



C2  
PA

Coalition for  
Content Provenance  
and Authenticity

# C2PA Generator Product Security Requirements

C2PA Technical Working Group Conformance Task Force

Version 0.1, 2025-06-02

# Table of Contents

1. Scope .....	1
2. Glossary .....	1
3. Overview .....	8
4. Threats .....	9
5. Security objectives .....	9
6. Definition of security requirements that satisfy the objectives up to different levels. ....	10
7. Footnotes .....	20
Appendix A: Example Implementation Architectures .....	20
Appendix B: Non-normative guidance .....	26
Appendix C: Generator Product Security Architecture Document Template .....	28

# 1. Scope

This document outlines the security requirements for the Generator Product (GP). Security requirements for the Validator Product are out of scope of this document.

## 2. Glossary

### 2.1. Administering Authority

The party that the C2PA [Governing Authority](#) empowers to operate its Conformance Program on its behalf. It recognizes and accredits key conformance roles which agree to participate in the program. The C2PA Conformance Task Force of the Technical Working Group operates in this capacity.

### 2.2. Applicant

An entity that has created a [Generator Product](#) or a [Validator Product](#) and wishes for it to be deemed a [Conforming Product](#) according to the governance framework of the [C2PA Conformance Program](#), and added to the [C2PA Conforming Products List](#).

### 2.3. Applicant's Representative

A natural person who is a duly-authorized employee or agent of the [Applicant](#).

### 2.4. Assertion

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

### 2.5. Asset

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

### 2.6. Assurance Level

An indication to a [Relying Party](#) of the level of confidence that it may have that assertions and claims signed with a given [C2PA Claim Signing Certificate](#) reflect the intended behavior of the [Generator Product](#) instance. A higher Assurance Level allows the [Relying Party](#) to have a greater level of confidence.

The Assurance Level is conveyed through the `c2pa-al` (`1.3.6.1.4.1.62558.3`) X.509 v3 certificate extension in a [C2PA Claim Signing Certificate](#). The value of this extension is an encoded OID value that corresponds to a numeric value no higher than the [Max Assurance Level](#) for a given conforming [Generator Product](#). The OID values corresponding to each Assurance Level are defined in the C2PA `oid.txt` MIB definition file.

The Assurance Level in the [C2PA Claim Signing Certificate](#) that is issued to an instance of a conforming [Generator Product](#) may be lower than the [Max Assurance Level](#) that the [Generator Product](#) is potentially eligible for, based on the [Dynamic Evidence](#) that is presented by that instance of the [Generator Product](#).

## 2.7. Attestation

"The process of providing a digital signature for a set of measurements securely stored in hardware, and then having the requester validate the signature and the set of measurements." [NIST](#)

## 2.8. C2PA Certificate Policy

A document that sets the requirements that SHALL be met by [Certification Authorities](#) (CAs) in the process of issuing digital certificates to [Subscribers](#) that implement C2PA Conforming Products that create [assets](#) with [digital content](#) and [C2PA manifests](#), and the requirements that SHALL be met by the Subscribers in their use of the certificates.

## 2.9. C2PA Claim

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.10. C2PA Claim Signing Certificate

An X.509 certificate issued by one of the [Certification Authorities](#) on the [C2PA Trust List](#) to an instance of the [Conforming Implementer](#)'s conforming [Generator Product](#), and names the [Generator Product](#) as the subject of the certificate.

## 2.11. C2PA Conformance Program

A risk-based governance program intended to hold Applicants who want to demonstrate their conformance to its requirements and then differentiate themselves through C2PA recognition by satisfying program requirements being acknowledged as achieving that level of conformance. It consists of a set of processes, policies, and requirements governing the designation of [Applicant Generator Products](#) or [Validator Products](#) as [Conforming Products](#), and the designation of [Certification Authorities](#) as adhering to the [C2PA Certificate Policy](#), as defined by the C2PA Technical Working Group Conformance Task Force.

Processes include:

- Evaluation of the C2PA-related functions of the [Applicant Generator Product](#) or [Validator Product](#) as adhering to the normative requirements of the C2PA Content Credentials specification
- Evaluation of security attributes of the [Target of Evaluation](#), which includes the [Applicant Generator Product](#) against the [Generator Product Security Requirements](#), which results in assigning it a [Max Assurance Level](#)
- Evaluation of the processes, controls, and technical capabilities of [Certification Authorities](#) as required by the [C2PA Certificate Policy](#)
- Signing of the requisite legal agreements to become a member of the program.

## 2.12. C2PA Conforming Products List

The canonical record of all [Conforming Products](#) that have been deemed conformant according to the stipulations of the C2PA Conformance Program

## 2.13. C2PA Content Credentials

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.14. C2PA Content Credentials Specification

A globally recognized standard for providing digital asset content provenance and authenticity. It is designed to enable global, opt-in, adoption of digital provenance techniques through the creation of a rich ecosystem of digital provenance enabled applications for a wide range of individuals and organizations while meeting appropriate security requirements.

## 2.15. C2PA Governance Framework

A collection of governance documents which defines the C2PA trust ecosystem including roles, requirements and processes used by the C2PA [Governing Authority](#) to achieve greater assurance over the provenance and authenticity of digital asset content.

## 2.16. C2PA Manifest

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.17. C2PA Trust List

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

In the context of the [C2PA Conformance Program](#), a C2PA-managed list of X.509 certificate trust anchors (either root or subordinate [Certification Authorities](#)) that issue certificates to conforming [Generator Products](#) under the [C2PA Certificate Policy](#).

## 2.18. C2PA TSA Trust List

A C2PA-managed list of X.509 certificate trust anchors (either root or subordinate [Certification Authorities](#)) that issue time-stamp signing certificates to Time-Stamping Authorities (TSA).

## 2.19. Certification Authority

A trusted entity that issues, signs, and revokes digital certificates that bind public keys to subscriber identities. CAs are also known as PKI Certificate Authorities because they issue certificates based on public key infrastructure (PKI). These certificates contain credentials that confirm the possession of a private key by an entity, among other verified attributes. [Generator Products](#) sign C2PA Manifests using digital signing credentials issued by CAs.

An entity on the [C2PA Trust List](#) that is trusted by the [C2PA Conformance Program](#) to issue X.509 [C2PA Claim Signing Certificates](#) to instances of conforming [Generator Products](#).

An organization that operates a Certification Authority may also operate a [Time-Stamping Authority](#).

## 2.20. Claim Generator

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.21. Conformance Criteria

A set of normative requirements that the C2PA expects a [Governed Party](#) to demonstrate its conformance as part of the [C2PA Conformance Program](#). This criteria consists of requirements derived from the [C2PA Content Credentials Specification](#) itself, and other ancillary requirements outside of the C2PA Specification including [Generator Product Security Requirements](#) document and requirements in the [C2PA Certificate Policy](#).

## 2.22. Conforming Implementer

An [Applicant](#) who has become a member of the [C2PA Conformance Program](#), and has at least one [Generator Product](#) or [Validator Product](#) in good standing on the [C2PA Conforming Products List](#).

## 2.23. Conforming Product

A [Generator Product](#) or a [Validator Product](#) that has been deemed conformant by the C2PA Conformance Program and added to the [C2PA Conforming Products List](#) with a status of conformant. A [Generator Product](#) that is deemed conformant is also assigned a [Max Assurance Level](#) that is recorded on the C2PA Conforming Products List.

Only instances of conforming [Generator Products](#) are eligible to receive [C2PA Claim Signing Certificates](#) from a [Certification Authority](#) on the [C2PA Trust List](#).

## 2.24. Digital Content

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.25. Dynamic Evidence

Attributes that a [Certification Authority](#) evaluates during automated enrollment for a [C2PA Claim Signing Certificate](#) by an instance of a [Generator Product](#), usually relayed to the Certification Authority in the form of a verifiable hardware-backed artifact, such as a key or platform attestation report.

Dynamic Evidence may result in a particular instance of a Generator Product receiving a certificate of an [Assurance Level](#) that is lower than the [Max Assurance Level](#) that the [Generator Product](#) is potentially eligible for.

## 2.26. Generator Product

The set of software, hardware, and platform configurations created by an [Applicant](#) that work together as a system to produce digital [Assets](#) with C2PA manifests. The asset's active manifest contains assertions made by the Generator Product, and features a claim signed by a certificate where the Generator Product is the subject, about the provenance of the asset.

A Generator Product may integrate [Claim Generator](#) functions monolithically, or rely on a discrete [Claim Generator](#) service available either locally (e.g. on-device), or remotely (e.g. hosted in a cloud

service). The monolithic or discrete [Claim Generator](#) service may be created by the [Applicant](#) or by a different entity.

Because the Generator Product is always the [Signer](#) in the [C2PA Conformance Program](#), and is always the entity listed on the [C2PA Conforming Products List](#), it is accountable for the conformance of the [Assets](#) with C2PA manifests that it generates with the normative requirements of the C2PA Content Credentials Specification, regardless of whether it integrates [Claim Generator](#) functions directly or relies on a discrete service.

## 2.27. Generator Product Security Architecture Document

A filled-out version of the Generator Product Security Architecture Document Template, submitted by the [Applicant](#) to the [C2PA Conformance Program](#) as part of its application for inclusion on the [C2PA Conforming Products List](#).

## 2.28. Generator Product Security Requirements

Security-related implementation requirements for a [Generator Product](#) to achieve a particular [Max Assurance Level](#), detailed in a document of the same name.

