Assertions and Tokens
+
Path tracing

SPIFFE/SPIRE

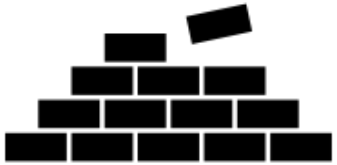
Dec/2022

Introduction

Main needs:

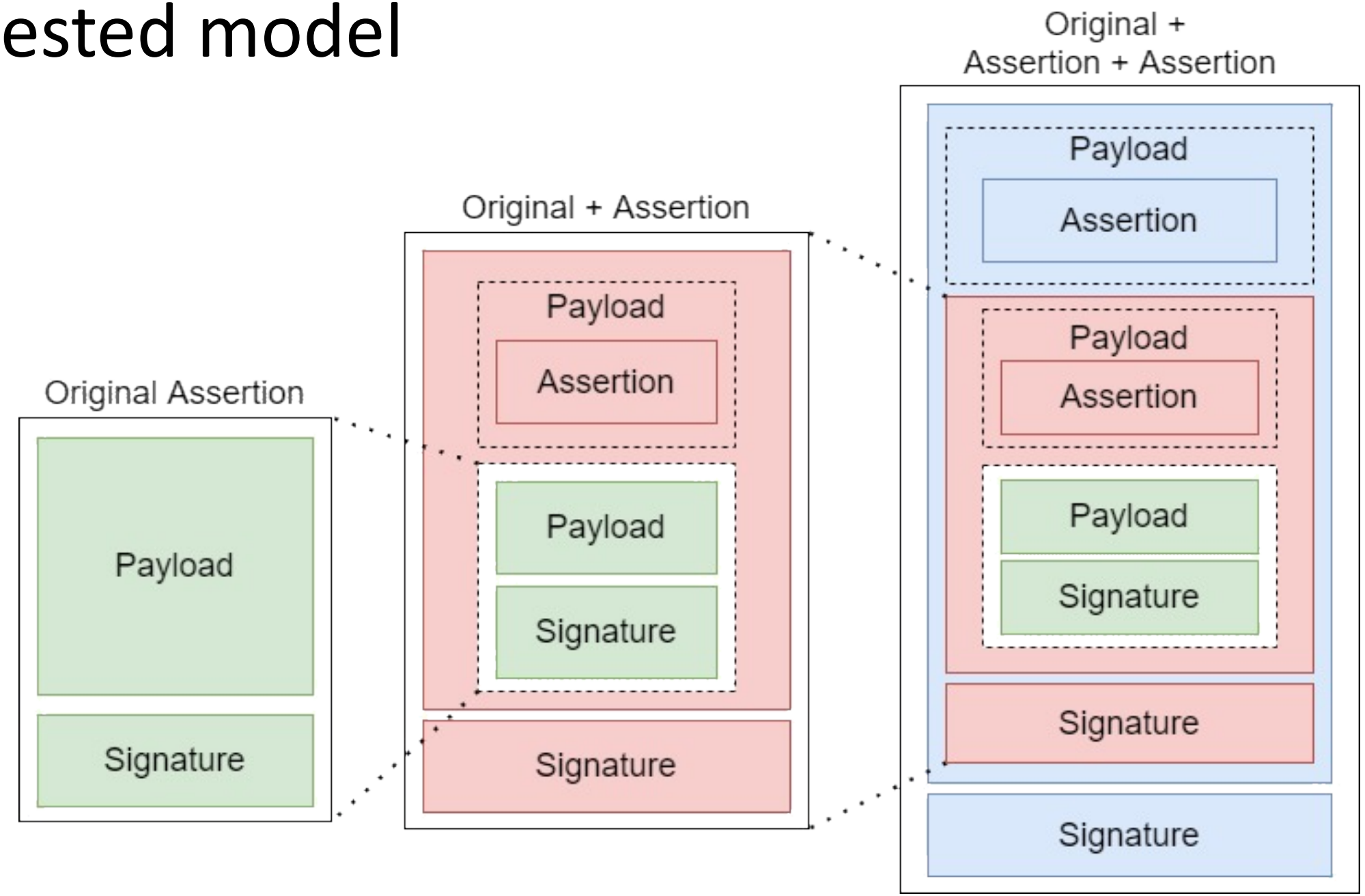
- A system that allow a subject to make arbitrary authenticated statements
- A token scheme that supports distributed signing, aggregate/concatenate signatures, and/or attenuations

