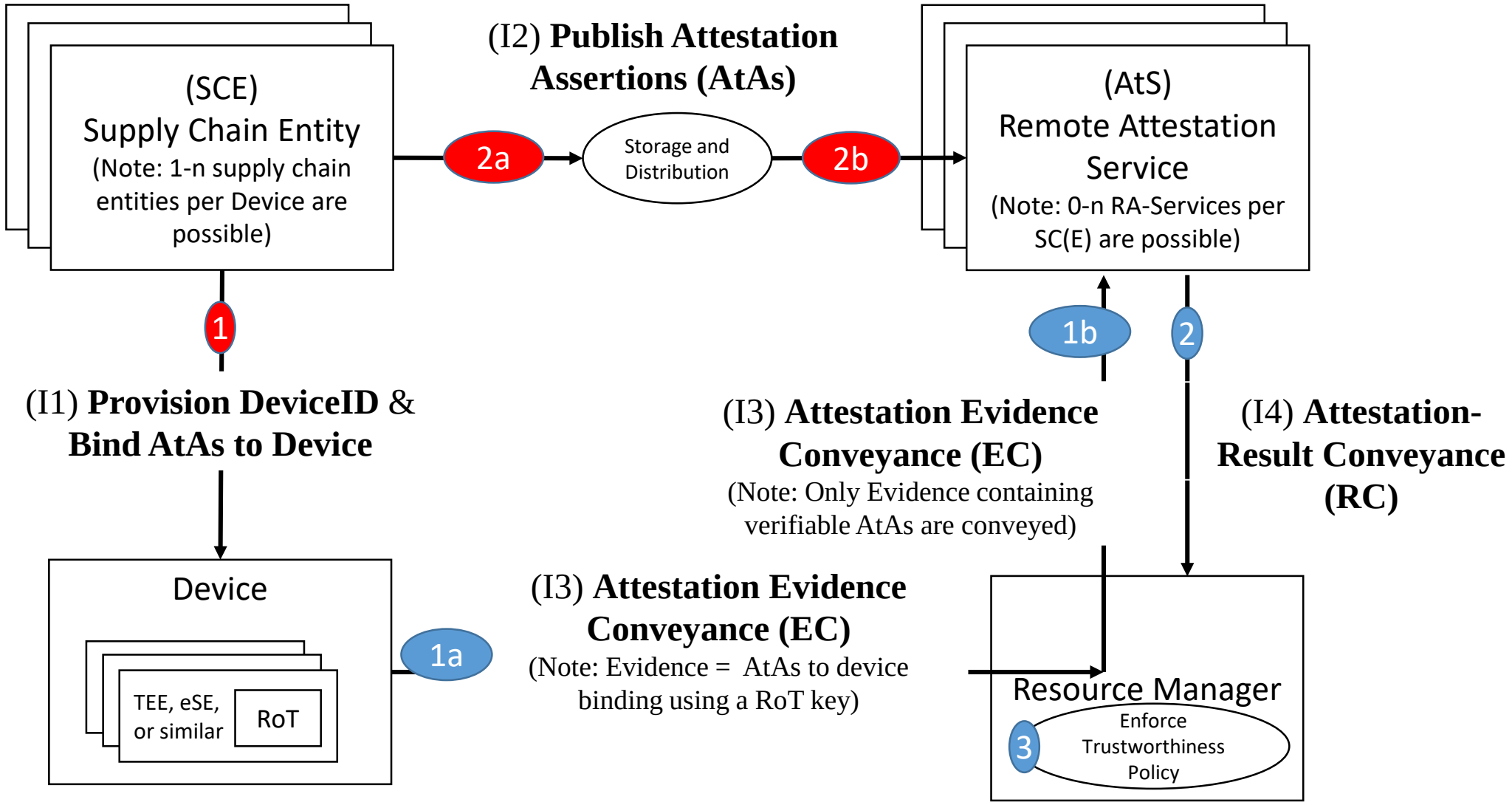
RATS Architecture & Terminology –RECAP–