## 2.29. Governed Party

An organization which desire to play a recognized role in the C2PA Conformance Program. It applies to the C2PA Conformance Program which requires them to sign a legal agreement and have their product reviewed prior to entering them on the C2PA Trust List or the Conforming Products List. Governed Parties of the C2PA ecosystem are [Certification Authorities](#) and [Applicants](#) that elect to apply and and abide by the C2PA Conformance Program requirements.

## 2.30. Governing Authority

The organization responsible for the trust of the ecosystem. It empowers an [Administering Authority](#) to manage the ecosystem and certifying entities to convey trust. The C2PA is the governing party of its conformance program driven by its Steering Committee.

## 2.31. Hosting Environment

Server-side environment hosting a subset of [Generator Product](#) or [Validator Product](#) mechanisms and functionalities.

## 2.32. Implementation Class - Backend

An implementation architecture for a [Target of Evaluation](#) in which assets, assertions, claims, and claim signatures are generated in one or more [Hosting Environments](#), including those hosted on premises or on commercial cloud service providers.

## 2.33. Implementation Class - Distributed

An implementation architecture for a [Target of Evaluation](#) which is composed of [Edge](#) and [Backend](#) subsystems, where the generation of assets, assertions, claims, and claim signatures is distributed between those subsystems.

## 2.34. Implementation Class - Edge

An implementation architecture for a [Target of Evaluation](#) in which assets, assertions, claims, and claim signatures are generated on an endpoint that operates at the edge of the network, such as:

- Smartphones and smartphone applications
- Laptop and desktop computers
- Fixed-function mirrorless cameras and surveillance cameras
- Portable audio recorders

## 2.35. Manifest Consumer

The number and variety of consumers that rely upon the content provenance and authenticity of digital objects using content credentials are too numerous to capture in this document. In order for Manifest Consumers to consume Content Credentials supported by the C2PA, they MUST use C2PA-approved service providers. In addition, the C2PA Specification cites mandatory requirements for Manifest Consumers. While the C2PA mandates these requirements and discloses them in the Specification, it does not hold Manifest Consumers accountable to conform to these requirements within its governance framework.

## 2.36. Max Assurance Level

A numeric designation, chosen at the discretion of the [C2PA Conformance Program](#), based on evaluating the security functions, properties, and attributes of an [Applicant Generator Product](#) against the [Generator Product Security Requirements](#) defined by the [C2PA Conformance Program](#).

## 2.37. Registration Authority

An entity authorized by the [Certification Authority](#) to collect, verify, and submit information provided by [Applicants](#) and/or [Subscribers](#) which is to be entered into public key certificates. The term RA refers to hardware, software, and individuals that collectively perform this function, including tasks such as validating platform attestations and presence of potential Subscriber implementations on the C2PA Conforming Products List. The RA operates under the CA's authority and adheres to the guidelines set forth in the [C2PA Certificate Policy](#).

## 2.38. Reliable Method of Communication

A method of communication, such as a postal/courier delivery address, telephone number, or email address, that was verified using a source other than the Applicant's Representative.

## 2.39. Relying Party

An entity that evaluates the trustworthiness of [Assertions](#) made by a [Signer](#) in a C2PA [Asset](#), based on the [Signer's](#) identity and the [Assurance Level](#) encoded into the [C2PA Claim Signing Certificate](#).

## 2.40. Rich Execution Environment

Refer to [NIST definition](#). Abbreviated as REE.



## 2.41. Root of Trust

Refer to [NIST definition](#). Abbreviated as RoT.

## 2.42. Security Incident

"An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies." [NIST](#)

## 2.43. Signer

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

In the C2PA Trust Model, the [Assertions](#) enumerated in the `created_assertions` object of the [C2PA Claim](#) are attributed to the Signer.

In the context of the [C2PA Conformance Program](#), an instance of the conforming [Generator Product](#) listed on the [C2PA Conforming Products List](#) is always the Signer.

## 2.44. Static Evidence

Attributes of the [Generator Product Target of Evaluation](#) that are documented in the [Generator Product Security Requirements](#) document which the [Administering Authority](#) evaluates during its assessment of the [Applicant's Generator Product](#), in order to assign a [Max Assurance Level](#).

## 2.45. Subscriber

An Applicant that has become a customer of one of the [Certification Authorities](#) on the [C2PA Trust List](#), and is eligible to receive [C2PA Claim Signing Certificates](#) for use by instances of their conforming [Generator Product](#).

## 2.46. Target of Evaluation

The system which is evaluated by the [C2PA Conformance Program](#) for its functional correctness and the security of its implementation. It consists of the sum total of the [Generator Product](#) or [Validator Product](#) created by an [Applicant](#), **and** the subsystems that it relies on to produce or validate [Assets](#) with [C2PA Manifests](#). Those subsystems need not be created by the [Applicant](#), but are necessary for the proper operation of the [Generator Product](#) or the [Validator Product](#).

The functional capabilities and security properties of those subsystems contribute to the overall security of the [Applicant's](#) product, and are thus considered by the [C2PA Conformance Program](#) when assigning an [Assurance Level](#) to a conforming [Generator Product](#).

Targets of Evaluation can have [Edge](#), [Backend](#), or [Distributed](#) implementation architectures.

## 2.47. Time-Stamping Authority

A server that provides electronic certification and trust services by creating a hash of a document or

digital information. The hash verifies the date and time of the document's creation or last modification, and acts as an independent witness to prove that the document has not changed since it was signed. This is similar to how a notary acts for online documents. TSAs, that are part of Applicant Certification Authorities, that want to be recognized as issuing digital signing credentials approved by the C2PA, must satisfy the requirements of the program to be considered approved and designated as such within its governance records.

## 2.48. Trusted Execution Environment

Refer to [NIST definition](#). Abbreviated as TEE.

## 2.49. Validator

Refer to Section 2, "Glossary", of the C2PA Content Credentials specification.

## 2.50. Validator Product

The set of software, hardware, and platform configurations created by an [Applicant](#) that work together as a system to validate digital [Assets](#) with C2PA manifests.

A Validator Product may integrate [Validator](#) functions monolithically, or rely on a discrete [Validator](#) service available either locally (e.g. on-device), or remotely (e.g. hosted in a cloud service). The monolithic or discrete [Validator](#) service may be created by the [Applicant](#) or by a different entity.

Because the Validator Product is always the entity listed on the [C2PA Conforming Products List](#), it is accountable for producing correct validation results in adherence with the normative requirements of the C2PA Content Credentials Specification, regardless of whether it integrates [Validator](#) functions directly or relies on a discrete service.

# 3. Overview

A structured approach to define security requirements for Generator Product consists of the following steps:

1. Identification of threats relevant to C2PA technology that need to be addressed by the Generator Product Target of Evaluation (TOE).
2. Translation of the threats into security objectives.
3. Definition of security requirements that satisfy the security objectives up to different levels.

Meeting security requirements of a specific level grants a respective Max Assurance Value on the C2PA Conforming Product List.

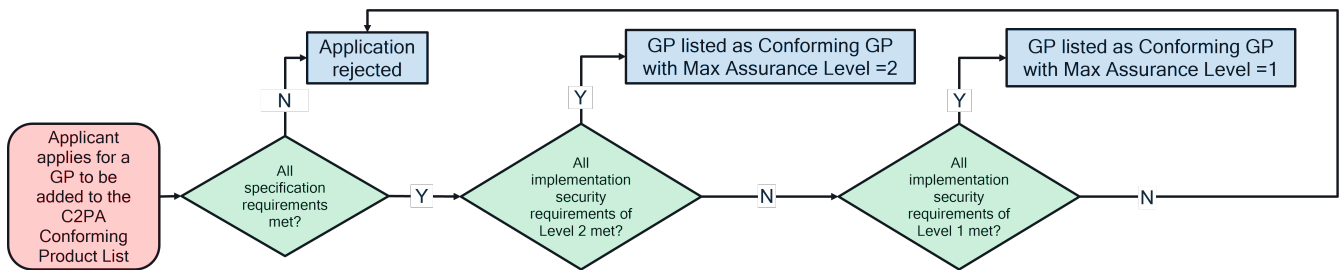


Figure 1. Flow Chart for Determining Maximum Assurance Level of Generator Product on the Conforming Product List.

## 4. Threats

This section identifies threats that are relevant to C2PA technology that need to be addressed by the Generator Product Target of Evaluation (TOE).

### 4.1. Threats identified within C2PA Security Considerations

As defined in the [C2PA Security Considerations](#) document.

- 4.3.2.2. Threat: Spoofing signed C2PA metadata via stolen key.
- 4.3.2.3. Threat: Spoofing signed C2PA metadata via misuse of Claim Generator.
- 4.3.3.4. Threat: Exploitation of the hosting environment.
- 4.3.3.5. Threat: Interception and/or modification of traffic between two trusted sources.

### 4.2. Additional identified threats

- Threat: Impersonating conforming GP during automated certificate enrollment.
  - Automated certificate enrollment flows can allow conforming GP instances to enroll for certificates or rotate certificates on an on-going basis without human intervention. An attacker may attempt to exploit the automated enrollment process with the Certification Authority, and impersonate the conforming GP to obtain a valid claim signing certificate for an attacker-owned key pair.
- Threat: Tampering with asset and/or assertions at generation.

## 5. Security objectives

The following table translates the threats defined above into security objectives.

Threat
Security Objective
T.1 - Impersonating a conforming GP during automated certificate enrollment.
O.1 - Conforming GP instance provides proof of its eligibility during automated certificate enrollment.
T.2 - Spoofing signed C2PA metadata via a stolen key.
O.2 - GP TOE protects the confidentiality of the signing key.