Developed work

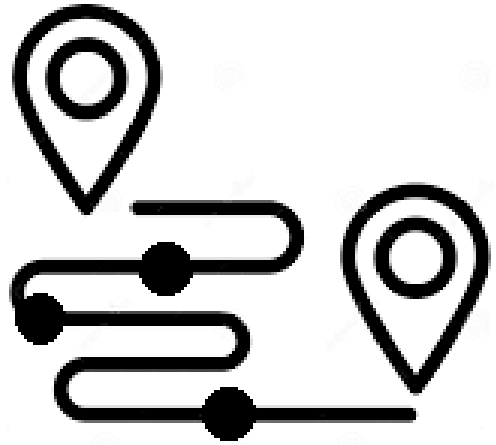


- Nested model
- Token path tracing
  - ID Mode
  - Anonymous mode
- Post-quantum signatures
- Selector-based Assertion

Nested model



Token path tracing



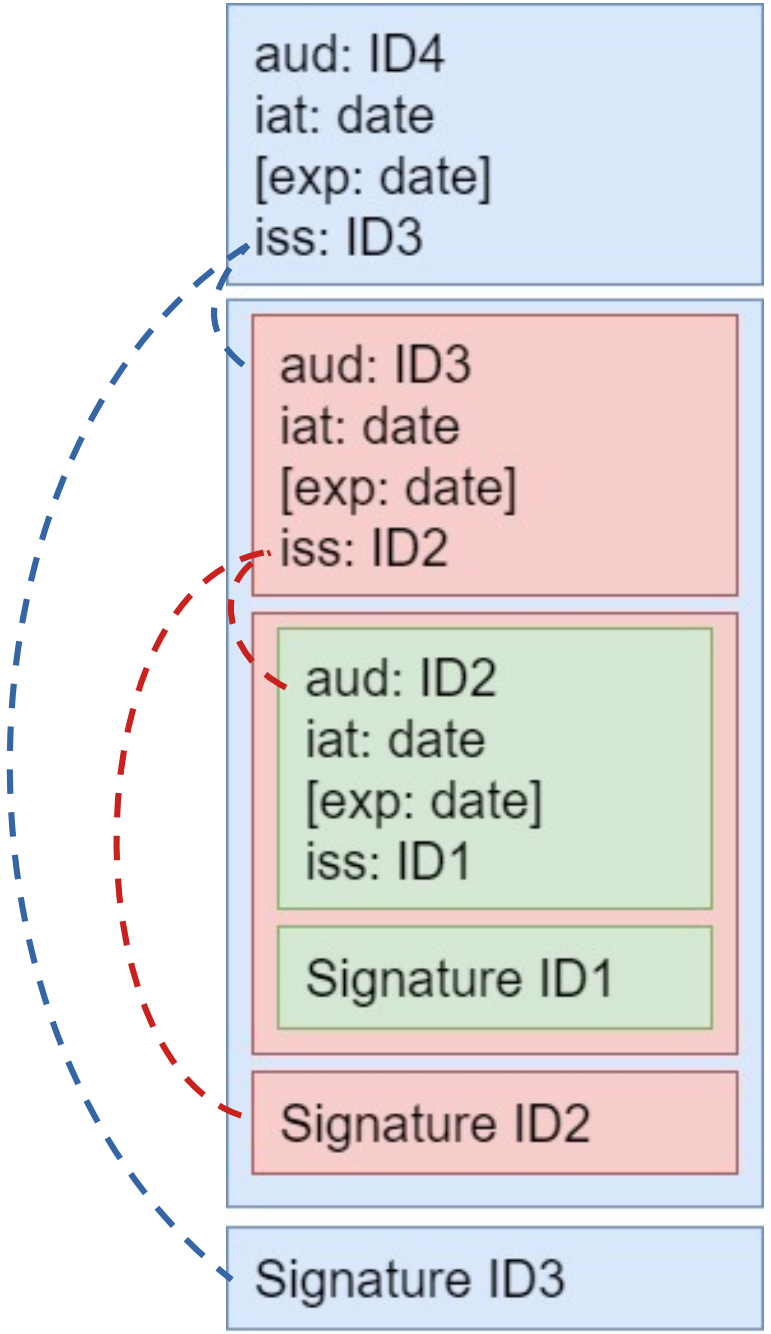
Provide the path of workloads that a request has passed

