Attested CSR

Hannes Tschofenig (University of Applied Sciences Bonn-Rhein-Sieg, Siemens/Germany)

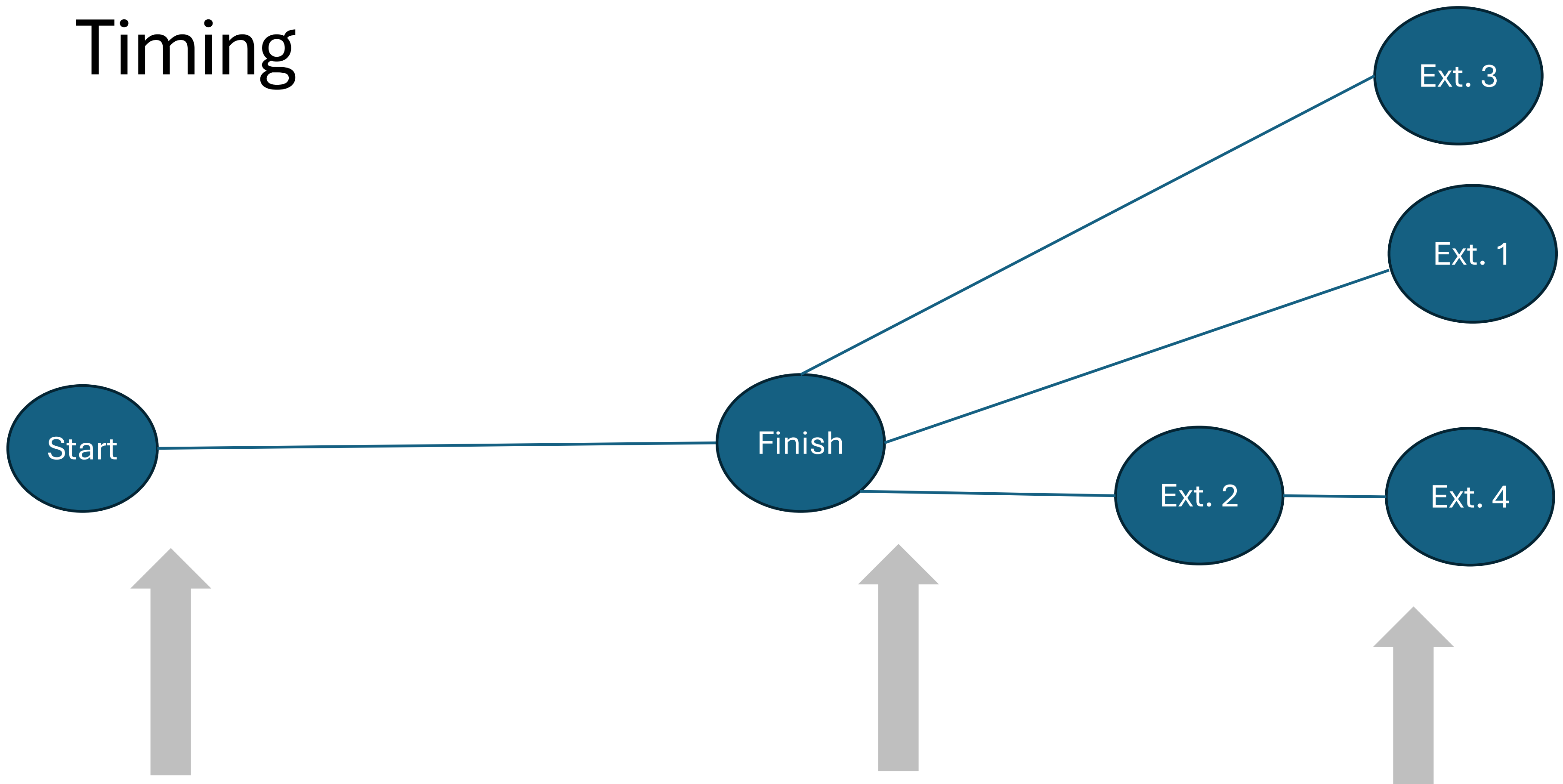
Designing Protocols is complex

- Privacy Considerations – RFC 6973
- Security Considerations – RFC 3552
  - Writing Protocol Models - RFC 4101
  - Pervasive Monitoring - RFC 7258
- Guidelines for Considering Operations and Management – RFC 5706
- Energy efficiency, internationalization, usability and accessibility
- Implementation experience
- Extensibility – RFC 6709, RFC 9413 and RFC 9170
- Deployment success
  - What makes for a successful protocol – RFC 5218
  - Technology adoption – RFC 7305
  - Design expectations and deployment reality – RFC 8980

