

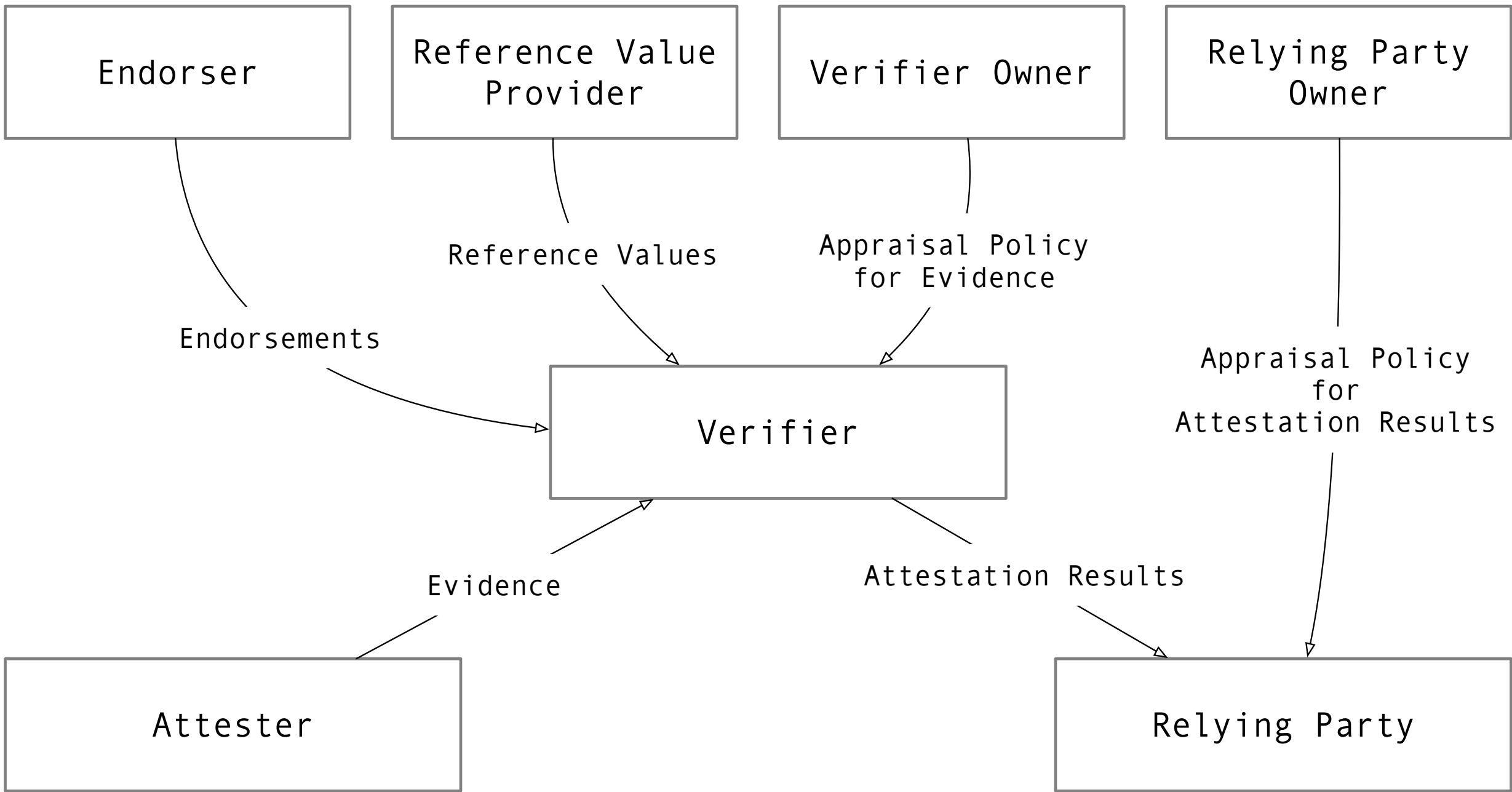
Describing Attesters to Verifiers

Ned Smith, Intel

Yogesh Deshpande, Arm

Henk Birkholz, SIT Fraunhofer

Thomas Fossati, Arm



RATS-ARCH, Conceptual Data Flow

Problem Statement

One or more authorized supply chain actors (OEM, ISVs, SiPs, etc.) need to come together and “describe” an Attester to a Verifier. So, when Evidence from that Attester is passed on to the Verifier, it can use the attributes that apply to the Attester to evaluate Evidence against the Appraisal Policy