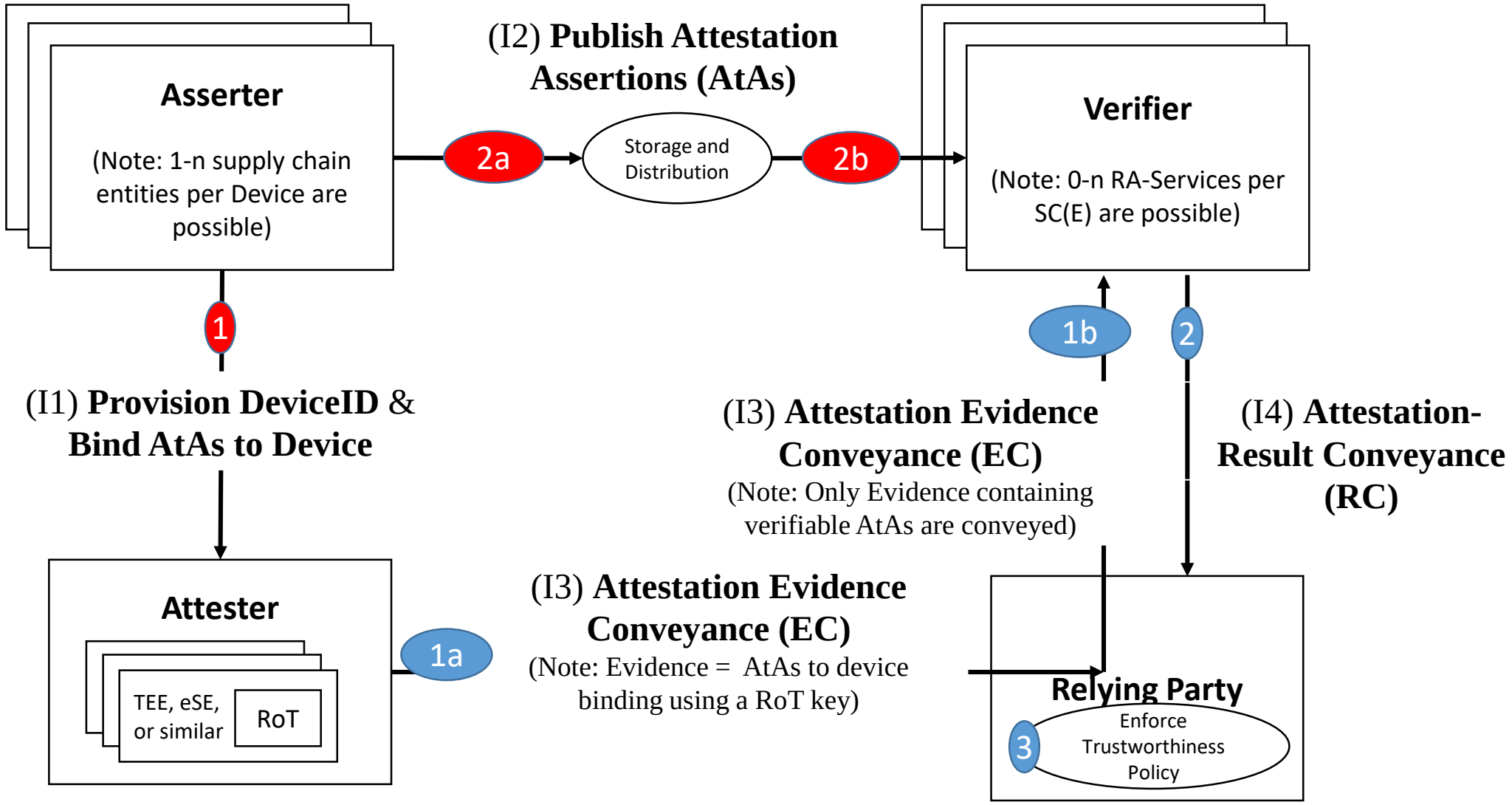
Henk Birkholz {henk.birkholz@sit.fraunhofer.de}
Ned Smith {ned.smith@intel.com}