- **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 SchCo Biscuits model, that results in smaller tokens and faster validation

ID Mode

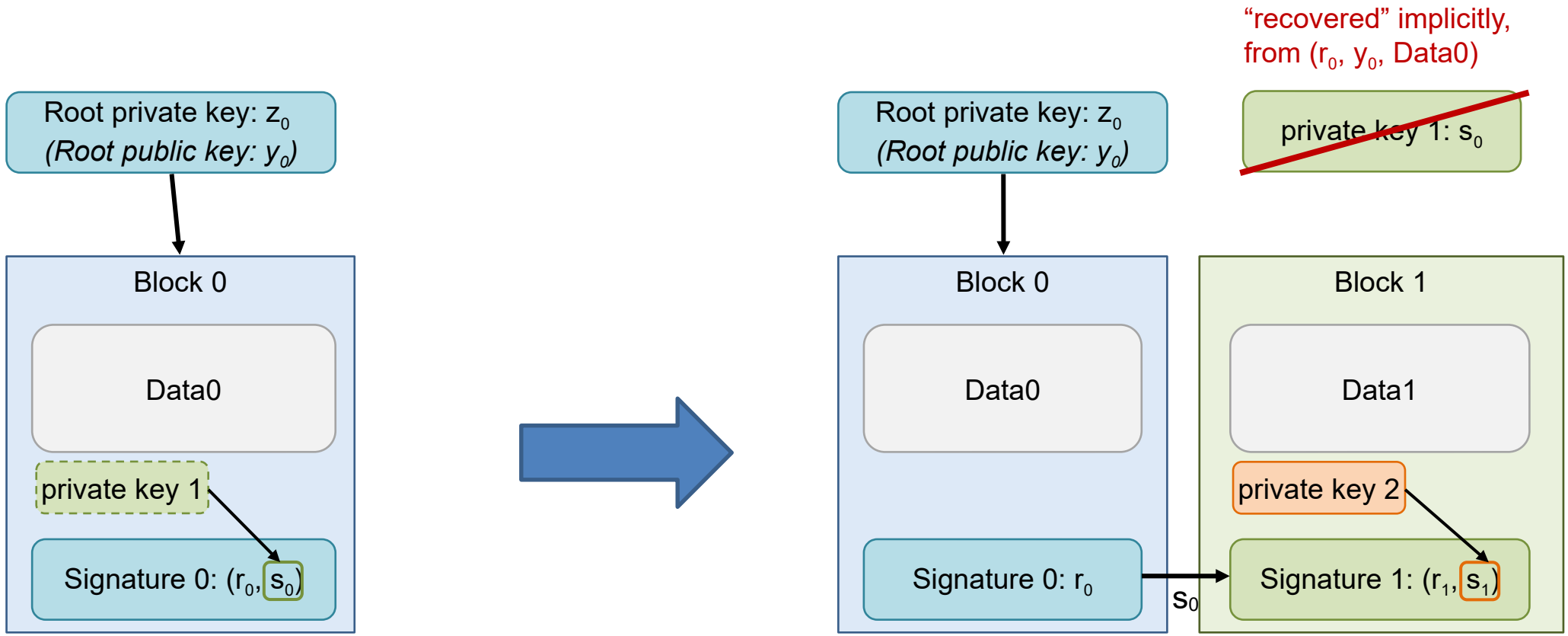
Link between issuer and audience

ID1 → ID2 → ID3 → ID4

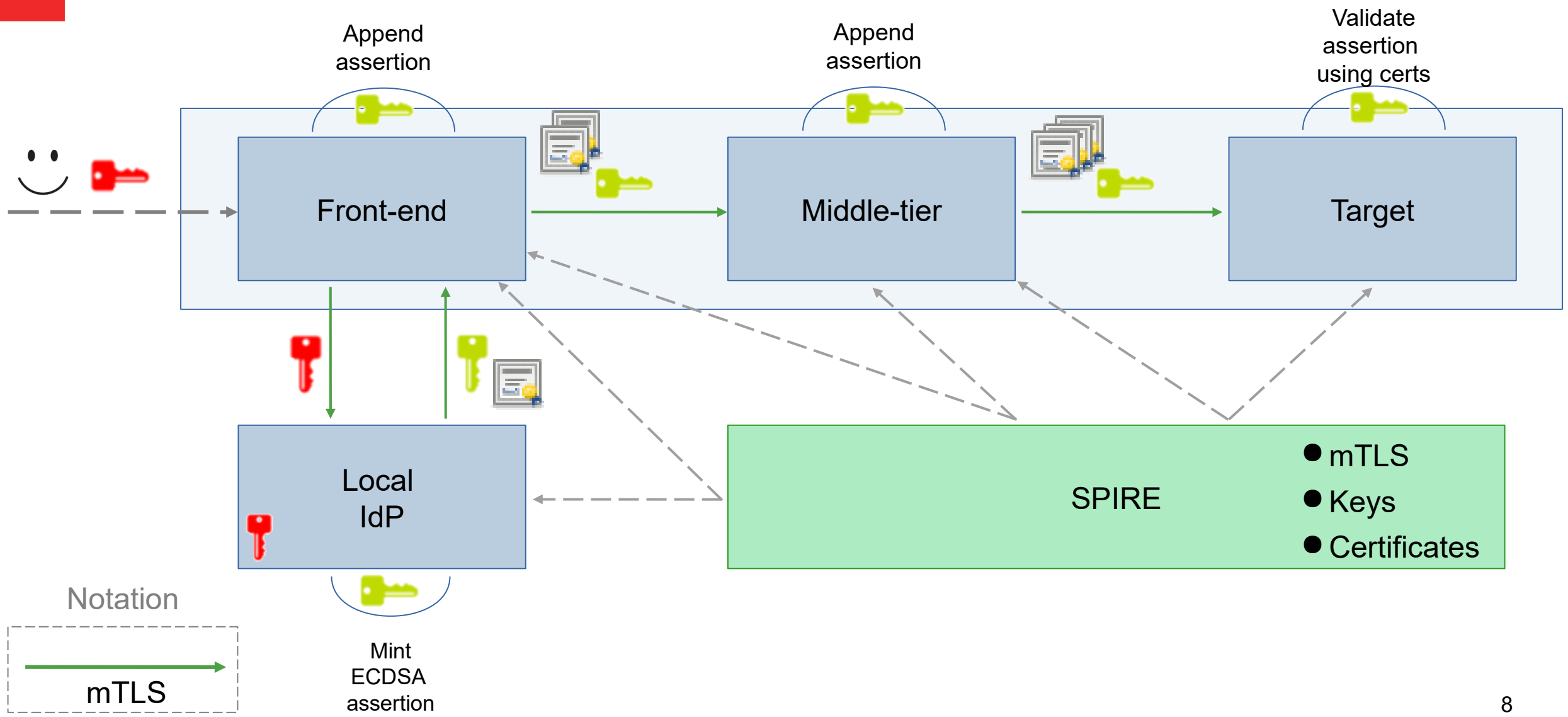


SchCo-Biscuits

