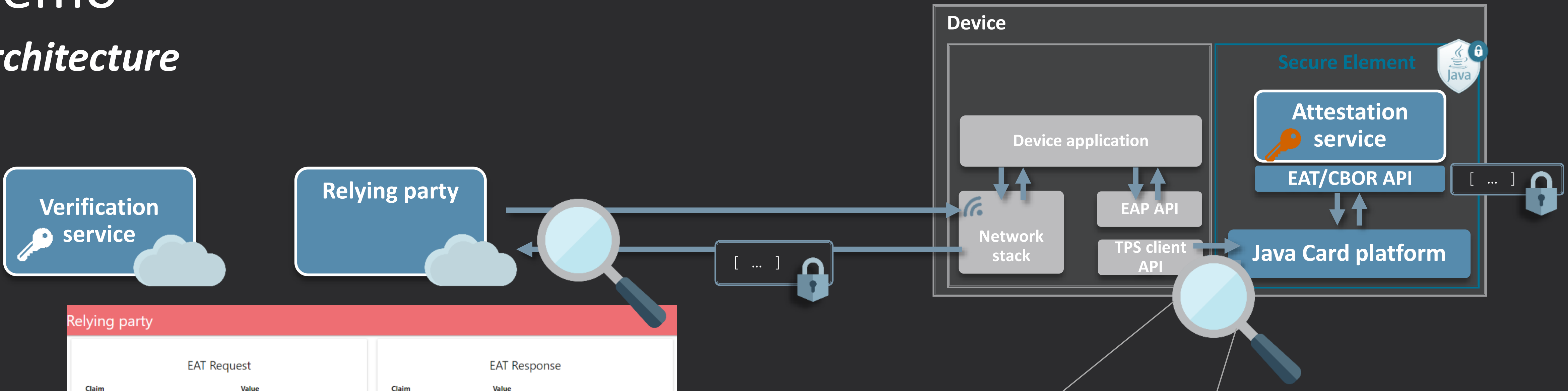
Device Attestation using Java Card

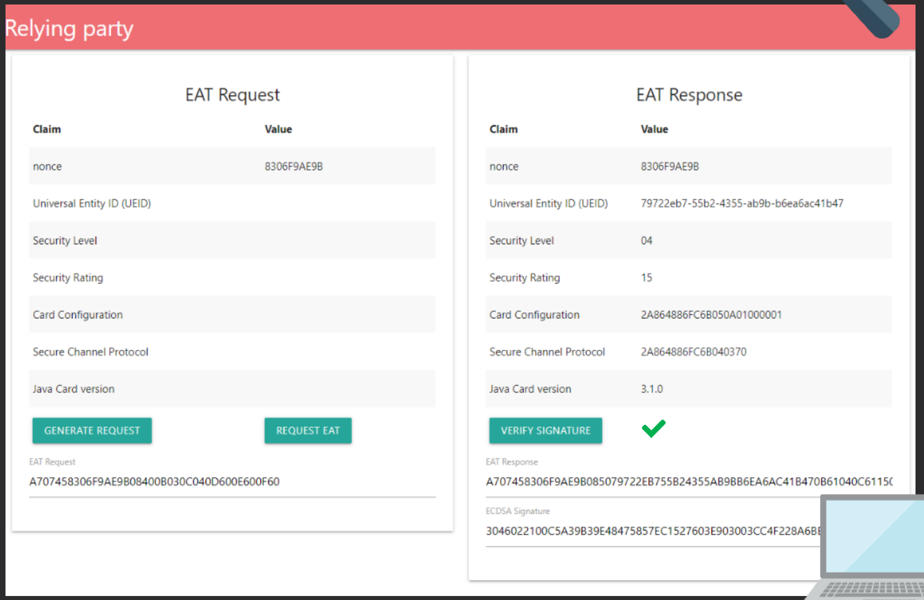
- Communication with remote server using device communication stack
- EAP API used to delegate process to the Attestation Service running in the Secure Element
- Attestation Service is a Java Card Applet
- EAT/CBOR library used to encode, decode and sign attestation tokens