IETF Virtual Interim, June 20th, RATS WG

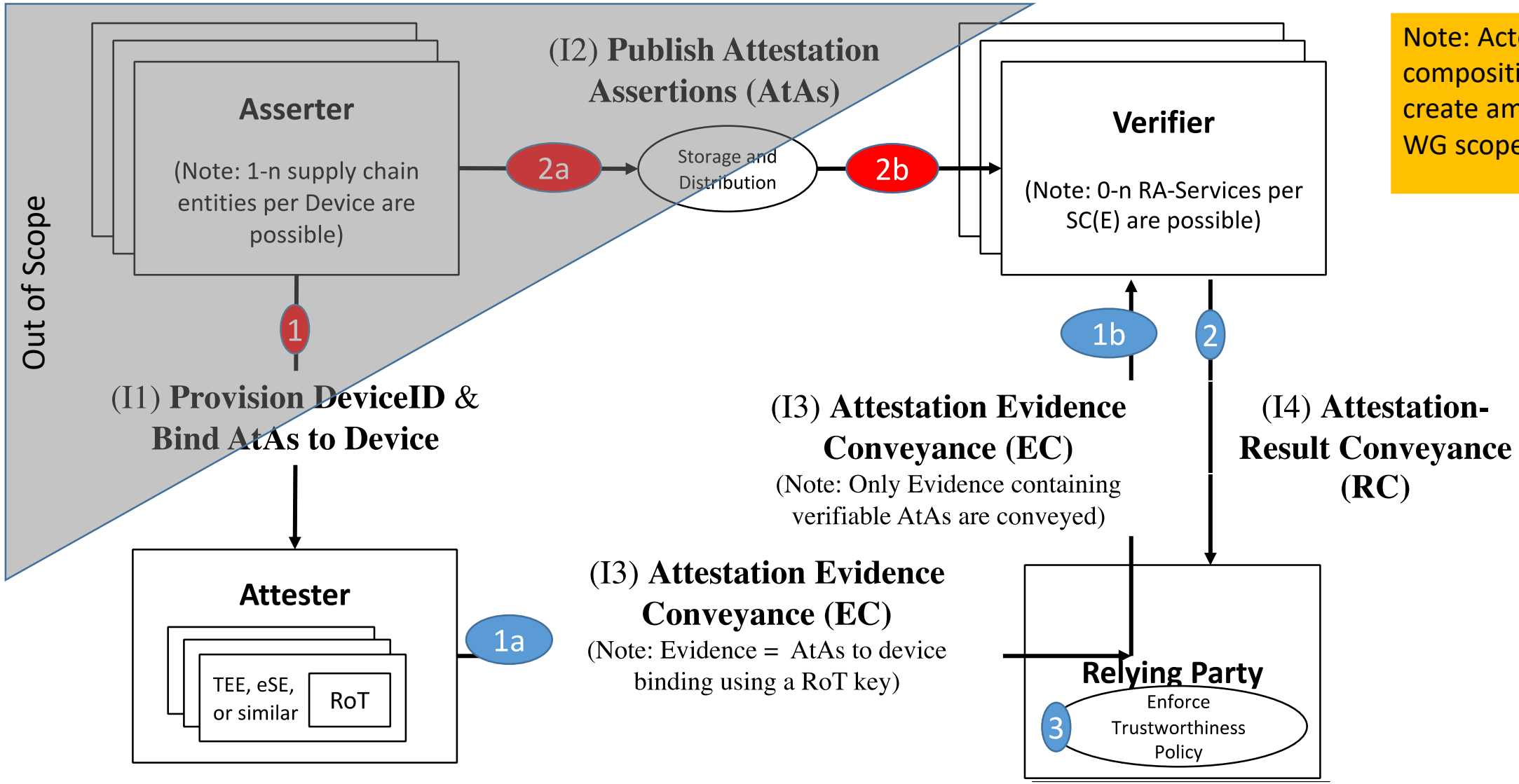
Current RATS Architecture: **Actors**



Current RATS Architecture: **Roles**



RATS WG Scoping



Note: Actor-Role compositions can create ambiguous WG scope scenarios

Overlap with other Working Groups

- **TEEP WG**
  - Trusted Execution Environments (TEE) in **Devices**
  - **Manifest Profiles**
  - TEE **Attestation Provenance** procedures
- **SUIT WG**
  - **Manifest** Format & **Information Model** (approach)
- **SACM WG**
  - Identity **Manifest** & **Information Model** (CoSWID)
- **NETCONF WG**
  - Managed **Trust Anchor** Repository (data at rest)
- **TAMP WG**
  - Protocol for configuring **Trust Anchor** policies (data in motion)

Current RATS Solution I-D, Data Models & Serializations

(and the types of roots-of-trust involved)

Henk Birkholz {henk.birkholz@sit.fraunhofer.de}

Michael Eckel {michael.eckel@sit.fraunhofer.de}

IETF Virtual Interim, June 20th, RATS WG

The Entity Attestation Token (current state)

- There is consensus on the list that **EAT are CWT**.
- EAT are a subset of CWT **defined by the claims** included in a CWT. Corresponding claims are defined by the EAT document.
- EAT are created by **Attesters**/Devices, typically using flavors of **Roots-of-Trust**.
- EAT are consumed by **Relying Parties**/Resource Managers or corresponding **Verifiers**/Remote Attestation Services, using **Trust Anchors**.
- https://datatracker.ietf.org/doc/draft-mandyam-rats-eat/
- https://github.com/eat-ietf-wg/eat

Arm's Platform Security Architecture (PSA) Attestation Token

- **PSA** are based on **EAT** (and therefore also use the CWT structure).
- PSA require the use of **EAT claims**: nonce and UEID.