Threat
T.3 - Spoofing signed C2PA metadata via misuse of Claim Generator.
O.3 - GP TOE protects the Claim Generator from exploits, misconfiguration and misuse.
T.4 - Tampering with asset and/or assertions at generation.
O.4 - GP TOE protects the asset and/or assertions from being tampered with at generation.
T.5 Interception and/or modification of traffic between two trusted sources.
O.5 GP TOE protects the traffic between subsystems and components of those subsystems from being intercepted and/or modified.
T.6 - Exploitation of the hosting environment.
O.6 - Hosting environment is protected from exploit, misconfiguration and misuse.

## 6. Definition of security requirements that satisfy the objectives up to different levels.

### NOTE

The requirements apply to all Implementation Classes (**Edge**, **Backend**, and **Distributed**) unless otherwise noted.

### NOTE

Where applicable, evidence SHALL be provided in both forms static and dynamic, rather than choosing between the two evidence forms.

### 6.1. O.1 - Conforming GP instance provides proof of its eligibility during automated certificate enrollment.

The following requirements are only applicable if conforming GP instances rely on automated certificate enrollment for initial certificate issuance or rotation.

#### 6.1.1. Level 1

##### Requirements

- GP TOE SHALL implement the secure authentication method required by the Certification Authority as part of an automated certificate enrollment process, which may rely on one of the following:
  - Shared secret/passphrase
  - Client certificate
  - Username/password
  - Challenge-response
  - Symmetric key MAC
- For **Edge** Implementation Class, the GP TOE binary/binaries SHALL NOT include authentication secrets

##### Static Evidence

- Applicant SHALL include details of the GP TOE's automated certificate enrollment authentication

method, including its design for managing authentication secrets, in the GP Security Architecture Document.

### Dynamic Evidence

1. The conforming GP instance SHALL authenticate with the CA using its secure credentials during automated certificate enrollment.

## 6.1.2. Level 2

### Requirements

In addition to the requirements defined for Level 1, Level 2 requires the following:

1. GP TOE SHALL be capable of producing or deriving verifiable artifacts backed by a hardware [Root of Trust](#), such as attestations or hardware-derived credentials, from its underlying platform, confirming the GP binary/binaries via package names, hashes, code signing certificates, other digital certificates, or a combination of the above.

A non-exhaustive list of examples of integrity attestation methods is available in [Integrity attestation methods](#)

### Static Evidence

1. Applicant SHALL include details of the GP TOE method for producing or deriving verifiable, hardware-backed artifacts confirming the identity of the GP binary/binaries in the GP Security Architecture Document.

### Dynamic Evidence

1. The conforming GP instances SHALL present the verifiable, hardware-backed artifacts for evaluation by the CA during automated certificate enrollment.

## 6.2. 0.2 - GP TOE protects the confidentiality of the signing key.

### 6.2.1. Level 1

### Requirements

1. GP TOE SHALL store the claim signing key in an encrypted form, using best current practice for encryption algorithm and key length<sup>1</sup>. This requirement does not prohibit the GP from having access to the plaintext claim signing key material once loaded in its memory space.
2. GP SHALL control access to the signing key in decrypted form, following the principle of [least privilege](#). Access SHALL be restricted to actors that have appropriate permissions, and those time-bound permissions are only granted to actors who have a justifiable requirement to access the key at that time.
3. GP TOE SHALL be capable of rotating the claim signing key.

Additional requirements for **Distributed** and **Backend** Implementation Classes:

4. The usage of the **Edge** subsystem authentication key (API Key) SHALL only be for the purposes of

limiting access to the **Backend** subsystem.

5. **Edge** and **Backend** subsystems SHALL be mutually authenticated; the role(s) of each subsystem across all communication channels SHALL be appropriate role(s), and SHALL be authenticated by the other.
6. The remote claim signing **Backend** subsystem of the GP TOE SHALL securely authenticate the calling client, positively confirming that the calling client is a valid instance of the **Edge** subsystem of the GP TOE, before signing a claim with the claim signing key.

Authentication methods may include: shared secret/passphrase; client certificate; username/password; challenge-response; symmetric key MAC.

### Static Evidence

1. Applicant SHALL include the following in the GP Security Architecture Document
  - a. Details of the claim signing key access controls (including encryption) in place that are designed to prevent unauthorized access
  - b. Details of the key rotation process
  - c. For **Distributed** and **Backend** Implementation Classes, documentation of the method for mutual authentication and role validation between the subsystems.

### Dynamic Evidence

No stipulation.

## 6.2.2. Level 2

### Requirements

In addition to the requirements defined for Level 1, Level 2 requires the following:

1. The GP TOE SHALL generate, store, and use the claim signing key within an environment with a higher privilege level than the privilege level of the Claim Generator execution environment, for example, a local platform keystore service or a hosted Key Management Service (KMS). The key management environment SHALL be one that has the following properties:
  - a. The key management environment restricts access to and usage of the key to authenticated callers
  - b. The key management environment sequesters the private key material such that the Claim Generator never has access to raw private key material in its memory space
  - c. The key management environment uses hardware-derived wrapping key(s) to store the claim signing key, and
  - d. One of the following:
    - i. The key management environment is capable of producing or deriving a verifiable artifact backed by a hardware Root of Trust (e.g. through attestation) confirming its possession of the claim signing private key, or
    - ii. An accredited, independent 3rd party auditor certifies that the claim signing key is stored in such an environment

A non-exhaustive list of examples of key management environments is available in [Key management environments](#)

A non-exhaustive list of examples of accepted certification schemes are listed in [Accepted certification schemes](#)

Additional requirements for **Distributed** and **Backend** Implementation Classes:

2. The subsystems (**Edge** and **Backend**) SHALL be capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust, such as attestations or hardware-derived credentials, from their underlying platforms
3. The subsystems SHALL be capable of decoding and validating the hardware-backed artifacts produced by their counterpart subsystem
4. The chosen method to produce or derive hardware-backed artifacts SHALL be one that allows the authentication of the counterpart subsystem via package names, hashes, code signing certificates, other digital certificates, or a combination of the above.
5. Before a calling client requests the signing of a claim, the **Backend** subsystem SHALL request then validate the verifiable, artifact backed by a hardware Root of Trust, resulting in a validated verdict.
6. If the the validated verdict does not positively confirm the calling client as a valid instance of the **Edge** subsystem, the **Backend** subsystem SHALL NOT sign the claim.

A non-exhaustive list of examples of integrity attestation methods is available in [Integrity attestation methods](#)

### Static Evidence

1. Applicant SHALL include the following in the GP Security Architecture Document:
  - a. Details of the key management environment that the GP TOE uses, and its security properties. This may include commonly-accepted security certifications for the key management environment, and if applicable, certification by an accredited, independent 3rd party auditor certifying that the claim signing key is stored in such an environment.
  - b. Details of the key rotation process
  - c. If the Implementation Class is **Distributed**, or **Backend** details of the method for producing or deriving verifiable artifacts backed by a hardware Root of Trust by the subsystems.

### Dynamic Evidence

1. The conforming GP instance SHALL present a verifiable artifact backed by a hardware Root of Trust confirming its possession of the claim signing private key for evaluation by the CA during automated certificate enrollment.

## 6.3. 0.3 - GP TOE protects the Claim Generator from exploits, misconfiguration and misuse.

### 6.3.1. Level 1

#### Requirements

1. Applicant SHALL ensure a Software Composition Analysis (SCA) or Software Bill of Materials (SBOM) analysis is performed to detect vulnerabilities from the NIST National Vulnerability Database (NVD, <https://nvd.nist.gov/>) in the Claim Generator.

2. Applicant SHALL ensure that fixes or other mitigations are applied to any Claim Generator security vulnerabilities detected with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater within 90 days of detection.

Examples of such tools are listed in [SCA and SBOM dependency vulnerability scanning tools](#)

### Static Evidence

1. Applicant SHALL document the following in the GP Security Architecture Document:
  - a. The SCA/SBOM dependency vulnerability scanning tools used during the Claim Generator build or integration process.
  - b. The process by which the build and deployment pipeline prevents the release, more than a 90 days after detection, of the Claim Generator with known **CRITICAL** or **HIGH** severity vulnerabilities.

### Dynamic Evidence

No stipulation.

## 6.3.2. Level 2

### Requirements

In addition to Level 1 requirements, Level 2 requires the following:

1. The Applicant SHALL ensure that the Claim Generator and its execution environment enable [basic exploit countermeasures](#).
2. The Applicant SHALL ensure [static analysis](#) of the Claim Generator and the platform/environment that the Claim Generator is built on.
3. The Applicant SHALL ensure that the Claim Generator is built on a platform/environment where access control and image authentication for the Claim Generator binaries and the platform/environment are enforced by (if available) an environment with a privilege level higher than the privilege level of the platform/environment.
4. The Applicant SHALL implement at least one of the following methods for ensuring the recency of security patches in the Claim Generator:
  - a. The Applicant SHALL ensure that the Claim Generator is capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust from its underlying platform, that provide evidence of how recently security patches were applied to this instance of the Claim Generator; or
  - b. The Applicant SHALL ensure that the Claim Generator is capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust from its underlying platform, which attest to the revision of the instance of the Claim Generator (e.g. version identifier, branch identifier, or commit identifier). If Applicant chooses this method, then Applicant SHALL design and execute a process by which Applicant SHALL:
    - i. Alert the CA of the minimum revision of the Claim Generator integrated with its GP that is eligible to enroll for certificates.
    - ii. Alert the CA of which revisions of the Claim Generator integrated with its GP have security vulnerabilities with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater and therefore make the integrating GP instance ineligible to enroll for certificates past 90 days



after detection

The following applies to Claim Generators that rely on external inputs:

5. The Applicant SHALL ensure that the Claim Generator enforces access control on the ingress points for external inputs
6. The Applicant SHALL ensure that the Claim Generator validates external inputs

### **Static Evidence**

1. Applicant SHALL include the following in the GP Security Architecture Document:
  - a. Documentation of Claim Generator build scripts and build flags confirming enablement of countermeasures
  - b. Countermeasures functional test report.
  - c. Static analysis tools used
  - d. If applicable, access control methods
  - e. If applicable, binary image authentication methods
  - f. If applicable, external input validation methods
  - g. If applicable, access control lists for external input ingress points
  - h. Details of the method that the Applicant has chosen to provide information about the recency of security patches for the Claim Generator integrated with the instance of the GP through verifiable artifacts backed by a hardware Root of Trust.