Demo

Architecture




Relying party

EAT Request

Claim	Value
nonce	8306F9AE9B
Universal Entity ID (UEID)	
Security Level	
Security Rating	
Card Configuration	
Secure Channel Protocol	
Java Card version	

GENERATE REQUEST REQUEST EAT

EAT Request
A707458306F9AE9B08400B030C040D600E600F60

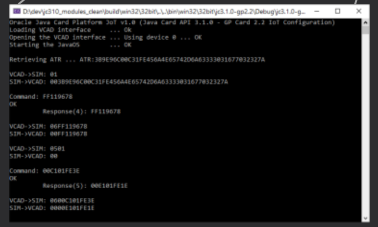
EAT Response

Claim	Value
nonce	8306F9AE9B
Universal Entity ID (UEID)	79722eb7-55b2-4355-ab9b-b6ea6ac41b47
Security Level	04
Security Rating	15
Card Configuration	2A864886FC6B050A01000001
Secure Channel Protocol	2A864886FC6B040370
Java Card version	3.1.0

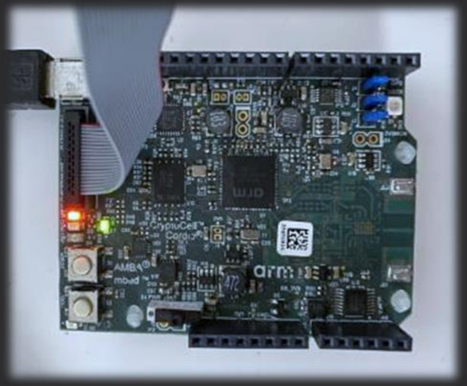
VERIFY SIGNATURE ✓

EAT Response
A707458306F9AE9B085079722EB755B24355AB9BB6EA6AC41B470B61040C6115

ECDSA Signature
3046022100C5A39B39E48475857EC1527603E903003CC4F228A6BE