- PSA Tokens are Attestation Tokens because they are used in Arm's **attestation API** of the Arm Platform Security Architecture.
- The PSA **Firmware Framework** makes uses of Root of Trust **security services** for secure applications.
- https://datatracker.ietf.org/doc/draft-tschofenig-rats-psa-token/

Time-Based Uni-Directional Attestation (TUDA)

- TUDA messages **are not using the CWT** structure.
- Message composition is very similar to the **CBOR Profile for X.509**
  - effectively a message “compression” using nested arrays
  - able to convey non-CBOR native structures via CBOR
  - requires canonical decomposition and recomposition to enable signature validation
- TUDA messages **do not require a nonce** and can provide trustworthy evidence about past operational state of an Attester.
- TUDA requires a **remote source of time** that is trusted and synchronized in a given scope (trust domain).
- TUDA requires several **Roots-of-Trusts**, mainly: for **Measurement**, for **Storage** and **Integrity**, and for **Reporting**.
- https://datatracker.ietf.org/doc/draft-birkholz-rats-tuda/

Remote Attestation YANG Module

- Provides Datastore and RPC statements for a YANG Server **running on an Attester**.
- The Challenge/Response procedures initiated by the **Verifier** require the **use of a nonce** and provide confidentiality via the use of **SSH** or **(D)TLS**.
- The protocols NETCONF, RESTCONF, and CORECONF provide serialization capabilities for **XML**, **JSON**, and **CBOR**.
- **Roots-of-Trusts Services** are provided by a set of **TPM-TSS** API: **SAPI, ESAPI, FAPI**. Corresponding RPC statements are specific to these API.
- https://github.com/ietf-rats/draft-birkholz-rats-basic-yang-module
- https://github.com/tpm2-software/tpm2-tss