Formal Methods in Protocol Design

- New: TLS working group requiring formal analysis for any (non-trivial) protocol extension.
  - Recording from IETF#119 (26min into the meeting)
  - Idea: Start with formal analysis very early in the protocol development
  - First candidate: Extended Key Update in TLS
- Influenced by the recently established IRTF Usable Formal Method Research Group: https://www.irtf.org/ufmrg.html
- Many challenges remain: From time pressure to the complexity of protocols with all their options.

Timing



Lots of “churn” in the early design phase.

Higher “reward” for researchers

“just an extension” – nobody wants to do those.

Scope of the analysis

Standardization work is not necessarily organized around the analysis of protocols.

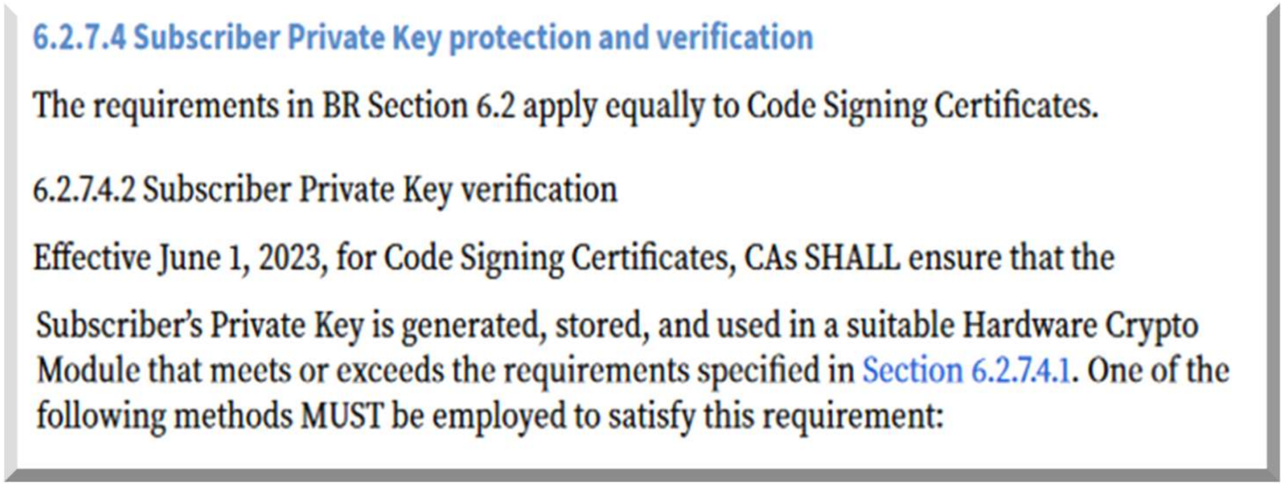
CSR Attestation

What is the attested CSR?

- CSR = Certificate Signing Request
  - PKCS#10 – RFC 2986
  - Certificate Request Message Format (CRMF) – RFC 4211
- IETF LAMPS working group item:
  - https://datatracker.ietf.org/doc/draft-ietf-lamps-csr-attestation/
- Developed in a design team of ~30 persons comprised of
  - HSMs: Entrust, Thales, Utimaco, I4P, Crypto4A, Fortanix, Arm, Intel (TPM)
  - CAs (and CA software vendors): Entrust, Digicert, KeyFactor, Smallstep
  - Users of the technology: Siemens, Bloomberg, Nokia, Ericsson
  - Various IETF, NIST and TCG veterans

CA/B Forum Code Signing Baseline Requirements

- To help prevent code signing keys from "walking away", the CA/Browser Forum instituted a requirement, effective June 1, 2023 that all publicly-trusted code signing keys must be in >= FIPS 140-2 level 2 or CC EAL 4+ hardware.