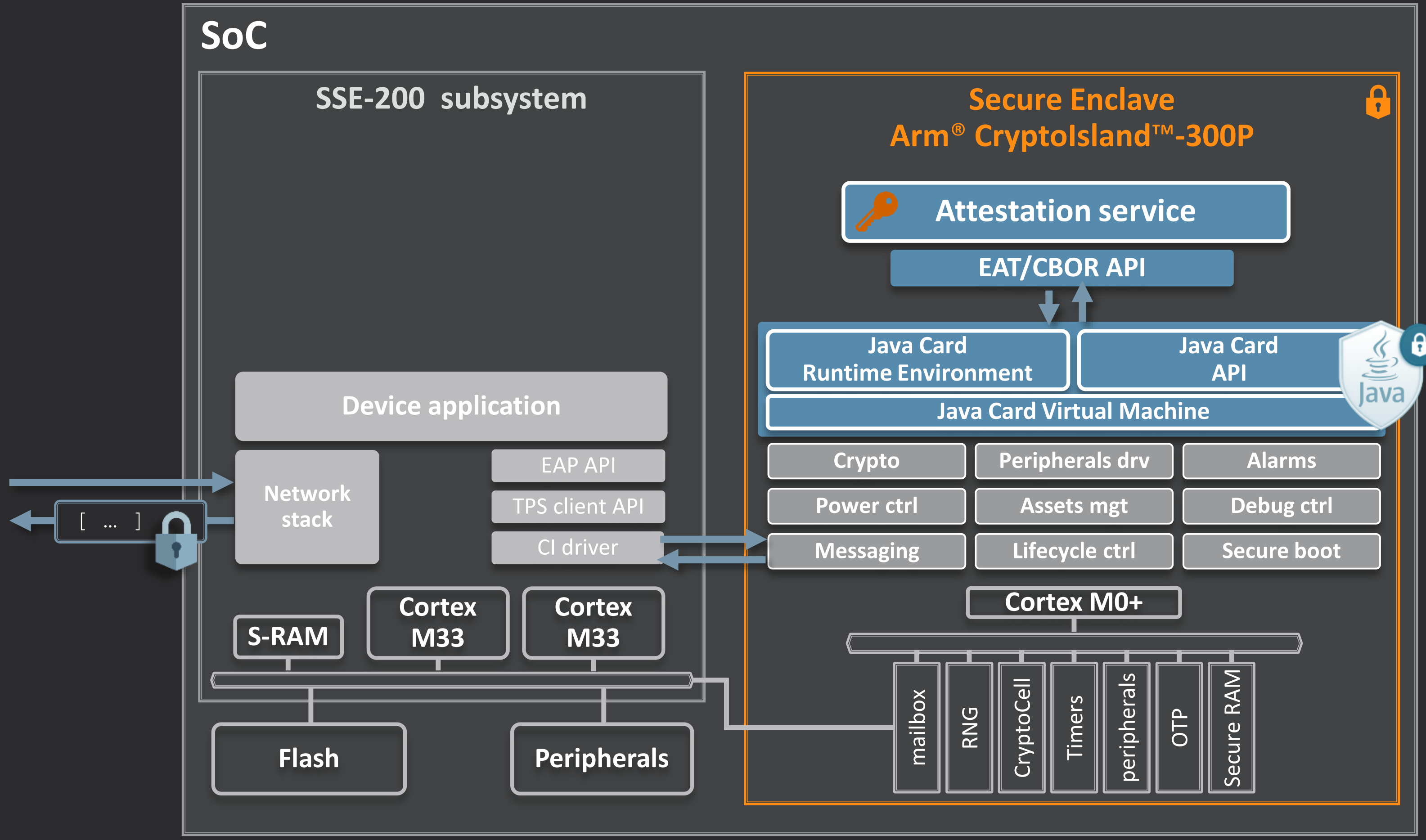

D:\dev\jc310_modules_clean\build\win32\32bit\..\..\bin\win32\32bit\jc3.1.0-gp2.2\Debug\jc3.1.0-g...

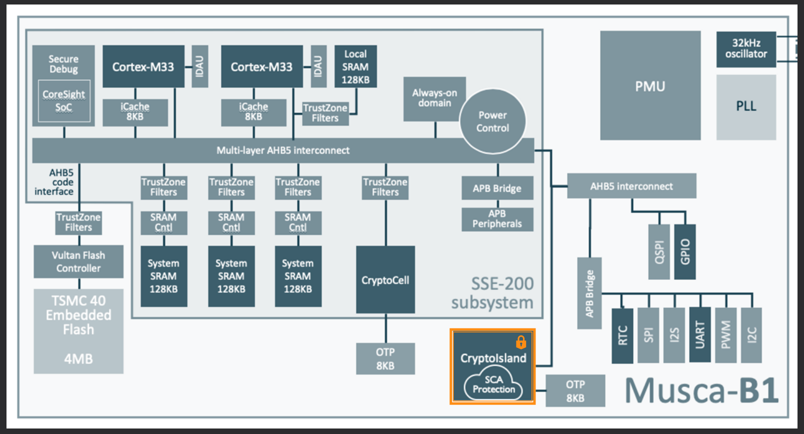
```
Oracle Java Card Platform IoT v1.0 (Java Card API 3.1.0 - GP Card 2.2 IoT Configuration)
Loading VCAD interface     ... Ok
Opening the VCAD interface ... Using device 0 ... OK
Starting the JavaOS        ... OK

Retrieving ATR ... ATR:3B9E96C00C31FE456A4E65742D6A633330316770323 27A

VCAD->SIM: 01
SIM->VCAD: 003B9E96C00C31FE456A4E65742D6A63333031677032327A

Command: FF119678
OK
         Response(4): FF119678

VCAD->SIM: 06FF119678
SIM->VCAD: 00FF119678

VCAD->SIM: 0501
SIM->VCAD: 00

Command: 00C101FE3E
OK
         Response(5): 00E101FE1E

VCAD->SIM: 0600C101FE3E
SIM->VCAD: 0000E101FE1E
```