Reference Remote Attestation Interaction Model

- **Nonce-based challenge/response** remote attestation procedures are used quite **frequently**.
- Alas, they are often **poorly documented** or deviate in vital details
- In order not to re-specify the same common interaction model (as it is used, for example, by the RATS YANG module), the intention of this I-D is to **avoid these inconsistencies** in the future and enable better interoperability by providing **a single reference**.
- Why is this I-D mentioned in this “solution” slide-deck?
  - The current editor’s version includes a **proof-of-concept example** of how to use the Reference Model. The example is based on **CoAP/CDDL/CBOR**.
- https://ietf-rats.github.io/draft-birkholz-rats-reference-interaction-model/draft-birkholz-rats-reference-interaction-model.html

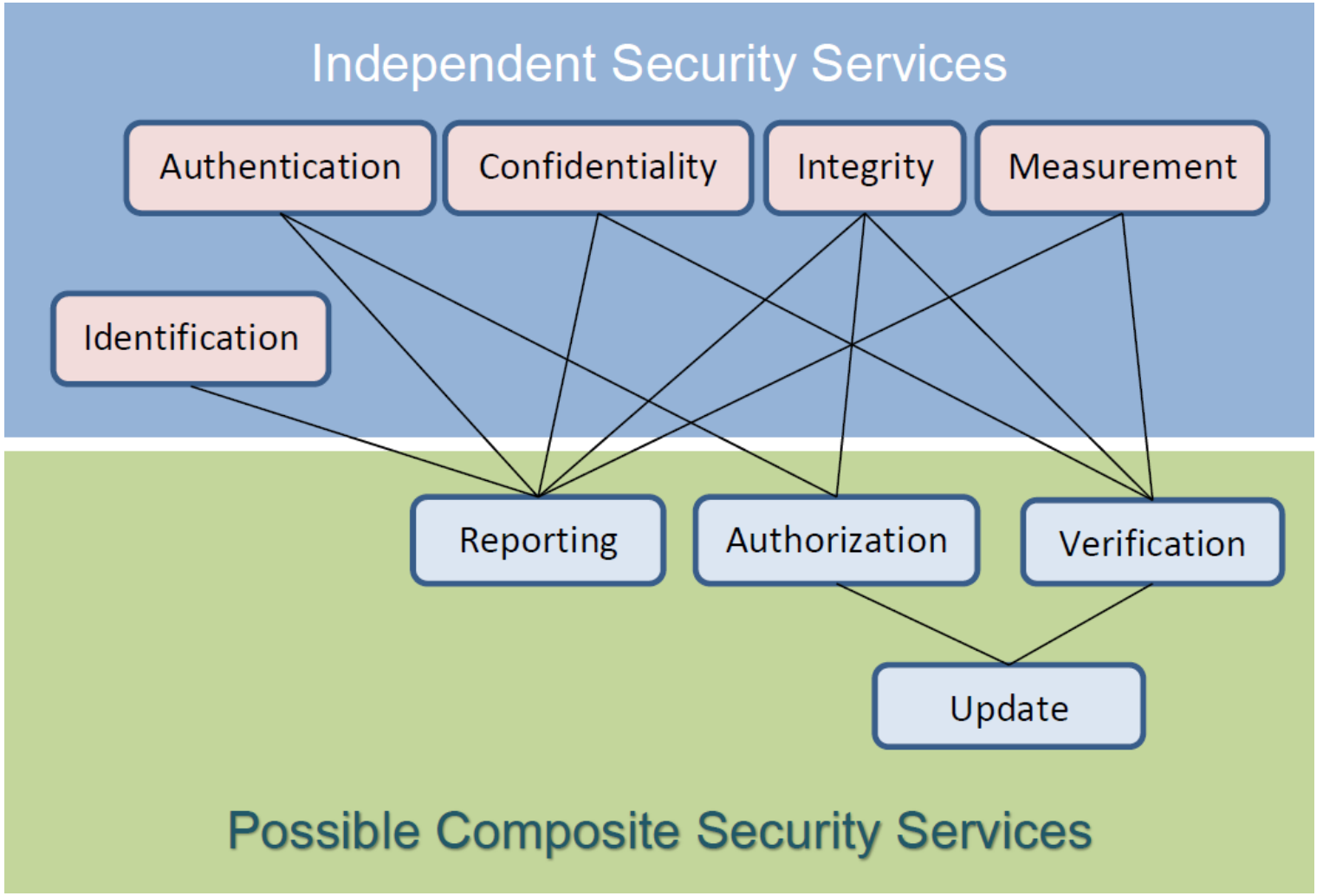
Quick Ratholing on Types of Roots-of-Trust