### **Dynamic Evidence**

1. The conforming GP instance SHALL present verifiable artifacts backed by a hardware Root of Trust for evaluation by the CA during automated certificate enrollment confirming (depending on which method the Applicant has chosen):
  - a. The recency of application of security patches of the Claim Generator integrated with the instance of the GP, is no less than the latest patch or version required to fix or otherwise mitigate any vulnerabilities with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater; or
  - b. The revision of the Claim Generator (e.g. a version identifier, a branch identifier, or a commit identifier) integrated with the instance of the GP, which the CA can compare against a list of allowed revisions

## **6.4. O.4 - GP TOE protects the asset and assertions from being tampered with at generation.**

### **6.4.1. Level 1**

Below requirements apply to all software in GP TOE that processes/modifies the [Digital Content](#) and/or assertions:

#### **Requirements**

1. Applicant SHALL ensure a Software Composition Analysis (SCA) or Software Bill of Materials

(SBOM) analysis is performed to detect vulnerabilities from the NIST National Vulnerability Database (NVD, <https://nvd.nist.gov/>) in all such software.

2. Applicant SHALL ensure that fixes or mitigations are applied to any Claim Generator security vulnerabilities detected with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater within 90 days of detection.

Examples of such tools are listed in [SCA and SBOM dependency vulnerability scanning tools](#)

## 6.4.2. Level 2

Below requirements apply to all software in GP TOE that processes/modifies the [Digital Content](#) and/or assertions:

### Requirements

1. All such software SHALL enable [basic exploit countermeasures](#).
2. The Applicant SHALL ensure [static analysis](#) on all such software.
3. Authentication of binary images of all such software SHALL be enforced by (if available) an environment with a privilege level higher than the privilege level of such software<sup>8</sup>.
4. The Applicant SHALL ensure that the GP TOE SHALL is capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust, such as attestations or hardware-derived credentials, confirming that all such software has been authenticated
5. The Applicant SHALL implement at least one of the following methods for ensuring the recency of security patches in the GP TOE:
  - a. The Applicant SHALL ensure that the GP TOE is capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust from its underlying platform, that provide evidence of how recently security patches were applied to all such software in the TOE.
  - b. The Applicant SHALL ensure that the GP TOE is capable of producing or deriving verifiable artifacts backed by a hardware Root of Trust from its underlying platform, which attest to the revision(s) of the GP TOE software (e.g. version identifier, branch identifier, or commit identifier). If Applicant chooses this method, then Applicant SHALL design and execute a process by which Applicant SHALL:
    - i. Alert the CA of the minimum revision of the GP TOE software is eligible to enroll for certificates.
    - ii. Alert the CA of which revisions of the GP TOE software have security vulnerabilities with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater, and therefore make the integrating GP instance ineligible to enroll for certificates past 90 days after detection.
6. GP TOE SHALL provide the following protection for the asset and all assertions, while these are processed by such software by ensuring usage of access control mechanisms enforced by (if available) an environment with privilege level higher than the privilege of such software:
  - a. isolation of the source processes and/or threads
  - b. protection of the inter-process communication channels
  - c. protection of memory.

## Static Evidence

1. Applicant SHALL include the following in the GP Security Architecture Document:
  - a. If applicable, description of image authentication methods of source and its execution environment for all sources of assets and assertions implemented in software.
  - b. Details of the method that the Applicant has chosen to provide information about the recency of security patches for the GP TOE through verifiable artifacts backed by a hardware Root of Trust.
  - c. Documentation of build scripts and build flags confirming enablement of countermeasures
  - d. Countermeasures functional test report
  - e. Static analysis tools used
  - f. If applicable, binary image authentication methods
  - g. Confirmation of support of isolation of the source processes and/or threads and protection of the inter-process communication channels by the kernel or operating system in use. This can be achieved by demonstrating that:
    - i. the processes and/or threads run under a unique operating system UID or user account from other processes on the device,
    - ii. forms of inter-process communication (Android broadcasters and receivers, IPC channels, etc.) are limited to only those necessary for the application to function and that ACLs limit the processes and/or threads that connect to them.

## Dynamic Evidence

1. The conforming GP instance SHALL present verifiable artifacts backed by a hardware Root of Trust for evaluation by the CA during automated certificate enrollment confirming the following properties of all software in the GP TOE that processes/modifies the [Digital Content](#) and/or assertions:
  - a. That all such software has been authenticated
  - b. Depending on the method for confirming recency of security patches that the Applicant has chosen:
    - i. The recency of application of security patches of the GP TOE software, is no less than the latest patch or version required to fix or otherwise mitigate any vulnerabilities with a CRITICAL or HIGH severity ratings in the NIST Common Vulnerability Scoring System (CVSS, <https://nvd.nist.gov/vuln-metrics/cvss>) version 3 or greater; or
    - ii. The revision(s) of the GP TOE software (e.g. a version identifier, a branch identifier, or a commit identifier), which the CA can compare against a list of allowed revisions

## 6.5. 0.5 - GP TOE protects the traffic between subsystems and components of those subsystems from being intercepted and/or modified.

### 6.5.1. Level 1

For **Distributed** and **Backend** Implementation Classes:

## Requirements

1. Network communication channels between the subsystems (**Edge** and/or **Backend**) SHALL be encrypted

using TLS v1.3 (or higher) or an equivalent protocol.

## Static Evidence

1. Applicant SHALL include the following in the GP Security Architecture Document:
  - a. Documentation of the TLS versions or equivalent protocols in use and the supported cryptographic protocols for network communication between the subsystems (**Edge** and/or **Backend**).

## Dynamic Evidence

No stipulation

## 6.5.2. Level 2

### Requirements

GP TOE SHALL provide the following kernel/operating system level protection for asset and all assertions transmitted within subsystems (**Edge** and/or **Backend**):

- a. isolation of the source processes and/or threads related to asset and assertion generation
- b. protection of the inter-process communication channels used to transmit assets or assertions between processes and/or threads

### Static Evidence

1. Applicant SHALL include the following in the GP Security Architecture Document:
  - a. Confirmation of support of isolation of the source processes and/or threads and protection of the inter-process communication channels by the kernel or operating system in use. This can be achieved by demonstrating that:
    - i. the processes and/or threads run under a unique operating system UID or user account from other processes on the device,
    - ii. forms of inter-process communication (Android broadcasters and receivers, IPC channels, etc.) are limited to only those necessary for the application to function and that ACLs limit the processes and/or threads that connect to them.

### Dynamic Evidence

No stipulation.

## 6.6. O.6 - Hosting environment is protected from exploit, misconfiguration and misuse.

This security objective apply only to the **Distributed** and **Backend** Implementation classes.

## 6.6.1. Level 1

### Requirements

1. Resources used for asset and/or assertion generation SHALL protected with an Identity and Access Management (IAM) system that implements Role-Based Access Control (RBAC) or a similar access control model.
2. The Applicant SHALL ensure vulnerability scanning and/or security review is performed for software dependencies and API surfaces, and ensure identified vulnerabilities SHALL be patched, fixed, or mitigated in a timely manner <sup>9</sup>. The process SHALL, at least cover vulnerabilities applicable to the implementation, if any, from: ..Open Worldwide Application Security Project (OWASP) (<https://owasp.org/www-project-top-ten/>) top 10 web application vulnerabilities.
3. **Basic exploit countermeasures** SHALL be applied. Updates and security patches for the operating system and relevant software SHALL be applied in a timely manner <sup>9</sup>.

### Static Evidence

Applicant SHALL include the following in the GP Security Architecture Document:

1. Description of IAM system employed <sup>10</sup> and coverage of security boundaries (e.g., virtual machines, cloud storage repositories) related to asset and claim generation.
2. Description of access policies for human and non-human principal (e.g., service accounts or production identities)
3. Description of IAM policies showing access for main cloud resources (e.g., VM instances, cloud storage buckets).
4. Description of the process used by the Applicant to ensure vulnerability scanning and/or security review is performed for software dependencies and API surfaces.
5. Description of the process used by the Applicant to ensure vulnerabilities are fixed or mitigated in a timely manner <sup>9</sup>.

### Dynamic Evidence

No stipulation.

## 6.6.2. Level 2

### Requirements

In addition to the requirements defined for Level 1, Level 2 requires the following:

1. Audit logging SHALL be enabled and monitored for security-relevant events (e.g., human access).
2. A Host-based Intrusion Detection System (HIDS) OR a distributed system that provides similar functionality SHALL be deployed for monitoring system integrity and detecting suspicious activities.
3. Network segmentation SHALL be implemented to isolate the hosting environment.