6.2.7.4 Subscriber Private Key protection and verification

The requirements in BR Section 6.2 apply equally to Code Signing Certificates.

6.2.7.4.2 Subscriber Private Key verification

Effective June 1, 2023, for Code Signing Certificates, CAs SHALL ensure that the

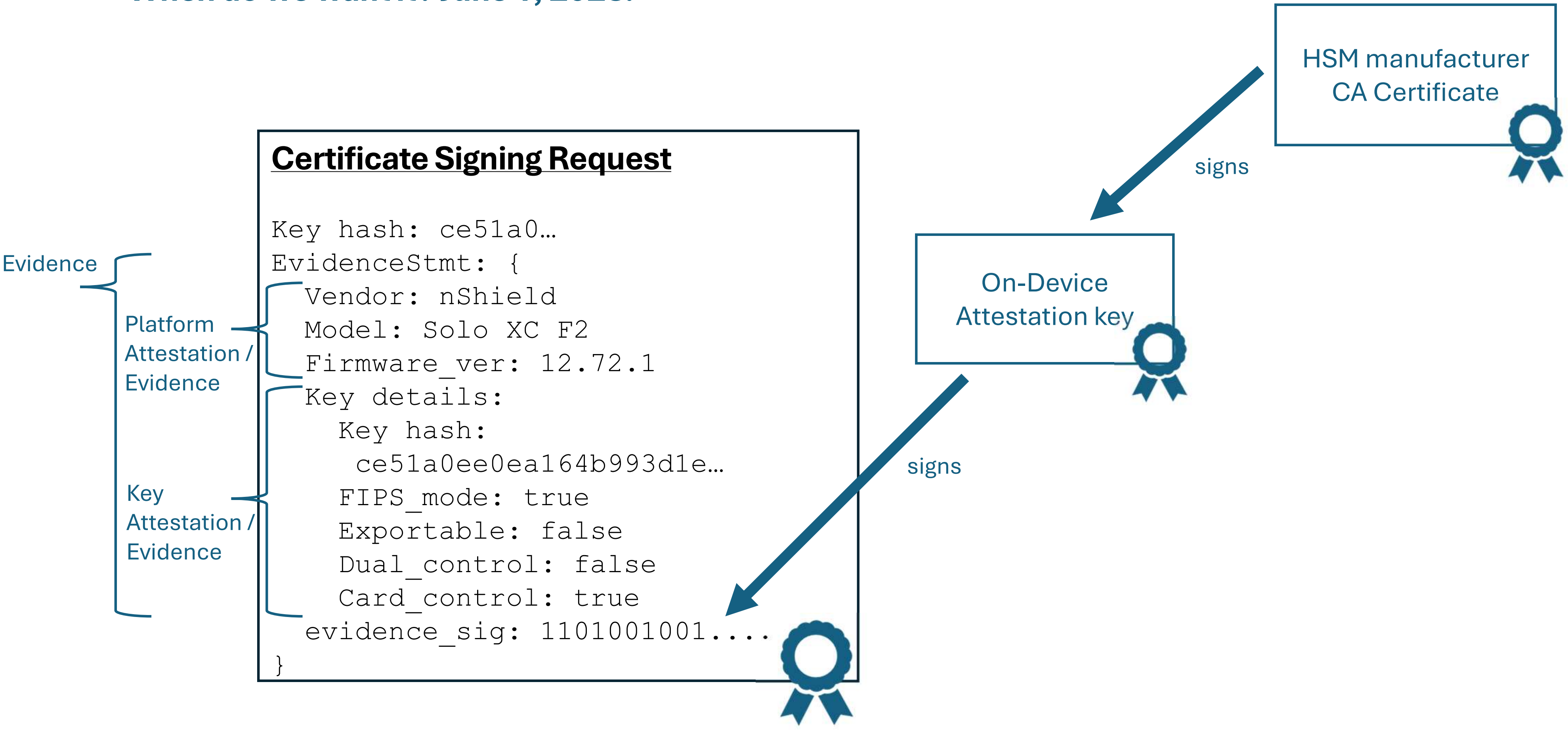
Subscriber's Private Key is generated, stored, and used in a suitable Hardware Crypto Module that meets or exceeds the requirements specified in Section 6.2.7.4.1. One of the following methods MUST be employed to satisfy this requirement:

- **Problem #1**: How is an HSM operator supposed to prove this to a CA?
- **Problem #2**: How is a CA supposed to decide what "evidence" counts and what doesn't?

https://cabforum.org/wp-content/uploads/Baseline-Requirements-for-the-Issuance-and-Management-of-Code-Signing.v3.3.pdf

What do we want? Key attestation!

When do we want it? June 1, 2023!



Certificate Signing Request

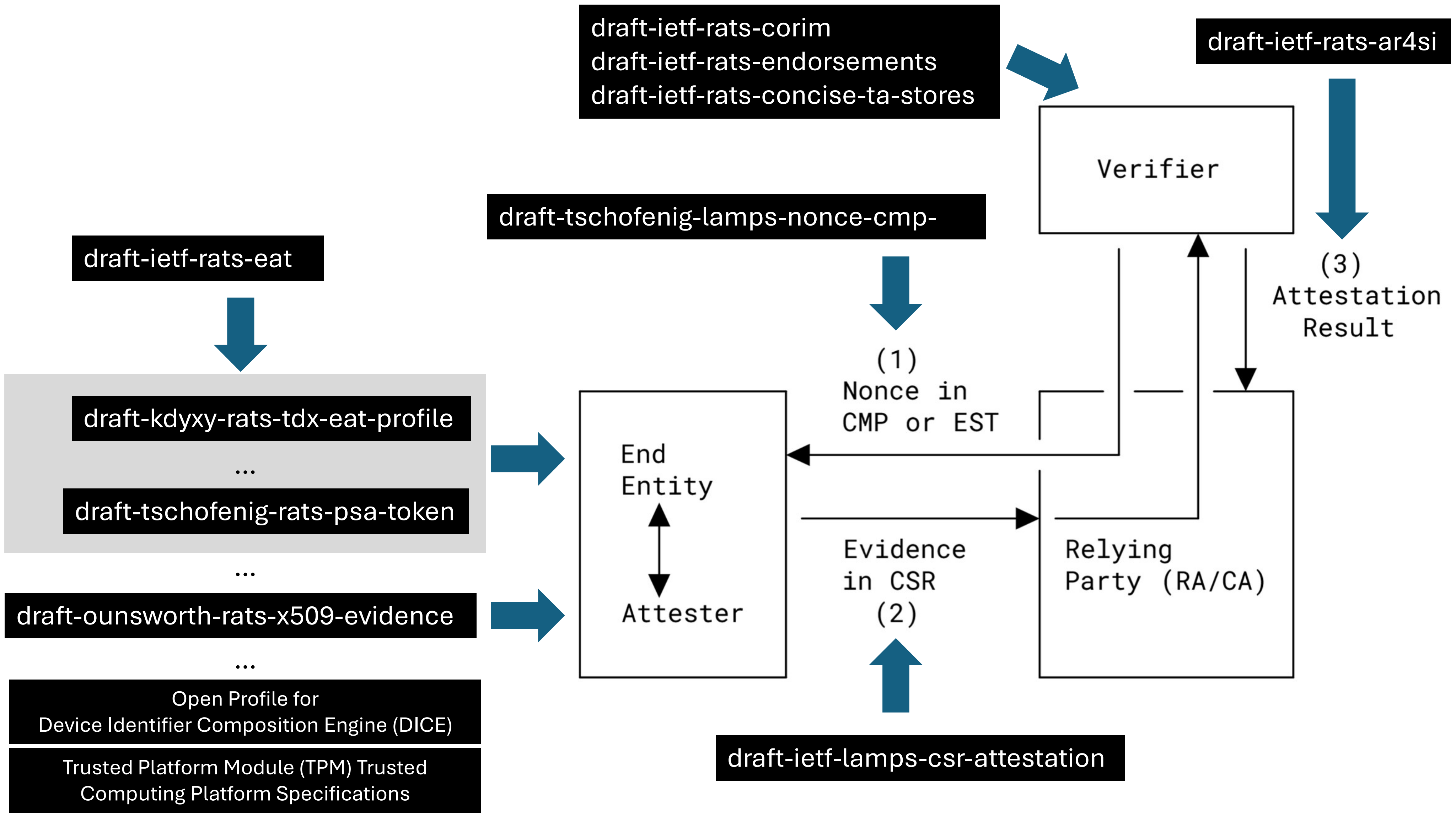
```
Key hash: ce51a0…
EvidenceStmt: {
  Vendor: nShield
  Model: Solo XC F2
  Firmware_ver: 12.72.1
  Key details:
    Key hash:
     ce51a0ee0ea164b993d1e…
    FIPS_mode: true
    Exportable: false
    Dual_control: false
    Card_control: true
  evidence_sig: 1101001001....
}
```