arm Musca-B1 test Chip Board

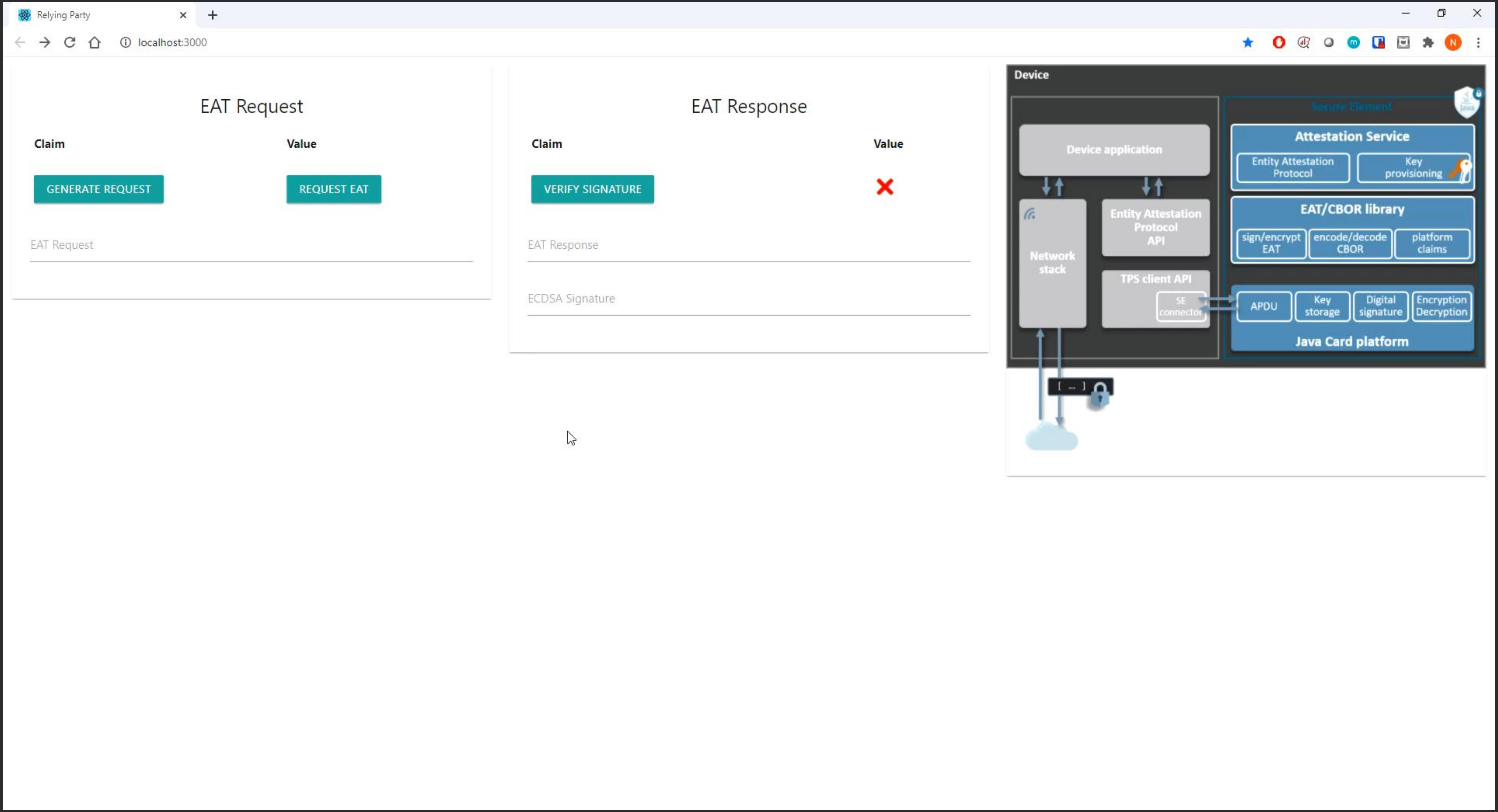
Device





arm

https://developer.arm.com/tools-and-software/development-boards/iot-test-chips-and-boards/musca-b-test-chip-board


Relying Party
localhost:3000
EAT Request
Claim
Value
GENERATE REQUEST
REQUEST EAT
EAT Request
EAT Response
Claim
Value
VERIFY SIGNATURE
EAT Response
ECDSA Signature
Device
Secure Element
Attestation Service
Entity Attestation Protocol
Key provisioning
Device application
EAT/CBOR library
sign/encrypt EAT
encode/decode CBOR
platform claims
Network stack
Entity Attestation Protocol API
TPS client API
SE connector
APDU
Key storage
Digital signature
Encryption Decryption
Java Card platform
[...]