### Static Evidence

Applicant SHALL include the following in the GP Security Architecture Document:

1. Description of audit logging system, event logging policies, and monitoring rules

2. Description of HIDS software or similar systems used
3. Description of the network architecture and configuration, including a description of network isolation boundaries
4. Report showing audit logging is enabled
5. Report showing active operation of a HIDS or similar systems
6. Report showing active network isolation operation covering the application's network infrastructure

### Dynamic Evidence

No stipulation.

## 7. Footnotes

<sup>1</sup> Catalogues of allowed cryptographic algorithms: NIST Cryptographic Standards and Guidelines (<https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines>); ENISA ECCG Agreed Cryptographic Mechanisms ([https://certification.enisa.europa.eu/publications/eucc-guidelines-cryptography\\_en](https://certification.enisa.europa.eu/publications/eucc-guidelines-cryptography_en)).

<sup>2</sup> Examples include key management within Trusted Execution Environment running at privilege level higher than Non-Secure World within ARM architecture.

<sup>3</sup> Examples include AWS Key Management Service, AWS CloudHSM, GCP Cloud KMS, Azure Key Vault, HashiCorp Vault

<sup>4</sup> See <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:20004:ed-2:v1:en> for more information on AVA\_VAN levels

<sup>5</sup> ASLR, stack canaries, guard pages, DEP, Safe Heap, NX.

<sup>6</sup> Acceptable static analysis tools: ...<sup>7</sup> CFI, PAC, MTE and sanitizers.

<sup>8</sup> Examples include: a) authentication of a source application by Android Bootloader using Android Verified Boot Process; b) authentication of source trusted virtual machine by a hypervisor.

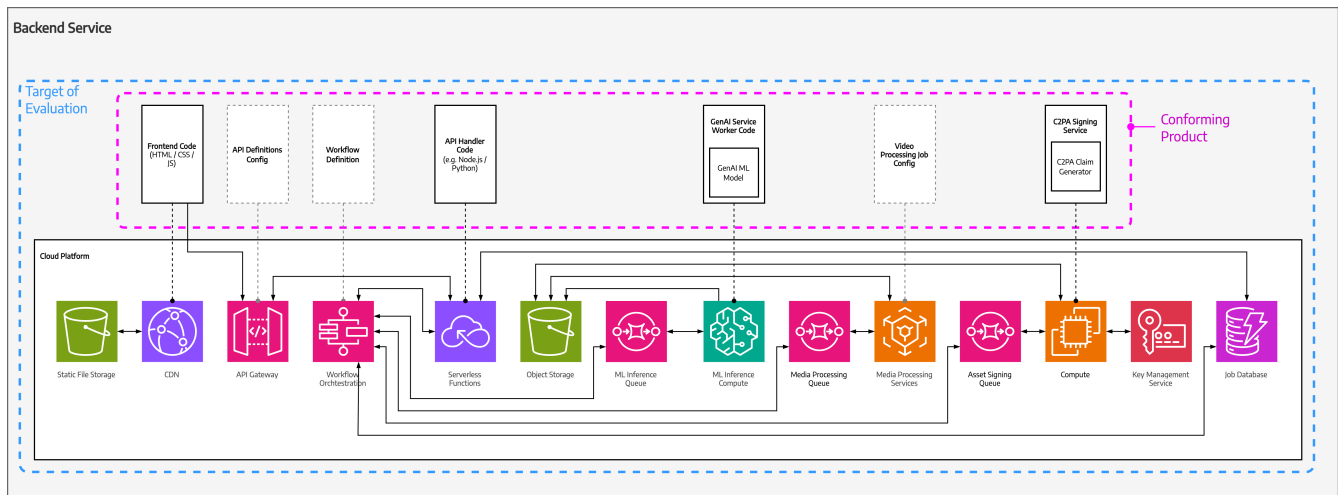
<sup>9</sup> Timely manner means 30/90/180 days to fix high, moderate, and low severity vulnerabilities as determined by the Common Vulnerability Scoring System (CVSS) (<https://www.first.org/cvss/>)

<sup>10</sup> Examples include: AWS - IAM (<https://aws.amazon.com/iam/>); Azure - Azure RBAC (<https://learn.microsoft.com/en-us/azure/role-based-access-control/overview>); GCP - Identity and Access Management (IAM) (<https://cloud.google.com/security/products/iam>)

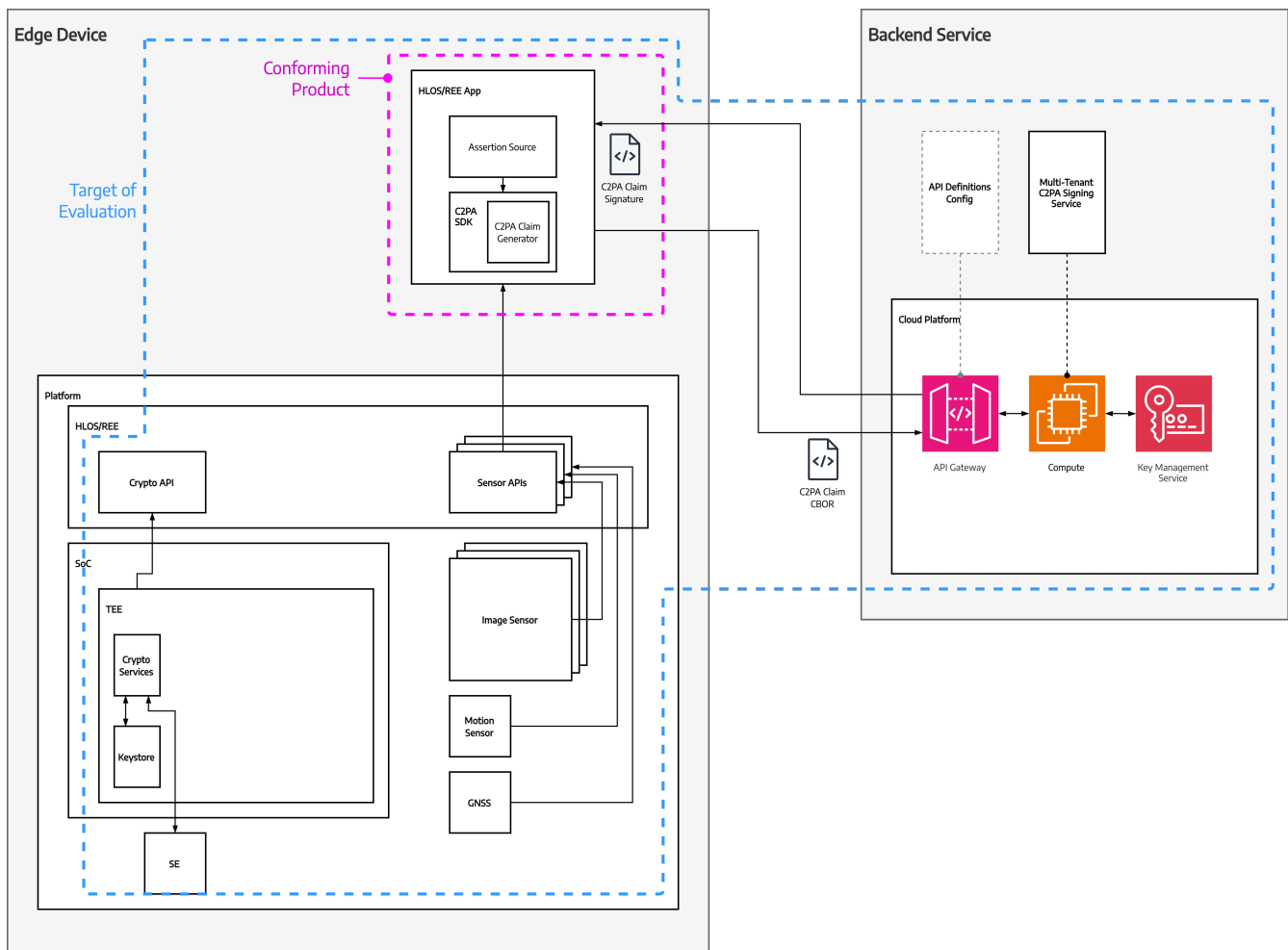
## Appendix A: Example Implementation Architectures

This appendix contains various examples of different implementation architectures for [Generator Products](#) and their Target of Evaluation. The Target of Evaluation is considered in its totality when assigning the Generator Product a [Max Assurance Level](#).