HSM manufacturer CA Certificate
signs
On-Device Attestation key
signs
Evidence
Platform Attestation / Evidence
Key Attestation / Evidence

Status

- Open issues captured at https://github.com/lamps-wg/csr-attestation
  - Open issue related to incomplete TPM example.
- IETF#119 hackathon team produced examples for TPM-based key attestation:
  - https://github.com/mwiseman-byid/csr-attestation-tpm-example
- Draft close to working group last call

From a building block to a system


draft-ietf-rats-corim
draft-ietf-rats-endorsements
draft-ietf-rats-concise-ta-stores
draft-ietf-rats-ar4si
Verifier
draft-tschofenig-lamps-nonce-cmp-
draft-ietf-rats-eat
(3) Attestation Result
(1) Nonce in CMP or EST
draft-kdyxy-rats-tdx-eat-profile
...
draft-tschofenig-rats-psa-token
End Entity
Attester
Evidence in CSR (2)
Relying Party (RA/CA)
...
draft-ounsworth-rats-x509-evidence
...
Open Profile for Device Identifier Composition Engine (DICE)
Trusted Platform Module (TPM) Trusted Computing Platform Specifications
draft-ietf-lamps-csr-attestation

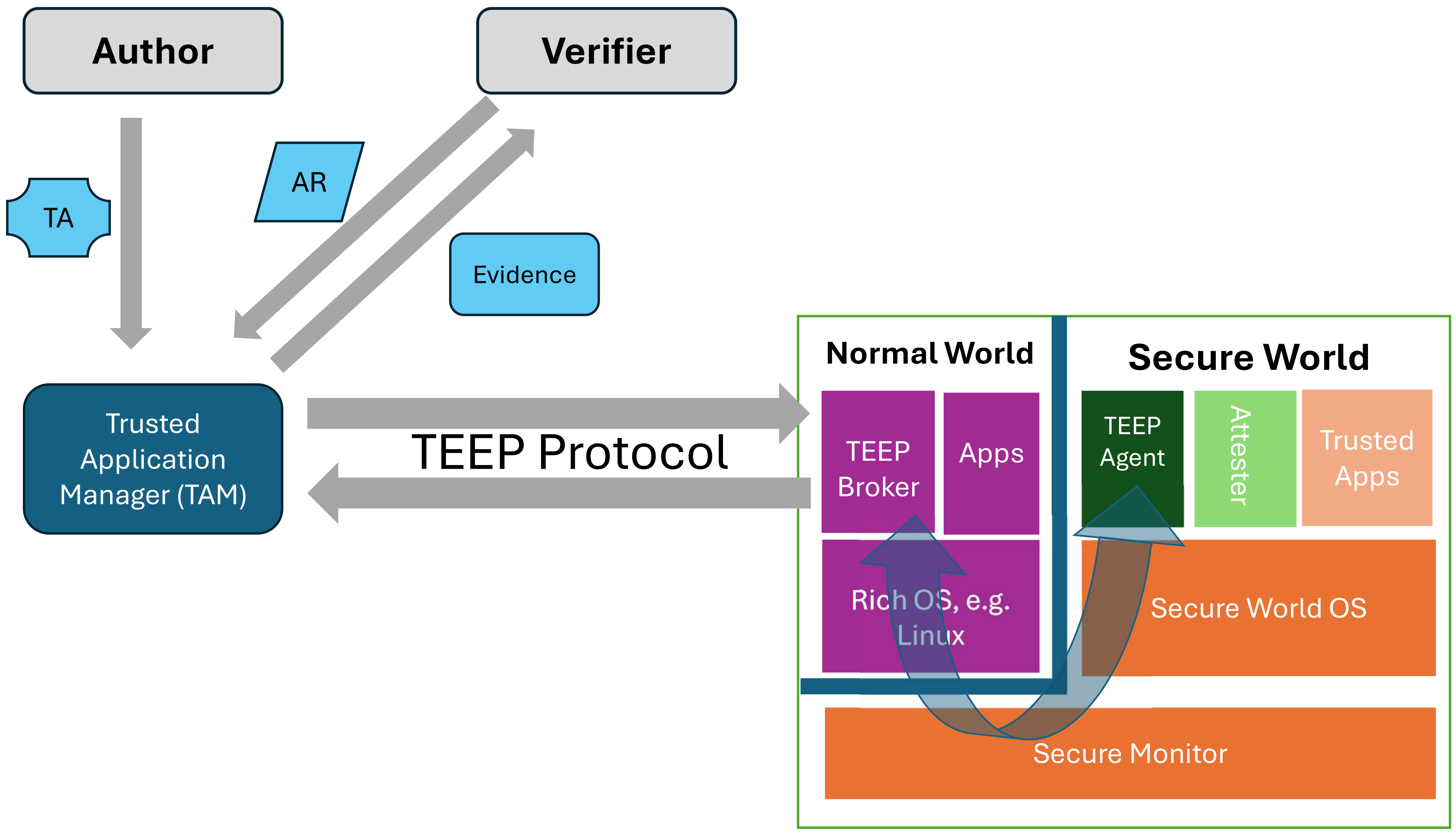
Trusted Execution Environment Provisioning (TEEP)

