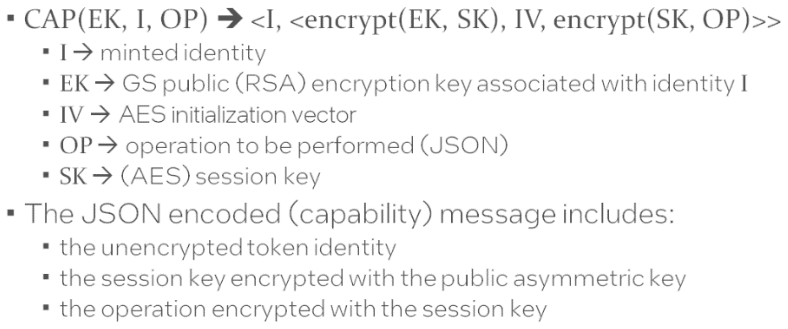
Token Guardian Protocol Description

Terminology:

- Offered Asset (DA) : A digital asset that OA wants to monetize
- Offering Agent (**OA**): The agent initially offering DA. Generally, the OA runs the GS
- Current Owner (**CO**)
- New Owner (**NO**)
- Guardian Service (**GS**): Service that manages access to DA. Verifies capabilities. Purely operational, no policies. Maps capabilities to operations on the asset
- Token Issuer Object (**TIO**): Created by the OA. Creates TOs for DA (issuer)
- Token Object (**TO**): Created by a TIO. Generates capabilities that can be evaluated by GS. A formal definition of capability is provided in the figure below:



- CAP(EK, I, OP) ➔ <I, <encrypt(EK, SK), IV, encrypt(SK, OP)>>
  - I → minted identity
  - EK → GS public (RSA) encryption key associated with identity I
  - IV → AES initialization vector
  - OP → operation to be performed (JSON)
  - SK → (AES) session key
- The JSON encoded (capability) message includes:
  - the unencrypted token identity
  - the session key encrypted with the public asymmetric key
  - the operation encrypted with the session key

Figure 1: A definition of capability as generated by the NFT smart contract after policy verification. Capabilities are processed by Capability engine located within the Guardian Service.

The protocol is divided into four phases:

1. Initialization Phase

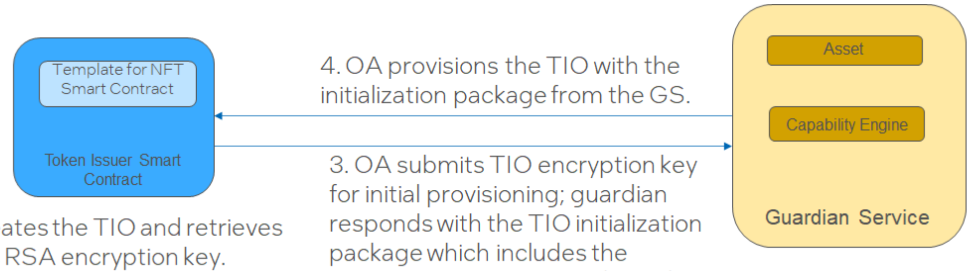
2. Creating an NFT

3. Transferring an NFT

4. Invoking an operation.

Protocol for each of these phases are shown below using the terminology described above.

1. Initialization Phase:



1. GS creates management encryption key pair (RSA)

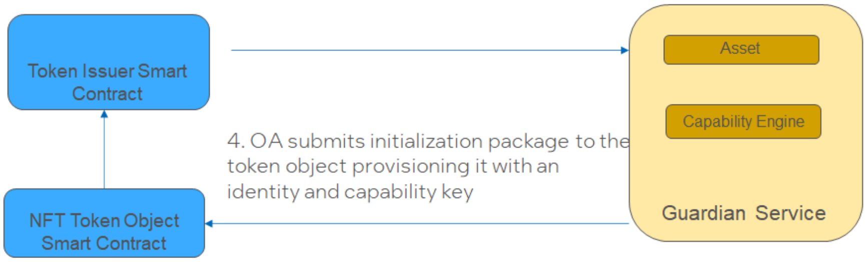
2. OA creates the TIO and retrieves the TIO's RSA encryption key.