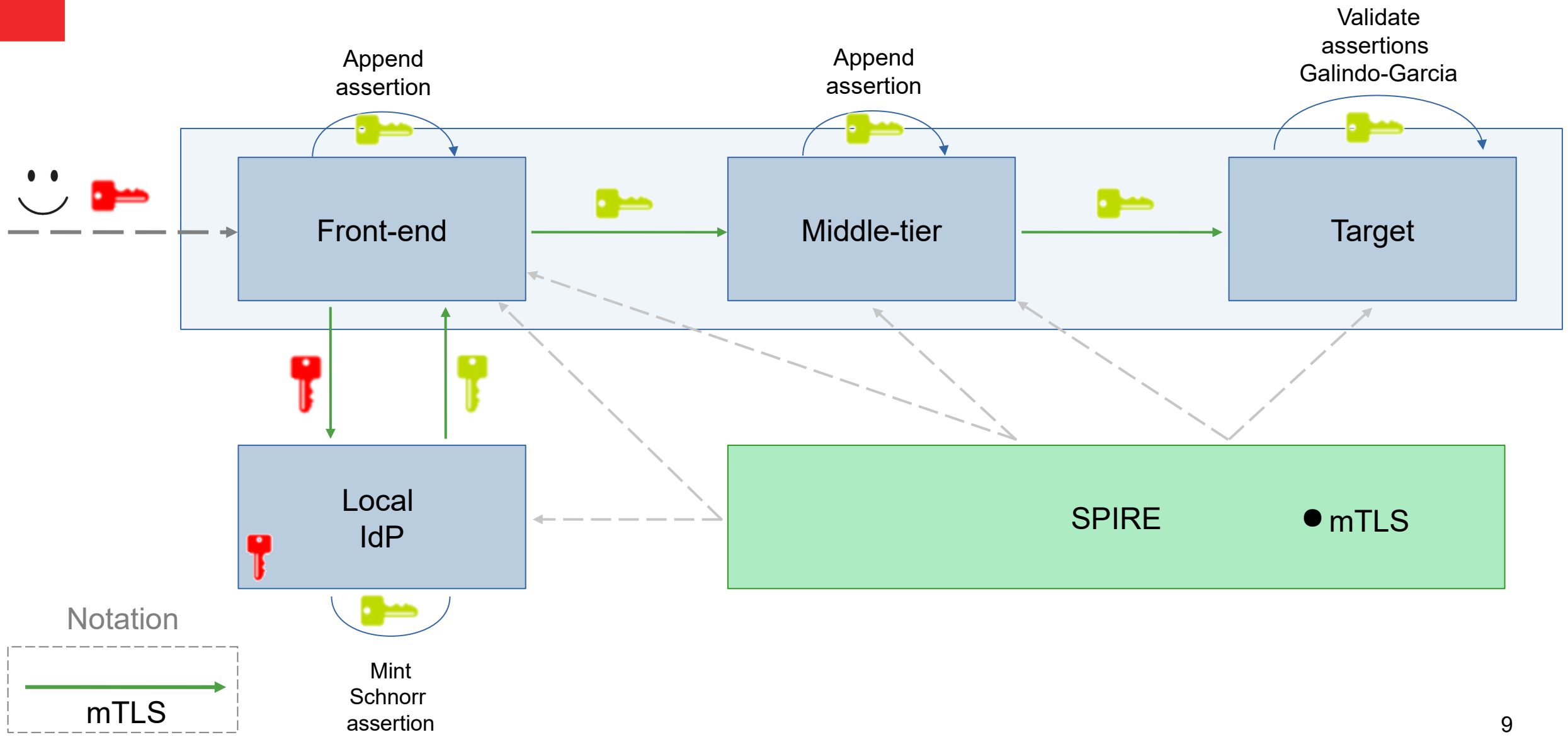
Anonymous mode based in Biscuits model and using concatenated Schnorr signatures (Galindo-Garcia style)