https://datatracker.ietf.org/doc/draft-mt-ufmrg-teep-sample/
https://datatracker.ietf.org/doc/draft-ietf-teep-protocol/

https://github.com/tetsuya-okuda-hco/public-teep-formal-verif

TEEP in a nutshell

- Allows a TEE to obtain software (Trusted Apps), configuration data and keys from a Trusted Application Manager.
  - Defined as an HTTP-based protocol
- TEE must be attested to TAM, and TEE may attest TAM
  - Uses the IETF RATS architecture for this purpose
- Exchanged data is either signed or signed & encrypted
  - Relies on COSE for security and SUIT for manifest meta-data description
  - Offers several key management options


Author
Verifier
TA
AR
Evidence
Trusted Application Manager (TAM)
TEEP Protocol
Normal World
Secure World
TEEP Broker
Apps
TEEP Agent
Attester
Trusted Apps
Rich OS, e.g. Linux
Secure World OS
Secure Monitor

Attestation in OAuth

Uses key attestation -- but its own version

https://datatracker.ietf.org/doc/draft-ietf-oauth-attestation-based-client-auth/

OAuth

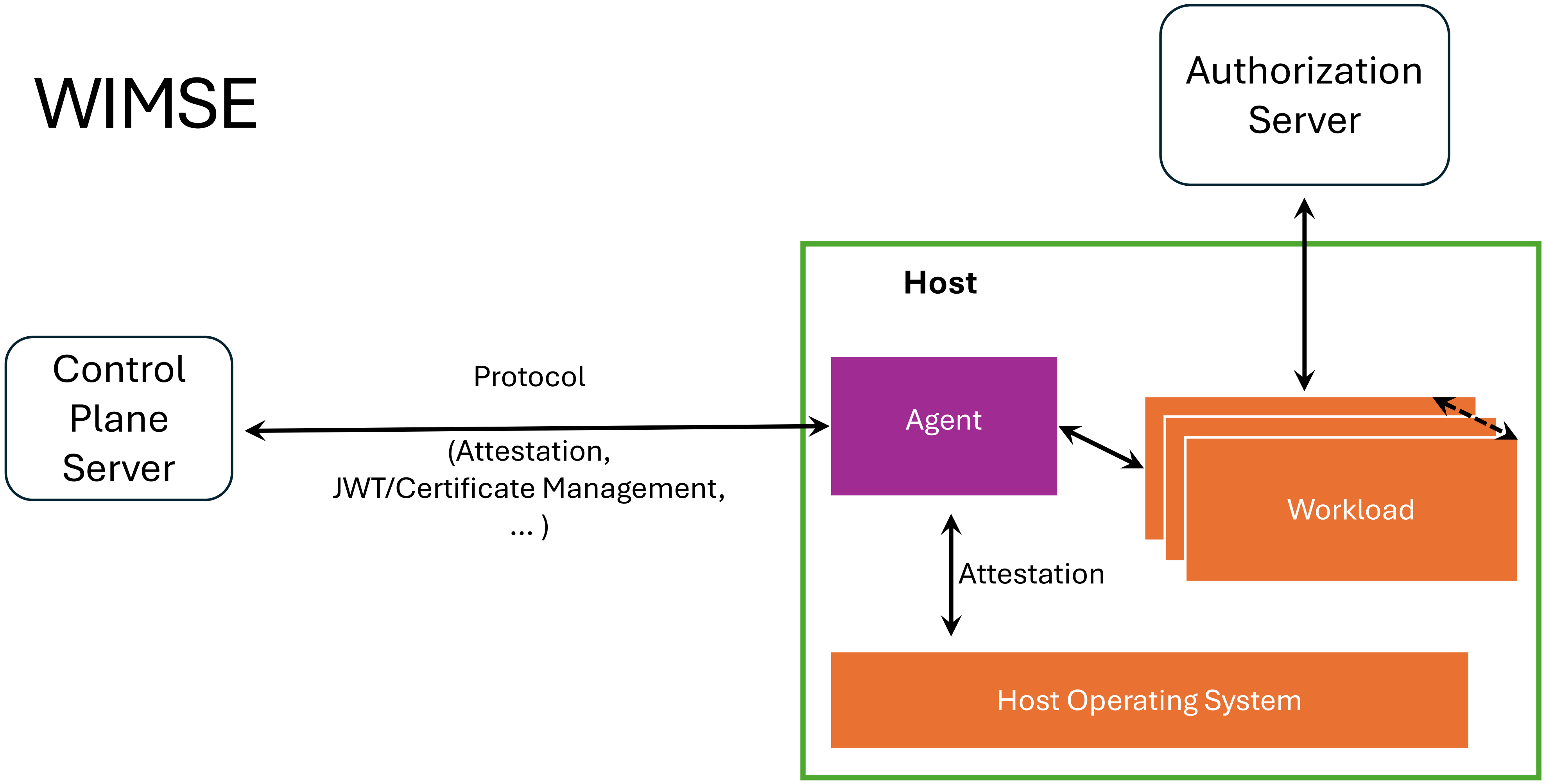
- Long history of formal analysis in OAuth (see https://wiki.ietf.org/group/ufm)
  - Started with OAuth security workshop series: https://oauth.secworkshop.events/
  - Next workshop: https://oauth.secworkshop.events/osw2024
- OAuth has many deployment variants (e.g. web server, browser, interface-constrained devices, native apps on smart phones, etc.)
  - Security model of a browser is different from those of native apps.
- OAuth Attestation conceptually similar to CSR attestation but
  - Uses a different encoding (based on JSON/JWT), and
  - specification is in an early stage.

Workload Identity in Multi-System Environments (WIMSE)

Architecture: https://datatracker.ietf.org/doc/draft-salowey-wimse-arch/

New working group: https://datatracker.ietf.org/wg/wimse/about/

WIMSE


Authorization Server
Host
Control Plane Server
Protocol
(Attestation, JWT/Certificate Management, …)
Agent
Workload
Attestation
Host Operating System

Summary & Outlook

Summary & Outlook

- Standardization community is expected to apply formal methods in their protocol designs.
- Formal analysis has to start early in the protocol design phase.
  - Requires new model for involving researchers in the standardization work and new incentives.
- New examples: WIMSE and OAuth attestation
  - Help needed!
- There are still challenges with the use of formal methods.