Assertions and Tokens
+
Path tracing

SPIFFE/SPIRE

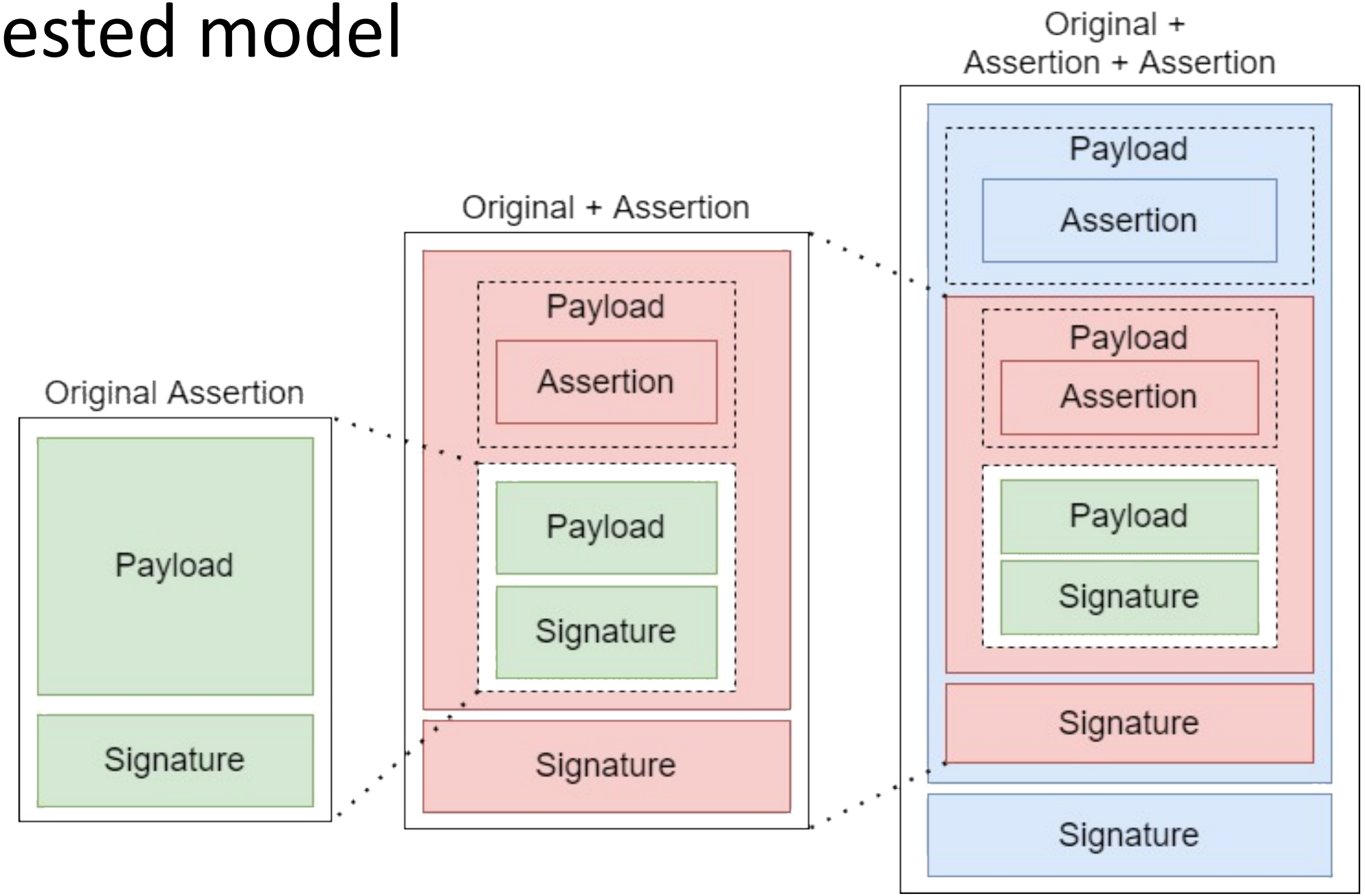
Nov/2022

Introduction

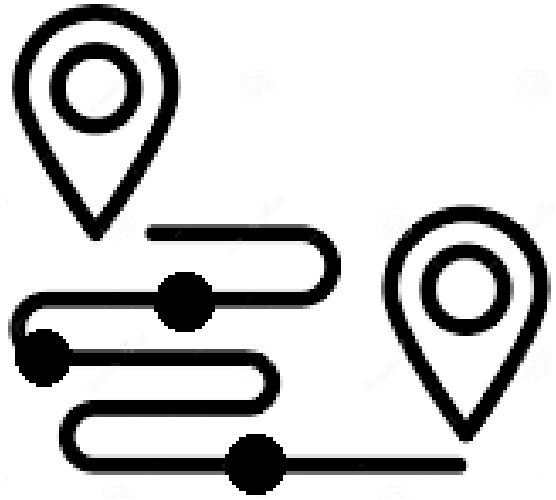
Main needs:

- A system that allow a subject to make arbitrary authenticated statements
- A token scheme that supports distributed signing, aggregate/concatenate signatures, and/or attenuations

Nested model



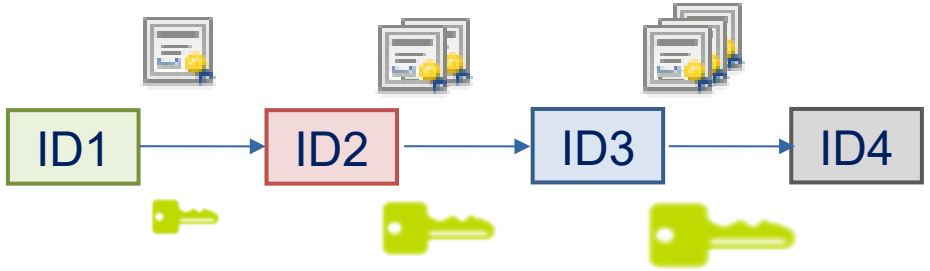
Token path tracing



Provide the path of workloads that a request has passed

- **ID mode**:
  - Uses SVID private key to sign, sending necessary certificates to identify the workload and validate the signature and iss/aud link
- **Anonymous mode**:
  - No ID associated to keys
  - Uses concatenated Schnorr signatures that results in smaller tokens and faster validation

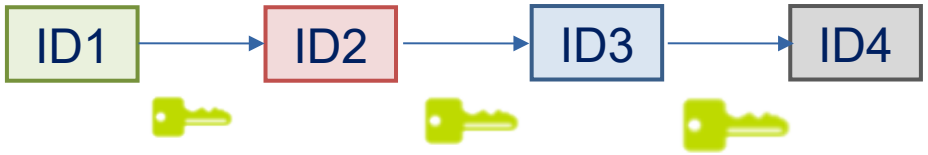
ECDSA – SVID
(ID mode)



Sign with SVID private key. Send SVID certificates with token

- Pros:
  - Certificates allow off-line validation and identification
  - Anonymous mode also available
- Cons:
  - ID mode requires more bandwidth
- Possibilities:
  - Use lightweight SVID