PoC 1: ECDSA – SVID (ID mode)



PoC 2: EdDSA – Schnorr (Anonymous mode)

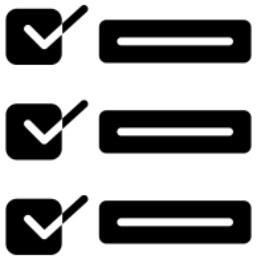


Post-quantum algorithms

Sign with SVID private key and, optionally, with a post-quantum signed algorithm

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

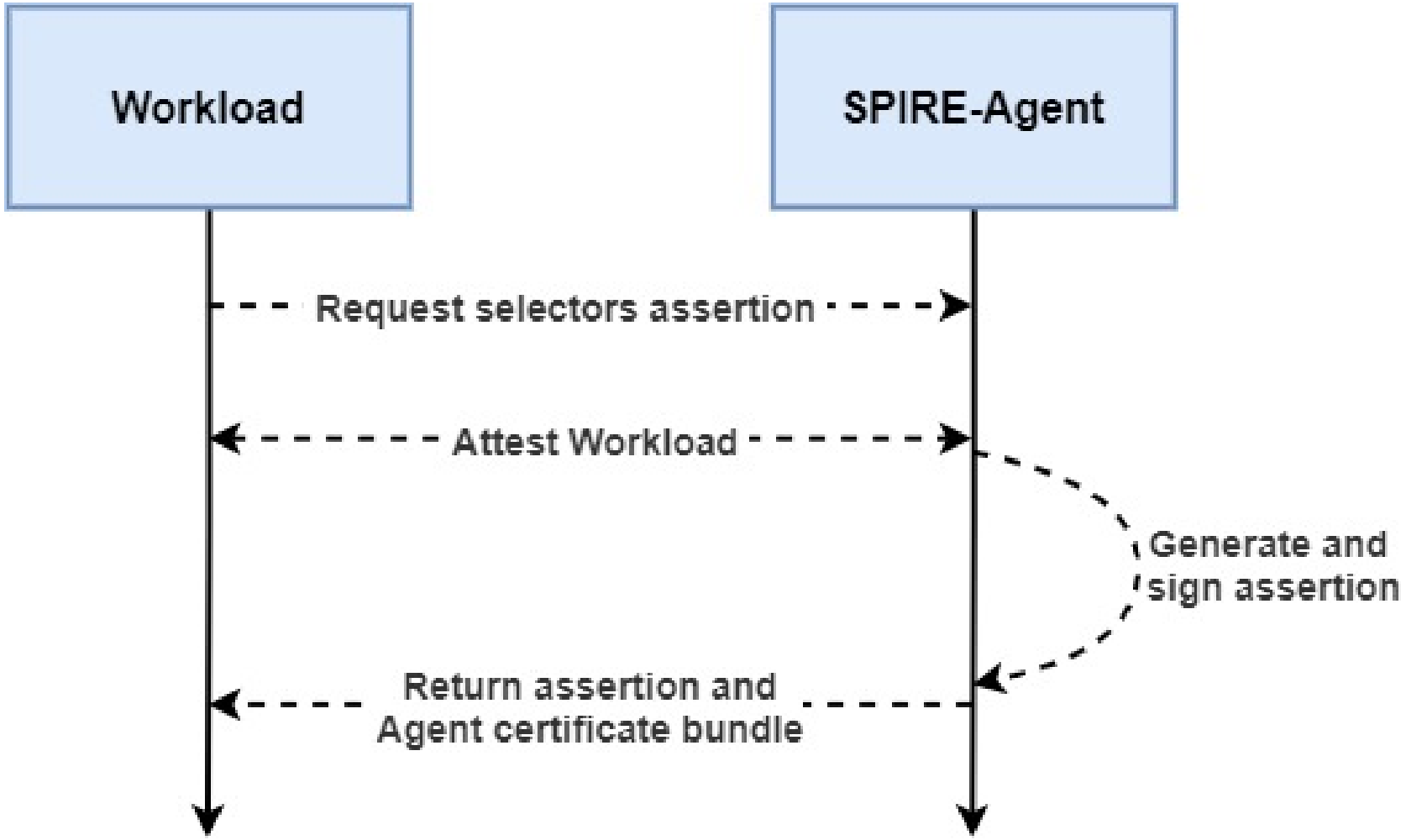
Selector-based Assertion



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

Selector-based Assertion



Selector-based Assertion


```
{
  "iat": 1670634234,
  "iss": "LS0tLS1CRUdJTiBFQyBQVUc57f3bJ1CRUdJTiBFQyBdJTVFQ==",
  "sub": "spiffe://example.org/localuser2",
  "sel": [
    { "type": "unix", "value": "uid:1005" },
    { "type": "unix", "value": "user:subject_wl" },
    { "type": "unix", "value": "gid:1005" },
    { "type": "unix", "value": "group:subject_wl" },
    { "type": "unix", "value": "supplementary_gid:1005" },
    { "type": "unix", "value": "supplementary_group:subject_wl" },
    { "type": "unix", "value": "path:/opt/spire/bin/spire-agent" },