Standard IM/DM is likely to lead to standard tooling – reduce fragmentation, lower barrier to entry for the supply chain actors

Problem Statement (cont)

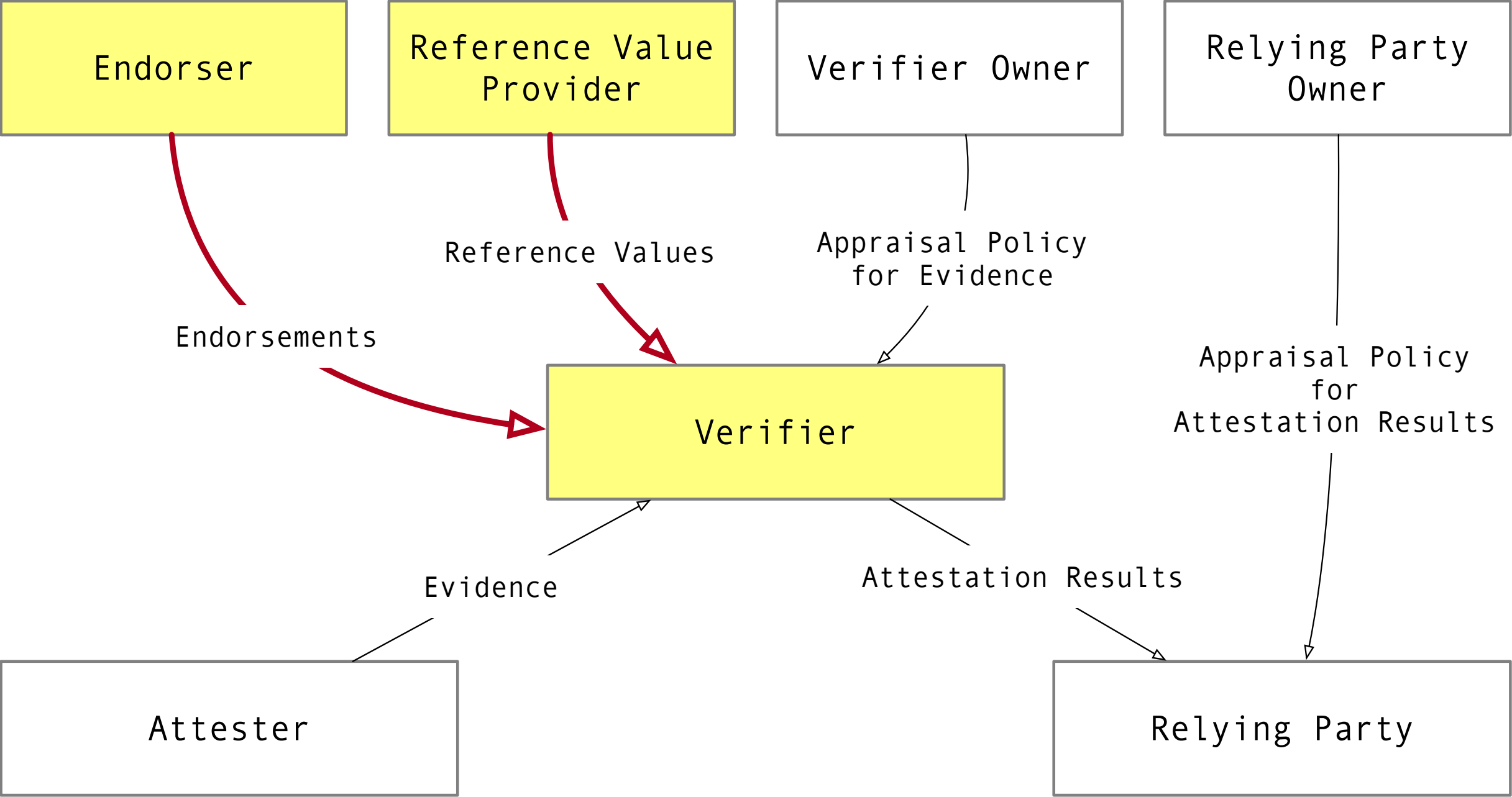
The descriptive material that flows from the supply chain to the Verifier can be, for example:

- Verification key material, certification status – “Endorsements”
- Measurements, for example, FW – “Reference Values”

But we also want to be able to describe the composition of an Attester from its relevant parts (i.e., its Attesting and Target Environments)

- This is not necessary for very simple attesters (AE:TE=1:1), but can come in handy for more complex topologies where the device structure is reflected in the Evidence structure
- Also, it can be useful for factoring out common parts that are reused across different attesters

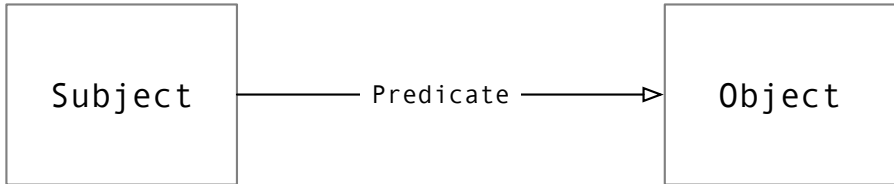
Out of scope – at least for the moment – is the delivery of Verification Policies to the Verifier by the Verifier Owner.



RATS-ARCH, Conceptual Data Flow

High level design

- Graph-based data model (RDF-like) with its own specialized vocabulary and data types

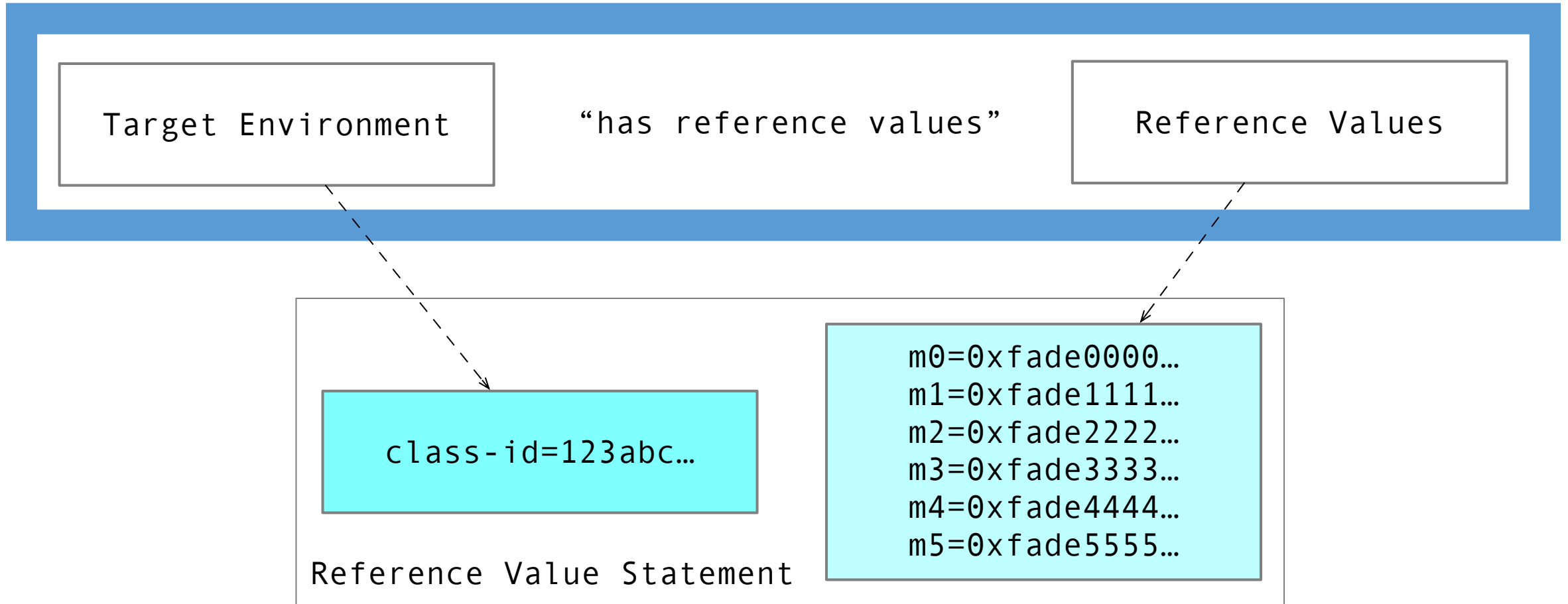
- The “triple”  is the core construct

- Used to define a “Device / Attester” ontology
- Tracking triples provenance via explicit cryptographic means
- **Compact representation (CoRIM, CoMID)**

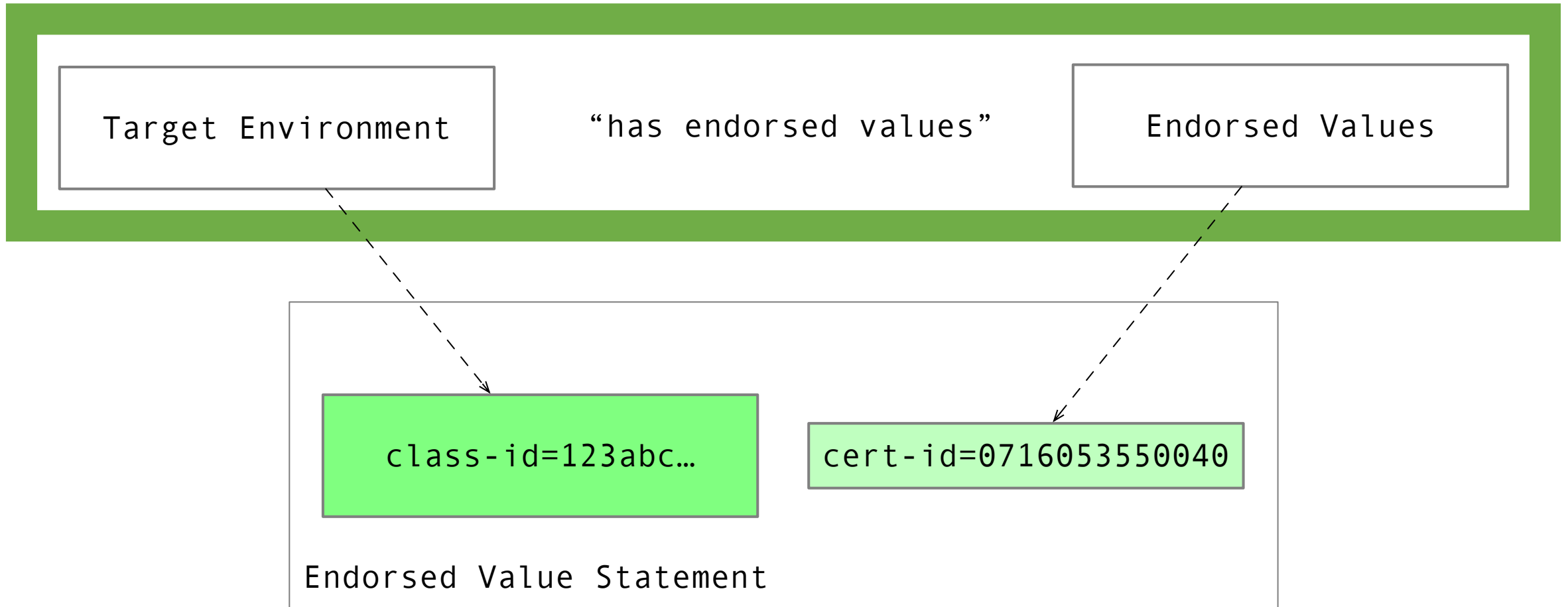
What kind of Triples do we need?