- Typically RATS require Roots-of-Trust.
- Their main characteristic is that you can only choose to trust them – or not – because:
  Roots-of-Trusts are a set of unconditionally trusted functions that must always behave in an expected manner because their misbehavior cannot be detected.
- Prominent examples of entities defining Roots-of-Trust are NIST, GlobalPlatform, or the Trusted Computing Group.
- A section elaborating on RoT and referencing the current state-of-the-art will be added to the RATS architecture I-D.
- Two examples about references in the next slides….

NIST SP 800-164 (draft)

- Root of Trust for **Storage** (RTS) provides a protected repository and a protected interface to store and manage keying material.
- Root of Trust for **Verification** (RTV) provides a protected engine and interface to verify digital signatures associated with software/firmware and create assertions based on the results.
- Root of Trust for **Integrity** (RTI) provides protected storage, integrity protection, and a protected interface to store and manage assertions.
- Root of Trust for **Reporting** (RTR) provides a protected environment and interface to manage identities and sign assertions.
- Root of Trust for **Measurement** (RTM) provides measurement used by assertions protected via the RTI and attested to with the RTR.

Global Platform – RoT Definitions & Requirements



Serialization of Data Models (current state)

- The following I-D use CBOR
  (and are using CDDL notation or CBOR diagnostic notation):
  - I-D.mandyam-rats-eat
  - I-D.tschofenig-rats-psa-token
  - I-D.birkholz-rats-tuda
  - I-D.birkholz-rats-reference-interaction-model
- EAT & PSA use CWT/COSE as a basis
  - Complementary CDDL specifications would simplify the potential use of JSON/JOSE
- The RATS YANG Module potentially could use CBOR using the CoRECONF I-D (I-D.ietf-core-comi), but running code is still at early stages and XML or JSON serialization are therefore more likely to be expected.

Calls for Adoption

- The time period of the Call for Adoption wrt to EAT and the corresponding TOKBIND I-D is in the past now:
  - Question to the WG: What is the current status?
- The authors of the RATS Basic YANG Module would like to initiate a Call for Adoption quite soon:
  - Question to the WG: If the latest comments and contributions are addressed and incorporated accordingly (which will be done before submission cut-off), when would be a good time to start a Call for Adoption?

RATS Information Model I-D

Henk Birkholz {henk.birkholz@sit.fraunhofer.de}
Ned Smith {ned.smith@intel.com}

IETF Virtual Interim, June 20th, RATS WG

Purpose of the RATS Information Model (IM)

- Every solution I-D defines assertions, such as, attributes, enumerations, claims or structures with specific semantic meaning.
- All these definitions serve a specific “attestation purpose”, for example, to identify attestation provenance.
- The RATS WG intends “to standardize an information model for assertions/claims which provide information about system components characteristics scoped by the specified use-cases” (charter item 3).
- In contrast, the RATS Architecture needs to build consensus on a core vocabulary, which is not the purpose of the IM.

A proposal on how to start the RATS IM I-D

- Pulling all Information Element definitions from the Reference Interaction Model I-D and adding them to the IM I-D (as they do not belong in the former)
- Copying and referencing the English textual description of the assertions defined by EAT, PSA, and other emerging token flavors
- Deriving missing information elements from the quickly evolving use-case I-D
- Classifying/Annotating Information Elements, e.g., by:
  - root-of-trust primitives required,
  - differentiating verifiable and non-verifiable assertions, or by
  - differentiating application-specific assertions and platform-specific assertions

Not quite about the RATS IM, but close…

- A question to the RATS WG:

  How do we plan to proceed with the registration of remote attestation specific claims to be used in CWT, in general?

This is the last slide