EdDSA – Schnorr Concatenated

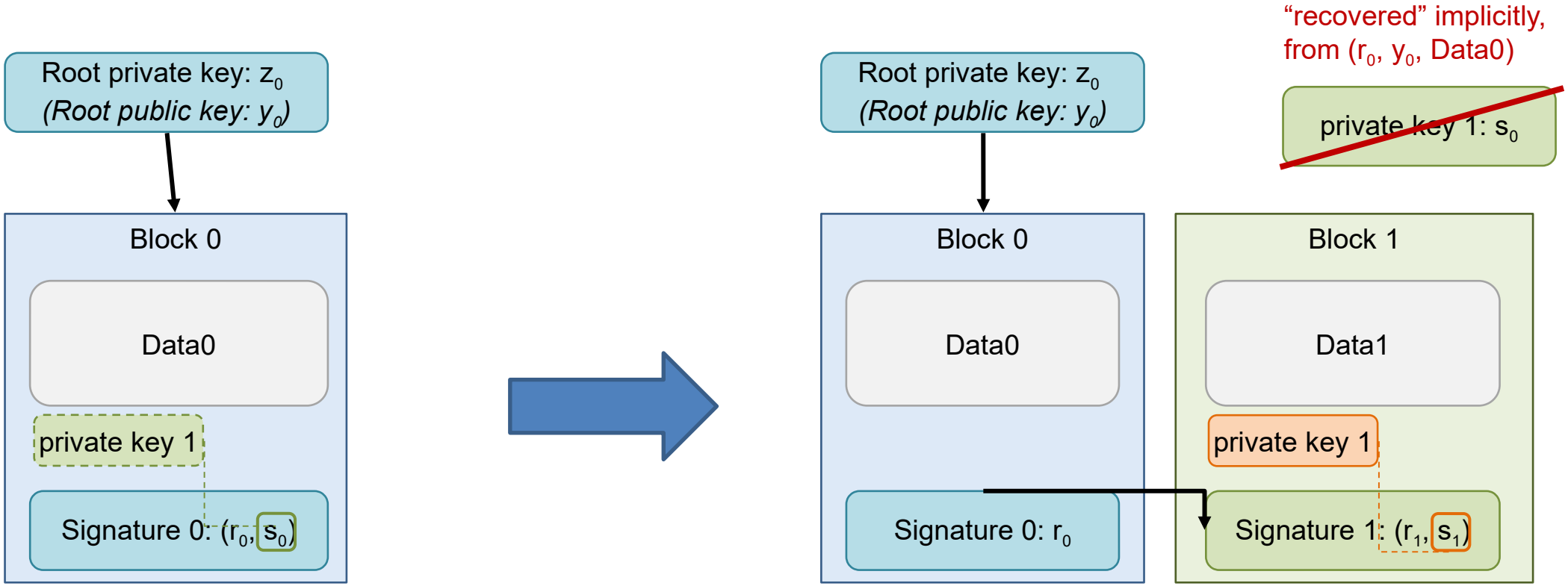


Biscuits-based solution. Each hop uses part of previous signature as private key

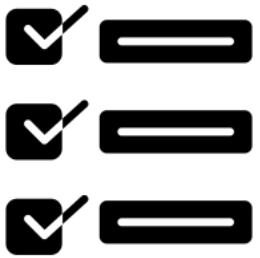
- Pros:
  - Smaller token size (compared to standard model and ECDSA)
  - Faster validation (using Galindo-Garcia) than sequencial model
  - Cryptographic-linked signatures
- Cons:
  - Only anonymous mode available
- Possibilities:
  - Study aggregated signatures state-of-art and ECDSA-Schnorr

SchCo-Biscuits

(using concatenated Schnorr-based signatures: Galindo-Garcia-style)