    {
      "type": "unix",
      "value": "sha256:10b90d0d0216fa2e6d467f3870bc57f3bb77c2d9e3fd335c148c9a2ee52fa7b7"
    }
  ]
}
```

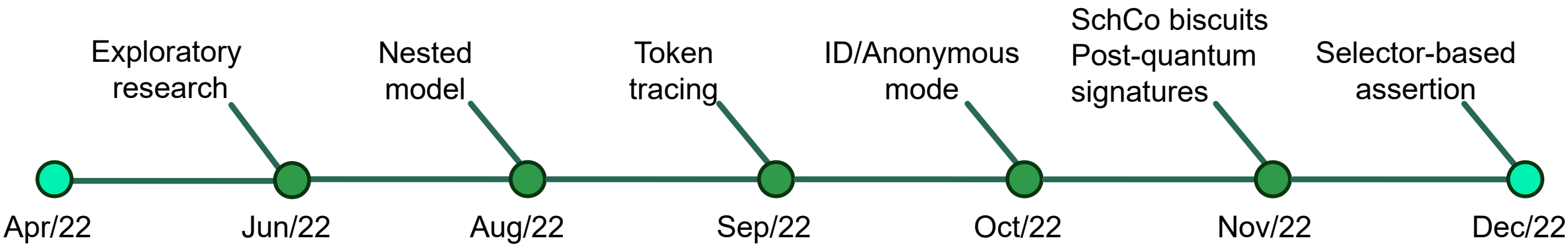
PoC 3: SPIRE-Agent fork and assertgen post-quantum signatures

- Post-quantum assertions: ECDSA + Dillithium signatures
- Selector-based assertions

Developed Proof of Concept

- Distributed application using nested model and token path tracing
  - Multiple workloads (e.g., front-end, middle-tiers) appending assertion
  - Uses ID or Anonymous mode
- SPIRE fork
  - SPIRE-Agent generates workload selector-based assertion
- Assertgen
  - Command-line tool to test developed prototypes

Phase-2: Timeline

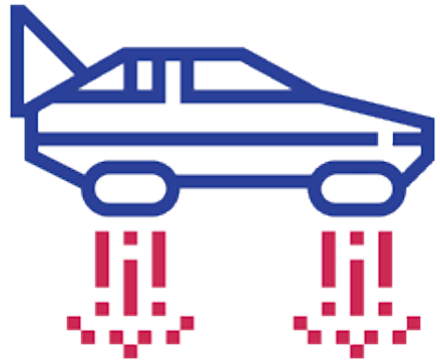


Next steps (2022)

- Add proxy to PoC 1 application scenario
- Finish benchmark paper

Future Work (2023)

- 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?
WHO? WHERE? 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?