3. OA submits TIO encryption key for initial provisioning; guardian responds with the TIO initialization package which includes the management encryption key. This package is encrypted with the TIO's encryption key.

4. OA provisions the TIO with the initialization package from the GS.

2. Creating an NFT:

2. TIO verifies the TO attestation package and generates a TO creation package that includes a capability (encrypted with the GS's management encryption key)

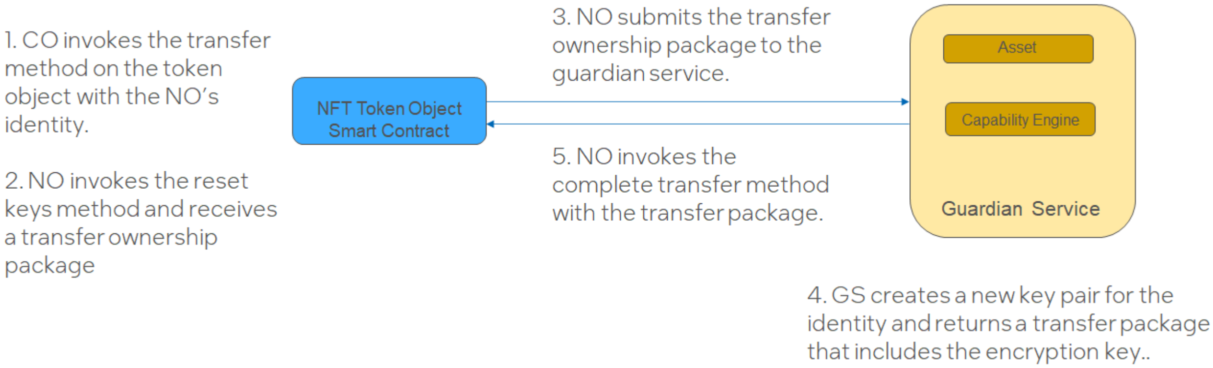


1. OA creates a TO and generates an attestation package; the attestation package is submitted to the TIO. OA is assigned as the current owner of the TO.

4. OA submits initialization package to the token object provisioning it with an identity and capability key

3. OA submits the TO creation package to the GS; GS creates a new identity with corresponding encryption key. GS returns a TO initialization package.

3. Transferring an NFT



1. CO invokes the transfer method on the token object with the NO's identity.

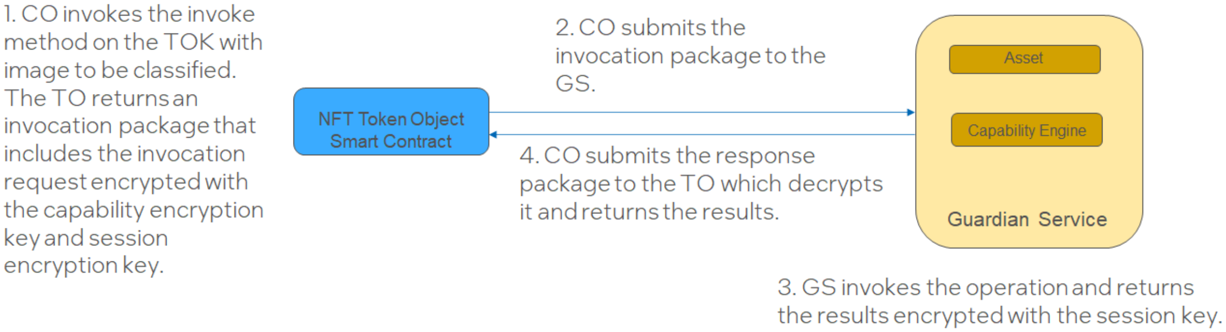
2. NO invokes the reset keys method and receives a transfer ownership package

3. NO submits the transfer ownership package to the guardian service.

5. NO invokes the complete transfer method with the transfer package.

4. GS creates a new key pair for the identity and returns a transfer package that includes the encryption key..

4. Invoking an operation



1. CO invokes the invoke method on the TOK with image to be classified. The TO returns an invocation package that includes the invocation request encrypted with the capability encryption key and session encryption key.

2. CO submits the invocation package to the GS.

4. CO submits the response package to the TO which decrypts it and returns the results.

3. GS invokes the operation and returns the results encrypted with the session key.