Example of claims used for demo

Based on current IETF draft for EAT: https://www.ietf.org/archive/id/draft-ietf-rats-eat-04.txt

Nonce:	Arbitrary number generated by the relying party
Universal Entity ID:	UEID's identify individual manufactured entities / devices […] UEID's must be universally and globally unique across manufacturers and countries.
Security level:	Describes security environment and countermeasures available on the end-entity / client device where the attestation key reside and the claims originate. {Unrestricted:1, restricted:2, secure-restricted:3, hardware:4}

Based on GP Entity Attestation Protocol draft

Security rating	Provides information about how secure the Entity is. {unknown: 0, basic: 5, substantial: 10, high: 15}
Card configuration	The configuration the Secure Element complies to. "GP Compact IoT Configuration 1.0 with asymmetric crypto": 2A 8648 86FC6B 05 0A 01 00 00 01
Secure Channel Protocol	The Secure Channel Protocol used by Issuer Security Domain. "GP Secure Channel Protocol 03 option i=70": 2A 8648 86FC6B 04 03 70
Java Card Version	3.1.0

Agenda

Device Attestation using Java Card

Benefits

Secure Runtime

- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

Portable

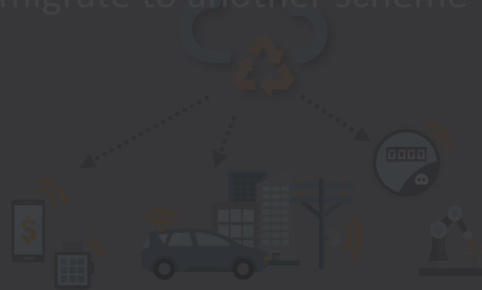
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

Adaptable & Extensible

- To support multiple attestation schemes
- To extend attestation service and include application specific claims

Manageable

- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme



Device Attestation using Java Card

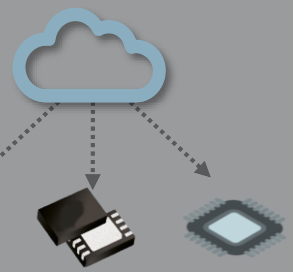
Benefits

Secure Runtime

- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

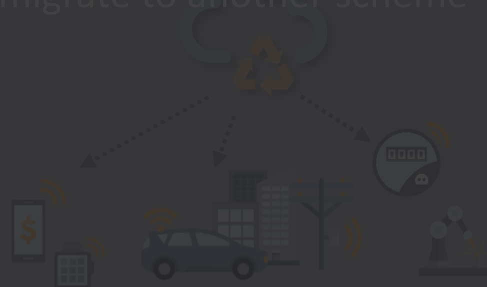


Adaptable & Extensible

- To support multiple attestation schemes
- To extend attestation service and include application specific claims

Manageable

- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme



Device Attestation using Java Card

Benefits

Secure Runtime

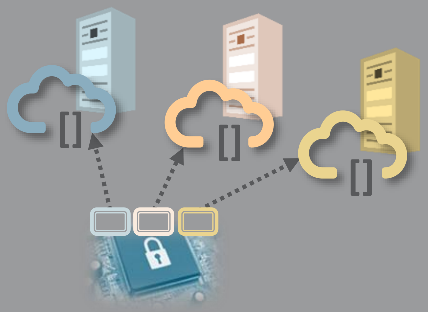
- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