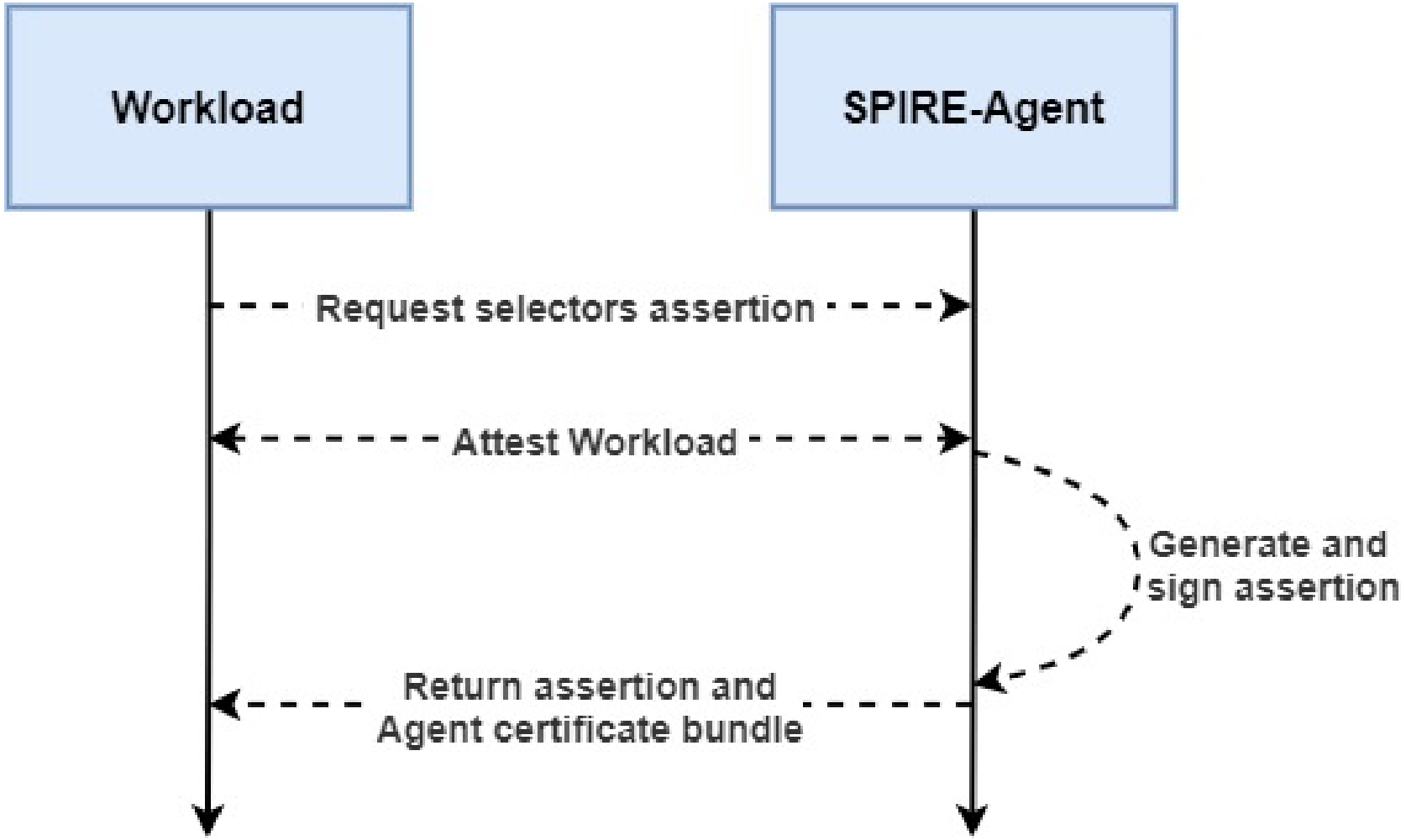
Selector-based Assertion



Contain selectors used by SPIRE-Agent during workload attestation process

- Generated and signed by SPIRE-Agent using its SVID
- Return the assertion and SPIRE-Agent certificate bundle

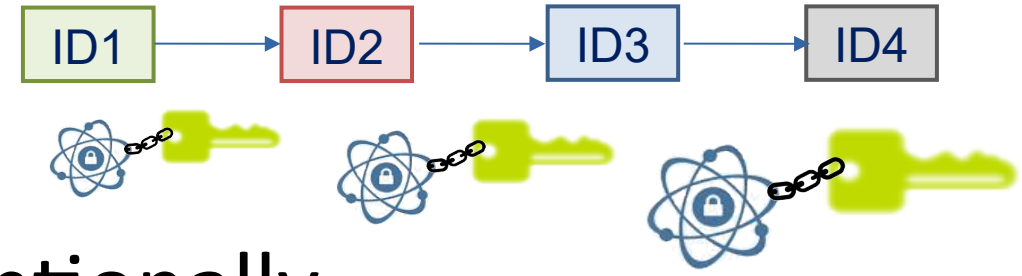
Selectors-based Assertion



Selector-based Assertion


```
{
  "iat": 1670470473,
  "sel": [
    { "type": "unix", "value": "uid:1000" },
    { "type": "unix", "value": "user:spire" },
    { "type": "unix", "value": "gid:1000" },
    { "type": "unix", "value": "group:spire" },
    { "type": "unix", "value": "supplementary_group:spire" },
    { "type": "unix", "value": "path:/opt/spire/bin/spire-agent" },
    {
      "type": "unix",
      "value": "sha256:ff9270b6c985fa1ad3476e5f1cf83648033cd1ec02b"
    }
  ]
}
```