### A.1. Backend Class: Video Generation Service

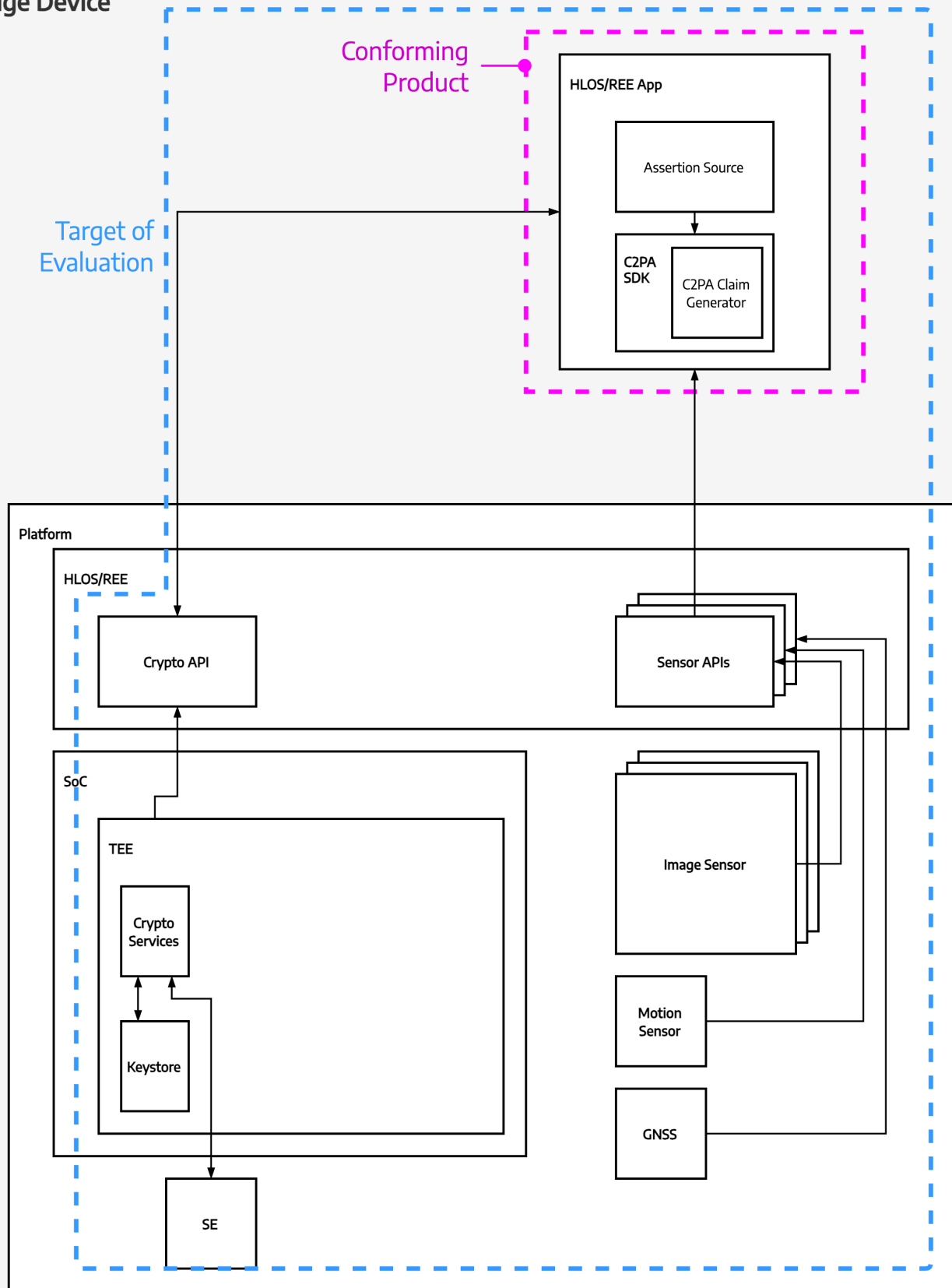


## A.2. Distributed Class: Edge REE App with Backend Claim Signing



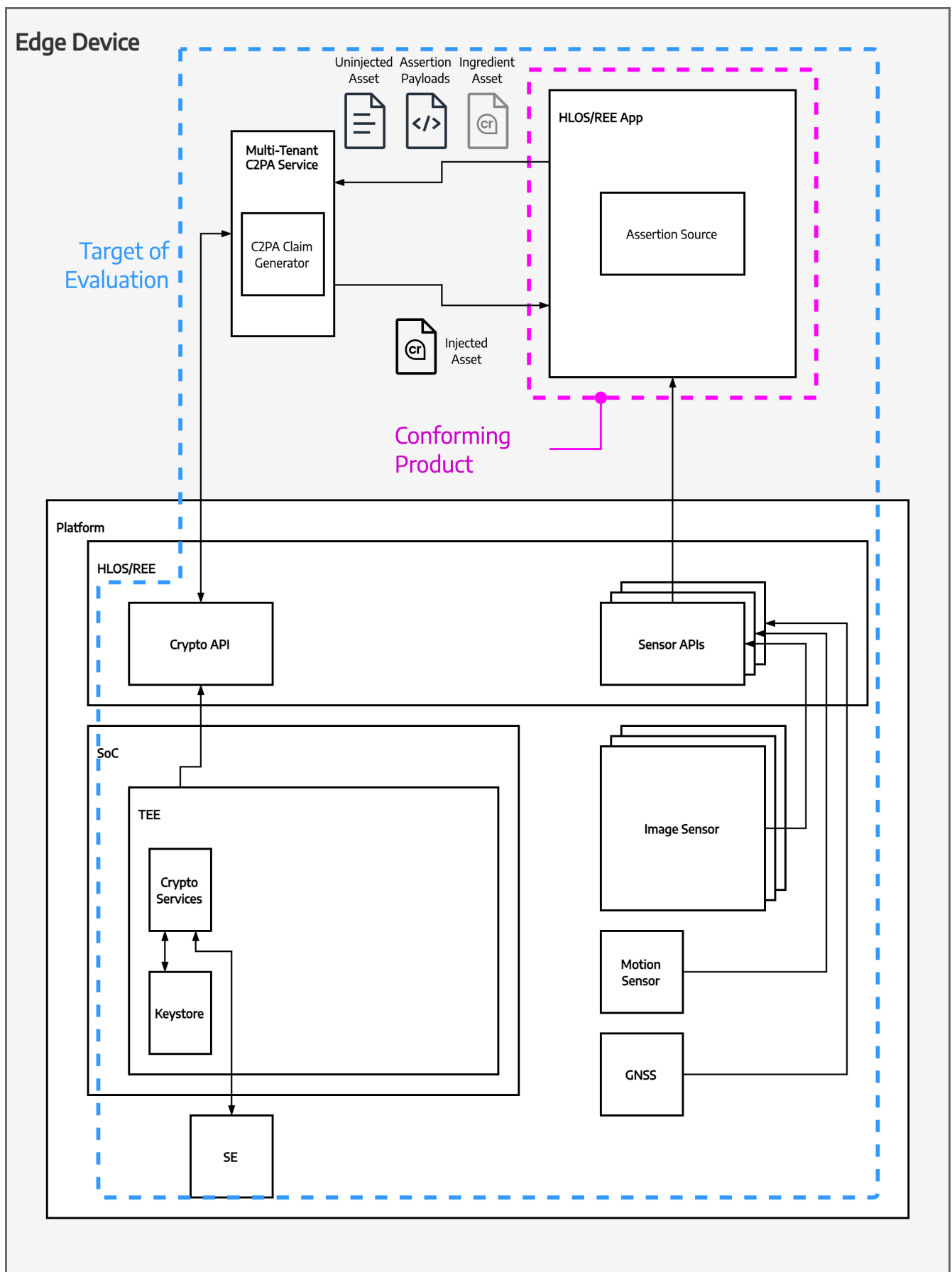
## A.3. Edge Class: Monolithic REE App

## Edge Device

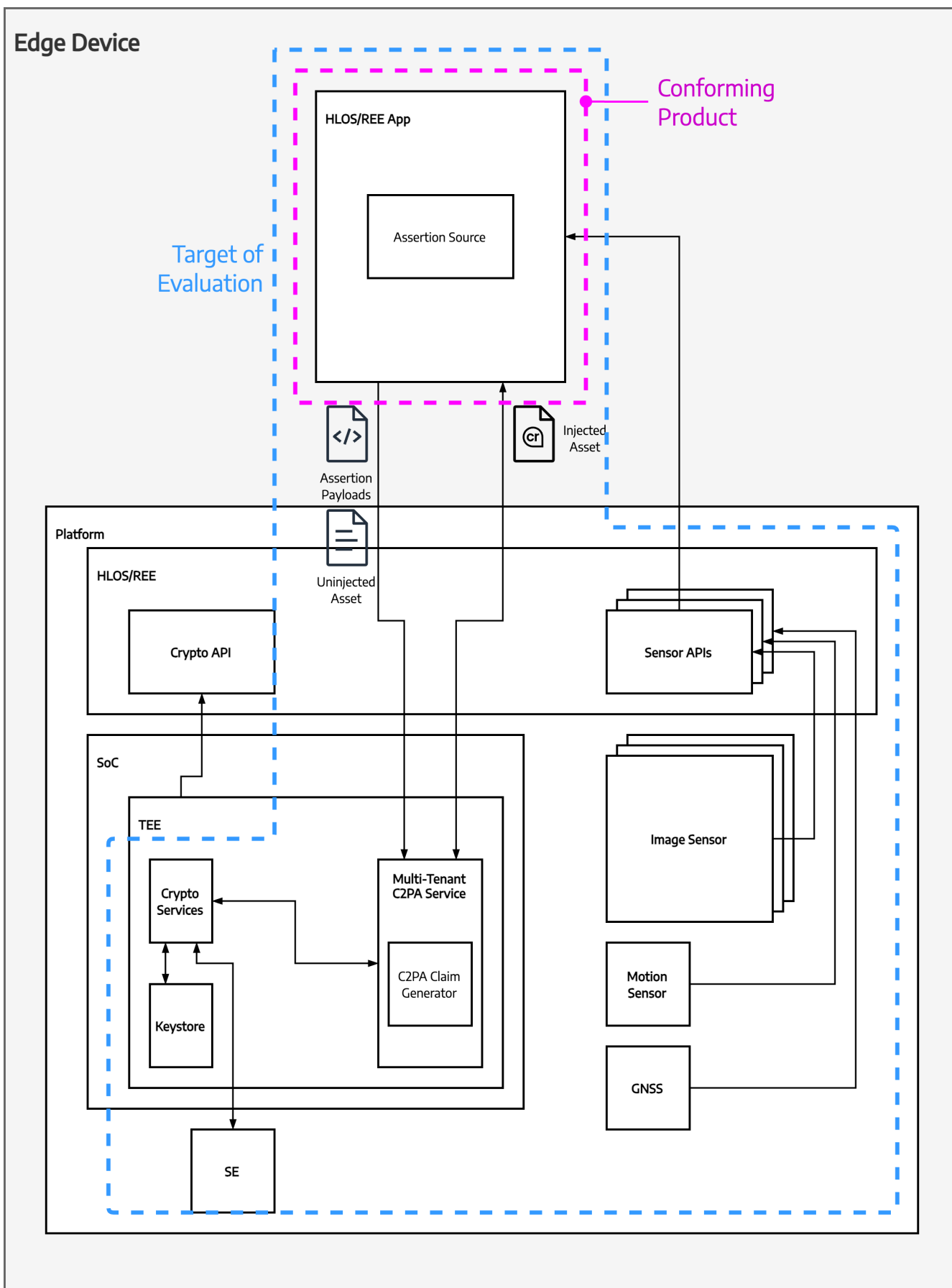




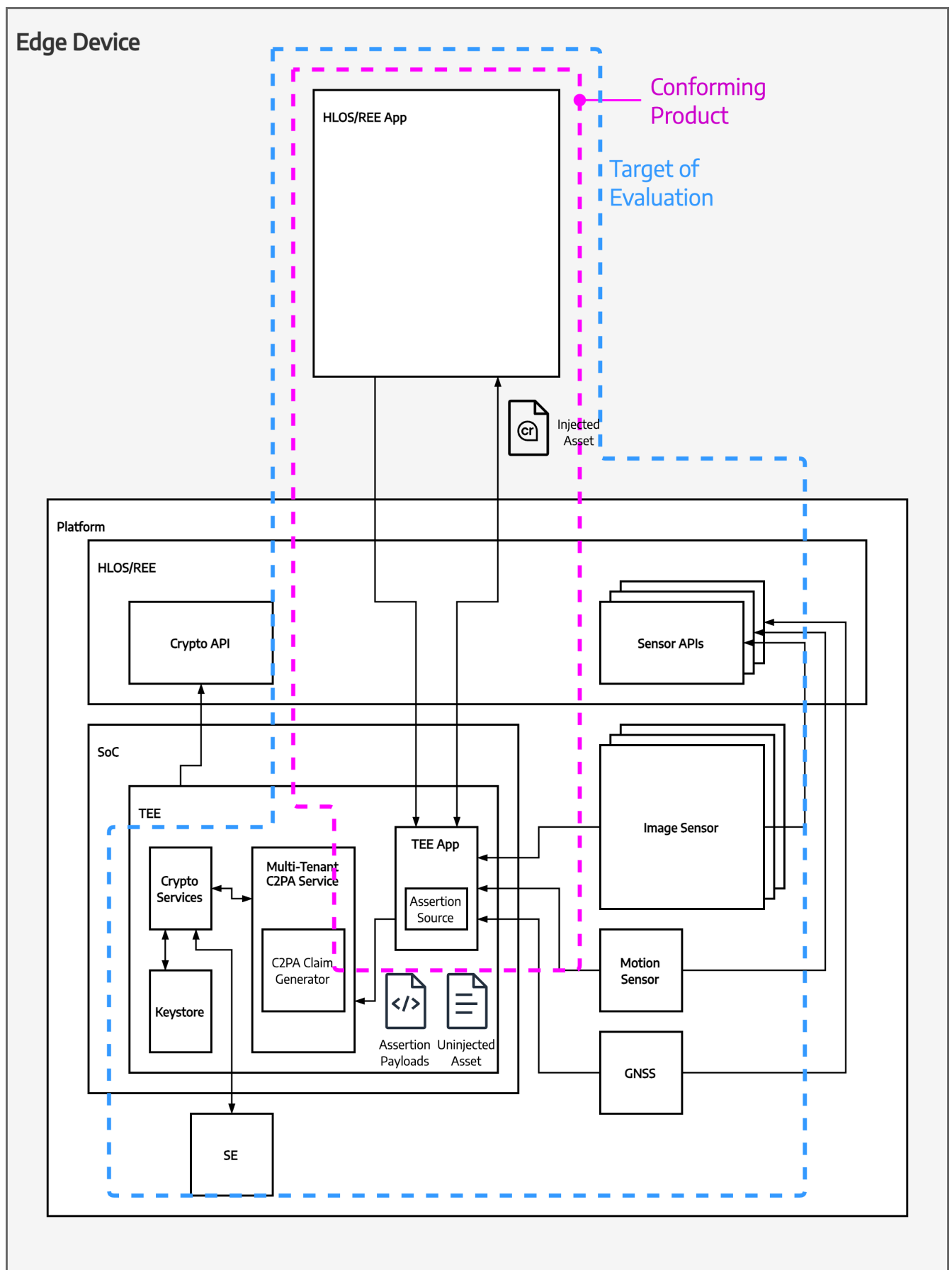
## A.4. Edge Class: REE App relying on 3p Discrete REE Claim Generator



### A.5. Edge Class: REE App relying on 3p Discrete TEE Claim Generator



## A.6. Edge Class: REE+TEE App relying on 3p Discrete TEE Claim Generator



# Appendix B: Non-normative guidance

This non-normative guidance is designed to help Applicants understand the types of security technologies that may help them achieve the desired Assurance Level for their implementation. The examples provided here do not represent formal endorsements by the C2PA, its Steering Committee, its Technical Working Group, or the TWG's Conformance Task Force of any specific commercial offering.

## B.1. AVA\_VAN Levels

See <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:20004:ed-2:v1:en> for more information on AVA\_VAN levels.

## B.2. Key management environments

Examples of key management environments include:

- For **Edge** Implementation Class:
  - [Apple Secure Enclave](#)
  - [Android Keystore](#)
  - [Microsoft Platform Crypto Provider](#)
  - Other keystore services that operate within a Trusted Execution Environment running at privilege level higher than Non-Secure World within ARM architecture.
- For **Backend** and **Distributed** Implementation Classes:
  - [AWS Key Management Service](#)
  - [Microsoft Azure Key Vault](#)
  - [Google Cloud Key Management Service](#)
  - [HashiCorp Vault](#)

## B.3. Integrity attestation methods

Examples of integrity attestation methods include:

- For **Edge** and **Distributed** Implementation Classes:
  - [Apple App Attest](#)
  - [Google Play Integrity API](#)
  - [Android Key Attestation](#)
- For **Backend** and **Distributed** Implementation Classes:
  - [AWS Nitro Enclave Attestation](#)
  - [Microsoft Azure Attestation](#)
  - [Google Cloud Attestation](#)

## B.4. Accepted certification schemes

### B.4.1. SOC2 Type 2

To achieve SOC 2 compliance, organizations must demonstrate that they have robust controls in place for managing private keys, including:

- \* Secure Storage: Protecting private keys from unauthorized access and ensuring they are stored securely.