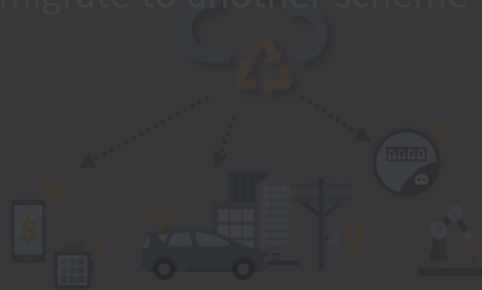
Adaptable & Extensible

- To support multiple attestation schemes
- To extend attestation service and include application specific claims



Manageable

- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme



Device Attestation using Java Card

Benefits

Secure Runtime

- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

Portable

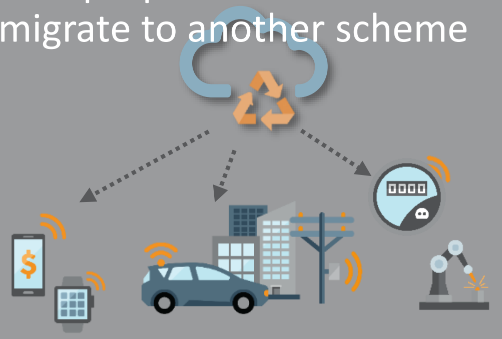
- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost

Adaptable & Extensible

- To support multiple attestation schemes
- To extend attestation service and include application specific claims

Manageable

- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme



Device Attestation using Java Card

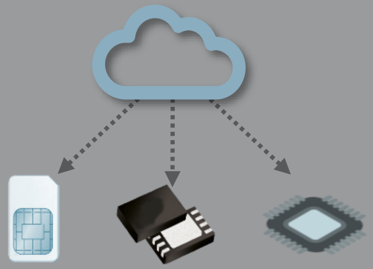
Conclusion

Secure Runtime

- To securely store and manage attestation keys
- To run the complete Attestation service in the Secure Element: retrieve claims, build attestations and sign them.

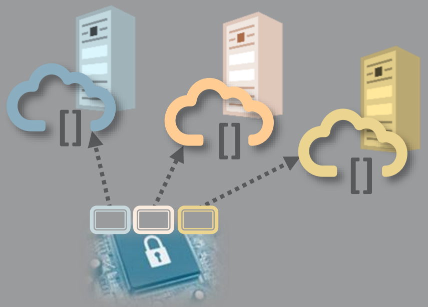
Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



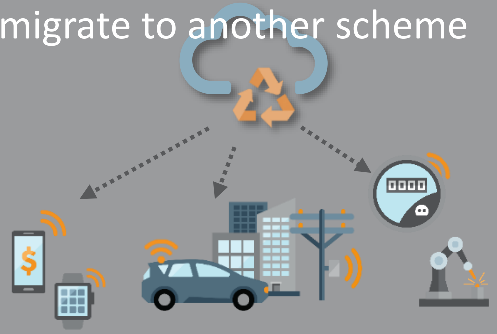
Adaptable & Extensible

- To support multiple attestation schemes
- To extend attestation service and include application specific claims



Manageable

- To update and upgrade the attestation service, remaining compliant with fast evolving security requirements and regulation.
- To repurpose a device or migrate to another scheme



Questions ?

More Information

https://www.oracle.com/java/technologies/java-card-tech.html

[Java Card Platform Specification 3.1]
Latest release of the Java Card specification and the reference for Java Card products.

[Java Card Development Kit Tools]
The Java Card Development Kit Tools are used to convert and verify Java Card applications. The Tools can be used with products based on version 3.1, 3.0.5 and 3.0.4 of the Java Card Specifications.

[Java Card Development Kit Simulator]
The Java Card Development Kit Simulator includes a simulation component and Eclipse plug-in.
Combined with the Java Card Development Kit Tools, it provides a complete, stand-alone development environment.

[Java Card IoT and Security blog]
This Blog covers the latest Java technology for small devices and security in the IoT, mobile, ID and Payment.

[Webcast – Secure Business Runs Java Card]

[Webcast – How to secure IoT Edge with Java Card]

[Webcast: Oracle Java Card 3.1 Boosts Security for IoT Devices at the Edge]

contacts: patrick.van.haver at oracle.com, nicolae.bors at oracle.com