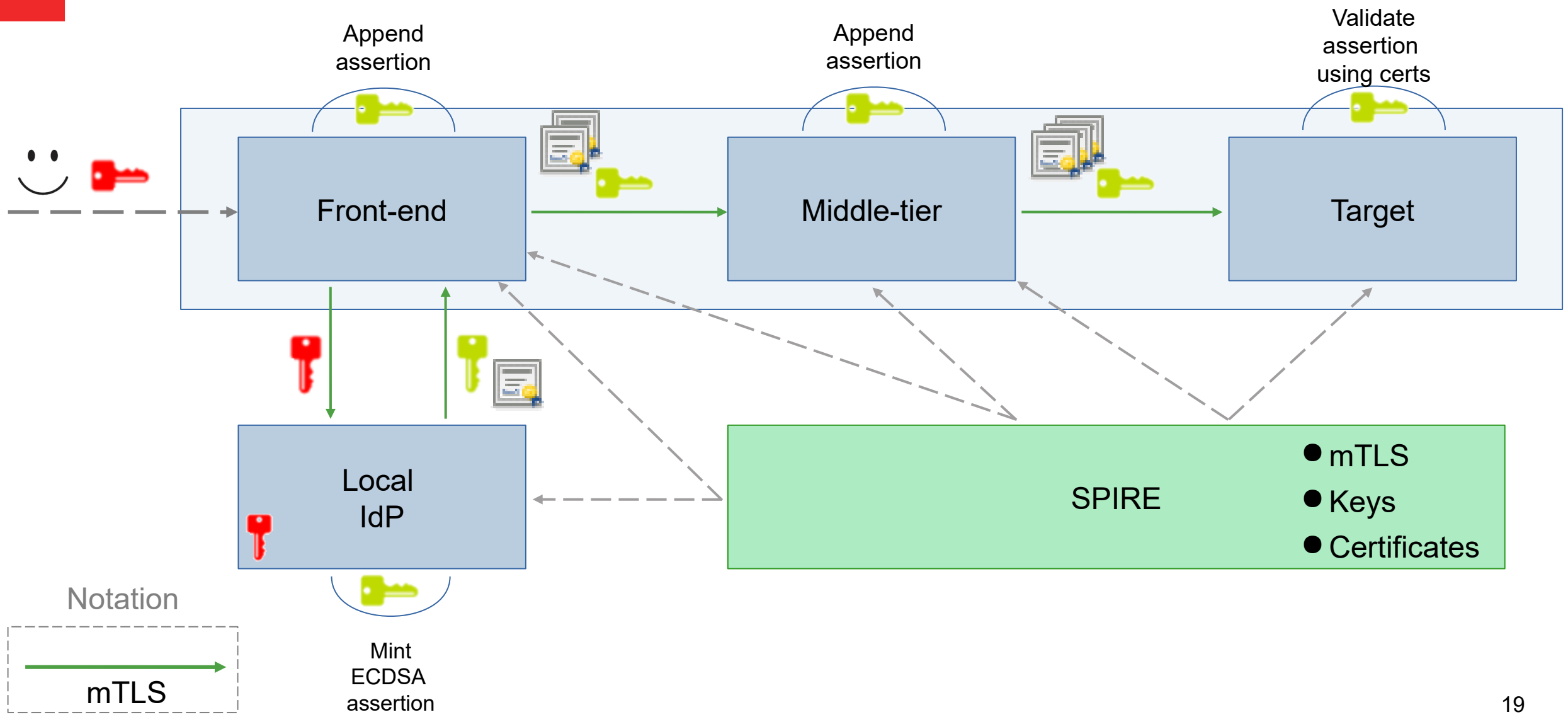
ECDSA – Dillithium



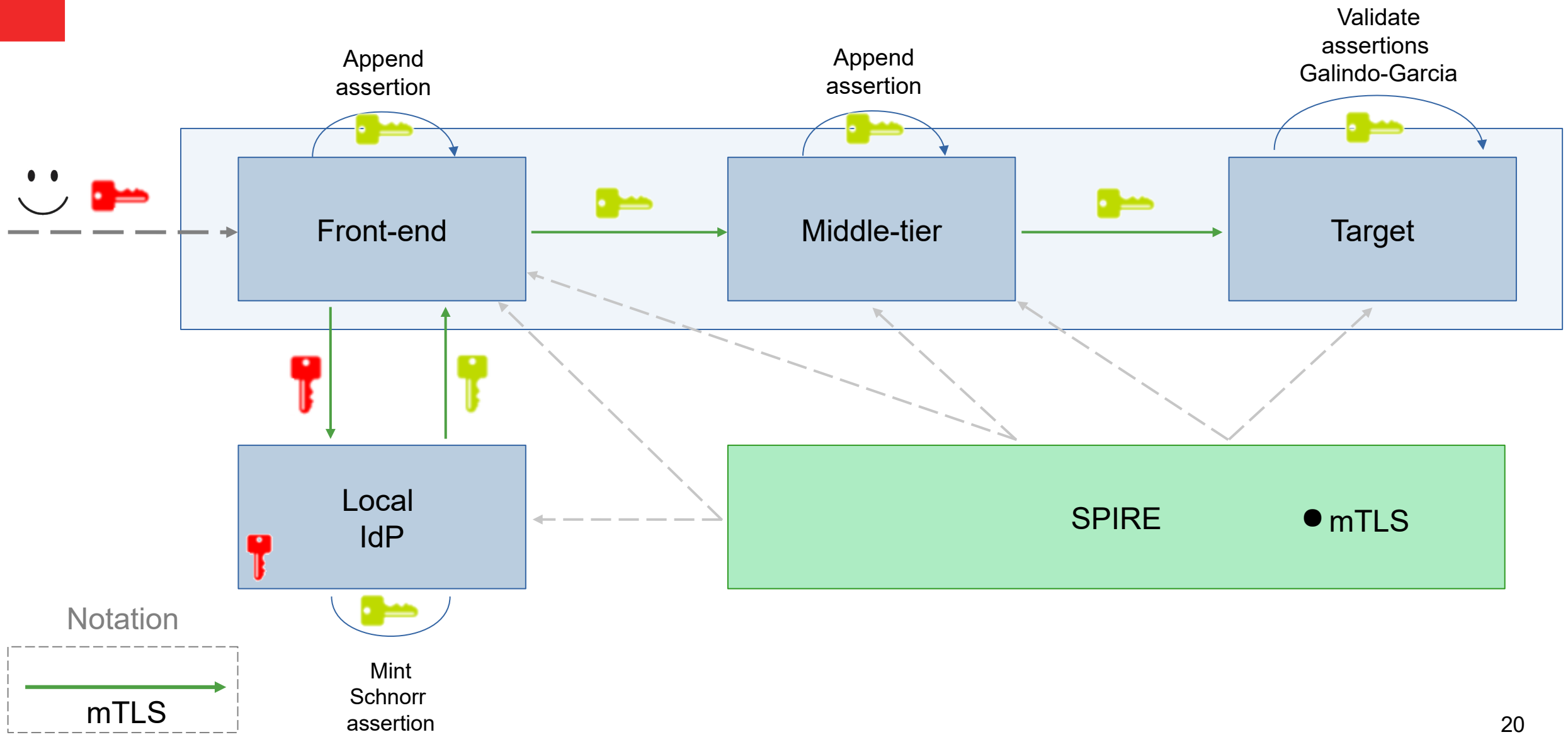
Sign with SVID private key adding, optionally, a post-quantum signature algorithm.

- Pros:
  - Improved security using post-quantum algorithm (ECDSA+Crystals)
- Cons:
  - Bigger keys/signatures
- Possibilities:
  - Optional to specific use cases
  - Follow-up state-of-art

Demo 1: ECDSA – SVID (ID mode)

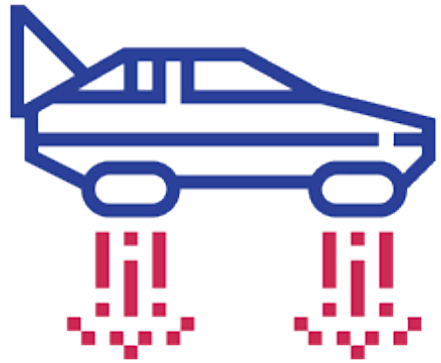


Demo 2: EdDSA – Schnorr (Anonymous mode)



Future Work

- Specify and implement lightweight SVID
- Identity-based SVID: lightweight SVID with Galindo-Garcia
- Use SchCo biscuits model in selectors assertion
- Post-Quantum algorithms (e.g. Crystals) analysis
- Protobuf / JSON analysis




WHO? WHERE?
WHEN? WHY? HOW?
HOW? WHAT? WHO?
WHERE?
WHO? WHERE?
WHAT?
WHY? WHAT?
WHEN?
WHERE? WHO?
HOW? WHAT?
WHO? WHERE?
WHY?
WHO? WHEN?
WHERE? WHAT?
WHO? WHY? HOW?
HOW? WHERE?
WHAT? WHY?
WHEN?
WHO? WHAT?
HOW?
WHAT? WHEN?
HOW? WHEN?
WHAT?
WHY? WHERE?
WHEN? HOW?
WHERE?