- \* Access Control: Limiting access to private keys based on the principle of least privilege.
- \* Key Rotation: Regularly rotating private keys to mitigate the risk of compromise.
- \* Encryption: Using strong encryption algorithms to protect data at rest and in transit.

## B.5. SCA and SBOM dependency vulnerability scanning tools

Examples of tools that enable identifying vulnerabilities in dependencies include:

- Open-source:
  - [OSV-Scanner](#)
  - [Grype](#)
  - [Dependency-Track](#)
- Commercial:
  - [GitHub security and analysis for repositories](#)
  - [Tenable Web App Scanning](#)
  - [Acunetix](#)
  - [Invicti, formerly Netsparker](#)
  - [Qualys SCA](#)
  - [Checkmarx One](#)
  - [NowSecure](#)
  - [Google Cloud Artifact Analysis](#)

## B.6. Basic exploit countermeasures

Examples of basic exploit countermeasures include:

- ASLR
- Stack canaries
- Guard pages
- DEP
- Safe Heap
- NX

## B.7. Static analysis tools

Examples of static analysis tools include:

- Klocwork
- Coverity
- IntelliJ
- Fortify
- Snyk Code
- SonarQube

## Appendix C: Generator Product Security Architecture Document Template

Please note that sections marked "Required" for a particular section must be present and filled in order for the Generator Product to be evaluated at its target Assurance Level. Even if a particular section is not marked "Required" for the target Assurance Level, it is strongly recommended that the Applicant provide as much detail as possible that may be relevant in that section.

### C.1. Generator Product Information

#### C.1.1. Applicant organization details

Please provide the full legal name of the Applicant organization, its address, and contact information.

#### C.1.2. Distinguished name

Please document the values of the following fields for the Generator Product's Distinguished Name:

1. Common Name (CN): the user-facing marketing name and/or model of the device, application, or service
2. Organization (O): the legal name of the Applicant organization
3. Organizational Unit (OU): (optional) the department or subdivision within the Applicant's organization that created this product
4. Country (C): Two-letter country code in ISO 3166-1 alpha-2 format where the organization named in the 'O' field has its base of operations.

#### C.1.3. Generator Product Description

Please provide a high-level description of the Generator Product, including its intended use cases, target audience, and key features.

#### C.1.4. Generator Product Target of Evaluation (TOE) Description

Please provide a high-level description of the complete Target of Evaluation of this submission, which encompasses the Generator Product as well as the underlying platform(s) on which the Generator Product will perform its functions.

**Architectural diagrams are strongly encouraged.**

### **C.1.5. Implementation Class**

Please specify the Implementation Class of this Generator Product: **Edge**, **Backend**, or **Distributed**.

### **C.1.6. Target Max Assurance Level**

Please specify your intended maximum Assurance Level for which this Generator Product should be evaluated.

### **C.1.7. Target Generator Product capabilities**

Please provide the list of claim generation and claim validation functions that this Generator Product will support for various media types.

Possibilities include one or more of:

1. Claim generation:
  - a. Still image media types:
    - i. image/jpeg
    - ii. image/jxl
    - iii. image/png
    - iv. image/svg+xml
    - v. image/gif
    - vi. image/x-adobe-dng
    - vii. image/tiff
    - viii. image/webp
    - ix. image/heic
    - x. image/heic-sequence
    - xi. image/heif
    - xii. image/heif-sequence
    - xiii. image/avif"
  - b. Video media types:
    - i. video/x-msvideo
    - ii. video/mp4
    - iii. video/quicktime
  - c. Audio media types:
    - i. audio/flac
    - ii. audio/MPA
    - iii. audio/mpeg
    - iv. audio/wav
    - v. audio/aac

vi. audio/mp4

d. Document media types:

i. application/pdf

ii. application/epub+zip

iii. application/vnd.openxmlformats-officedocument.wordprocessingml.document

iv. application/vnd.oasis.opendocument.text

v. application/oxps

e. Fonts:

i. font/otf

f. ML Models:

i. jax

ii. keras

iii. ml\_net

iv. mxnet

v. onnx

vi. openvivo.parameter

vii. openvivo.topology

viii. pytorch

ix. tensorflow

x. numpy

xi. protobuf

xii. pickle

xiii. savedmodel

2. Claim validation:

a. Still image media types:

i. image/jpeg

ii. image/jxl

iii. image/png

iv. image/svg+xml

v. image/gif

vi. image/x-adobe-dng

vii. image/tiff

viii. image/webp

ix. image/heic

x. image/heic-sequence

xi. image/heif

xii. image/heif-sequence



- xiii. image/avif"
- b. Video media types:
  - i. video/x-msvideo
  - ii. video/mp4
  - iii. video/quicktime
- c. Audio media types:
  - i. audio/flac
  - ii. audio/MPA
  - iii. audio/mpeg
  - iv. audio/wav
  - v. audio/aac
  - vi. audio/mp4
- d. Document media types:
  - i. application/pdf
  - ii. application/epub+zip
  - iii. application/vnd.openxmlformats-officedocument.wordprocessingml.document
  - iv. application/vnd.oasis.opendocument.text
  - v. application/oxps
- e. Fonts:
  - i. font/otf
- f. ML Models:
  - i. jax
  - ii. keras
  - iii. ml\_net
  - iv. mxnet
  - v. onnx
  - vi. openvivo.parameter
  - vii. openvivo.topology
  - viii. pytorch
  - ix. tensorflow
  - x. numpy
  - xi. protobuf
  - xii. pickle
  - xiii. savedmodel

## C.2. Security architecture details

## C.2.1. Authentication for certificate enrollment

### Certificate enrollment process

Describe the certificate enrollment process for instances of the Generator Product, including the steps involved, the entities involved, and the security measures in place.

### Management of certificate enrollment authentication secrets

*Required for Assurance Level 1 and Level 2*

Describe the method for managing authentication secrets used during certificate enrollment, including how they are generated, stored, and protected from unauthorized access.

### Confirming GP binary identity

*Required for Assurance Level 2*

Describe the GP and/or GP TOE produce verifiable artifacts backed by a hardware Root of Trust, which the GP must provide to a CA during automated certificate enrollment, that confirm GP binary/binaries via package names, hashes, code signing certificates, other certificates, or a combination of the above.

## C.2.2. Key generation, storage, and usage

### Key generation and storage method

*Required for Assurance Level 1 and Level 2*

Please describe:

1. How keys are generated and stored, including the algorithms used, key sizes, and storage mechanisms.
2. Details of the claim signing key access controls (including encryption) in place that are designed to prevent unauthorized access
3. Details of the key rotation process
4. For **Distributed** and **Backend** Implementation Classes, documentation of the method for mutual authentication and role validation between the subsystems.

### Attestation of key generation and storage

*Required for Assurance Level 2*

Please describe:

1. Details of the key management environment that the GP TOE uses, and its security properties. This may include commonly-accepted security certifications for the key management environment, and if applicable, certification by an accredited, independent 3rd party auditor certifying that the claim signing key is stored in such an environment.

### Authentication before using keys

*Required for Assurance Level 2 for **Distributed** or **Backend** Implementation Classes*

Please describe:

1. The method for producing or deriving verifiable artifacts backed by a hardware Root of Trust by the subsystems to confirm the identity of the calling client.

### **C.2.3. Protections against Claim Generator misconfiguration and abuse**

#### **Processes for detecting vulnerabilities in dependencies (SCA/SBOM)**

*Required for Assurance Level 1 and 2*

Please describe:

1. The SCA/SBOM dependency vulnerability scanning tools used during the Claim Generator build or integration process.
2. The process by which the build and deployment pipeline prevents the release, more than a 90 days after detection, of the Claim Generator with known **CRITICAL** or **HIGH** severity vulnerabilities.

#### **Basic exploit countermeasures, static analysis, software image authentication, and security patching**

*Required for Assurance Level 2*

Please describe:

1. Claim generator build scripts and build flags confirming enablement of countermeasures
2. Countermeasures functional test report.
3. Static analysis tools used
4. If applicable, access control methods
5. If applicable, binary image authentication methods
6. If applicable, external input validation methods
7. If applicable, access control lists for external input ingress points
8. Details of the verifiable artifacts backed by a hardware Root of Trust confirming the recency of security patches for the Claim Generator that can be provided to the CA for evaluation during certificate enrollment.

### **C.2.4. Protections against misconfiguration and abuse of software that processes or modifies Digital Content or assertions**

*Required for Assurance Level 1 and 2*

Please describe:

1. The SCA/SBOM dependency vulnerability scanning tools used during the build or integration process of software that processes and/or modifies Digital Content or assertions
2. The process by which the build and deployment pipeline prevents the release, more than a 90 days after detection, of such software with known **CRITICAL** or **HIGH** severity vulnerabilities.

## Basic exploit countermeasures, static analysis, software image authentication, and security patching

*Required for Assurance Level 2*

Please describe:

1. If applicable, description of image authentication methods of source and its execution environment for all sources of assets and assertions implemented in software.
2. Build scripts and build flags confirming enablement of countermeasures
3. Countermeasures functional test report
4. Static analysis tools used
5. If applicable, binary image authentication methods
6. Confirmation of support of isolation of the source processes (and/or threads, in RTOS based systems) and protection of the inter-process communication channels by the kernel or operating system in use. This can be achieved by demonstrating that:
  - a. the processes (and/or threads, in RTOS based systems) run under a unique operating system UID or user account from other processes on the device,
  - b. forms of inter-process communication (Android broadcasters and receivers, IPC channels, etc.) are limited to only those necessary for the application to function and that ACLs limit the processes (and/or threads, in RTOS based systems) that connect to them.
7. Details of the verifiable artifacts backed by a hardware Root of Trust confirming binary integrity and recency of security patches for software that processes or modifies Digital Content and/or Assertions, which can be provided to the CA for evaluation during certificate enrollment.

### C.2.5. Protections against interception and/or modification of traffic

#### Encryption of network traffic

*Required for Assurance Level 1 for **Distributed** and **Backend** Implementation Classes only*

Please describe:

1. the TLS versions or equivalent protocols in use and the supported cryptographic protocols for network communication between the subsystems (**Edge** and/or **Backend**).

#### Protection of inter-process communication

Please describe:

1. Support of isolation of the source processes (and/or threads, in RTOS based systems) and protection of the inter-process communication channels by the kernel or operating system in use. This can be achieved by demonstrating that:
  - a. the processes (and/or threads, in RTOS based systems) run under a unique operating system UID or user account from other processes on the device,
  - b. forms of inter-process communication (Android broadcasters and receivers, IPC channels, etc.) are limited to only those necessary for the application to function and that ACLs limit the processes (and/or threads, in RTOS based systems) that connect to them.

## C.2.6. Protections against exploitation of hosting environment

*Required for Assurance Level 1 and 2 for **Distributed** and **Backend** Implementation Classes only*

Please describe:

1. IAM system employed and coverage of security boundaries (e.g., virtual machines, cloud storage repositories) related to asset and claim generation.
2. Access policies for human and non-human principal (e.g., service accounts or production identities)
3. IAM policies showing access for main cloud resources (e.g., VM instances, cloud storage buckets).
4. The process used by the Applicant to ensure vulnerability scanning and/or security review is performed for software dependencies and API surfaces.
5. The process used by the Applicant to ensure vulnerabilities are fixed or mitigated in a timely manner.

*Required for Assurance Level 2 for **Distributed** and **\*Backend** Implementation Classes only*

Please describe:

1. Audit logging system, event logging policies, and monitoring rules
2. HIDS software or similar systems used
3. Network architecture and configuration, including a description of network isolation boundaries

Additionally, please attach the following:

1. Report showing audit logging is enabled
2. Report showing active operation of a HIDS or similar systems
3. Report showing active network isolation covering the application's network infrastructure