- Reference Values associated with a Target Environment
 - Endorsements associated with an Attesting or a Target Environment
 - Cryptographic identities associated with Attesting Environments
 - Decomposition of a device in its constituent Attesting and Target Environments and their relational features
 - Other that we haven't yet anticipated (built-in extensibility)
-
- Examples (see next)

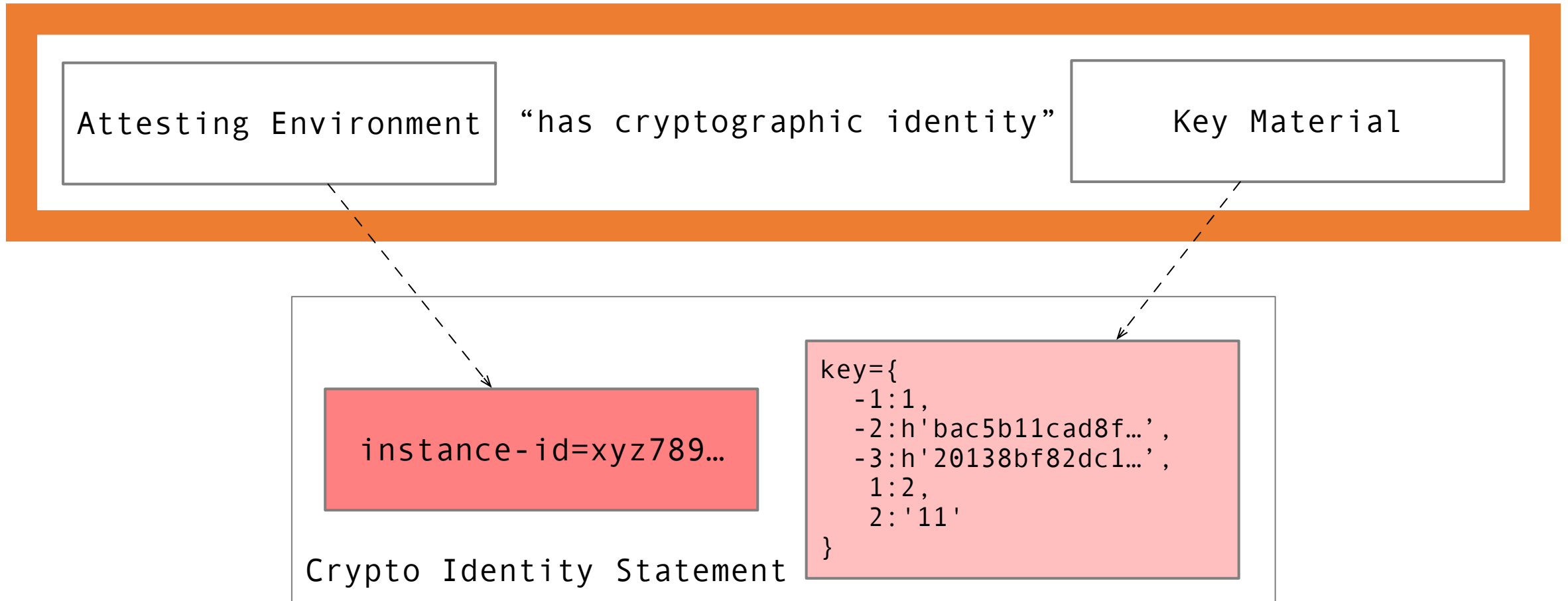
Reference Value Statement



Endorsed Value Statement

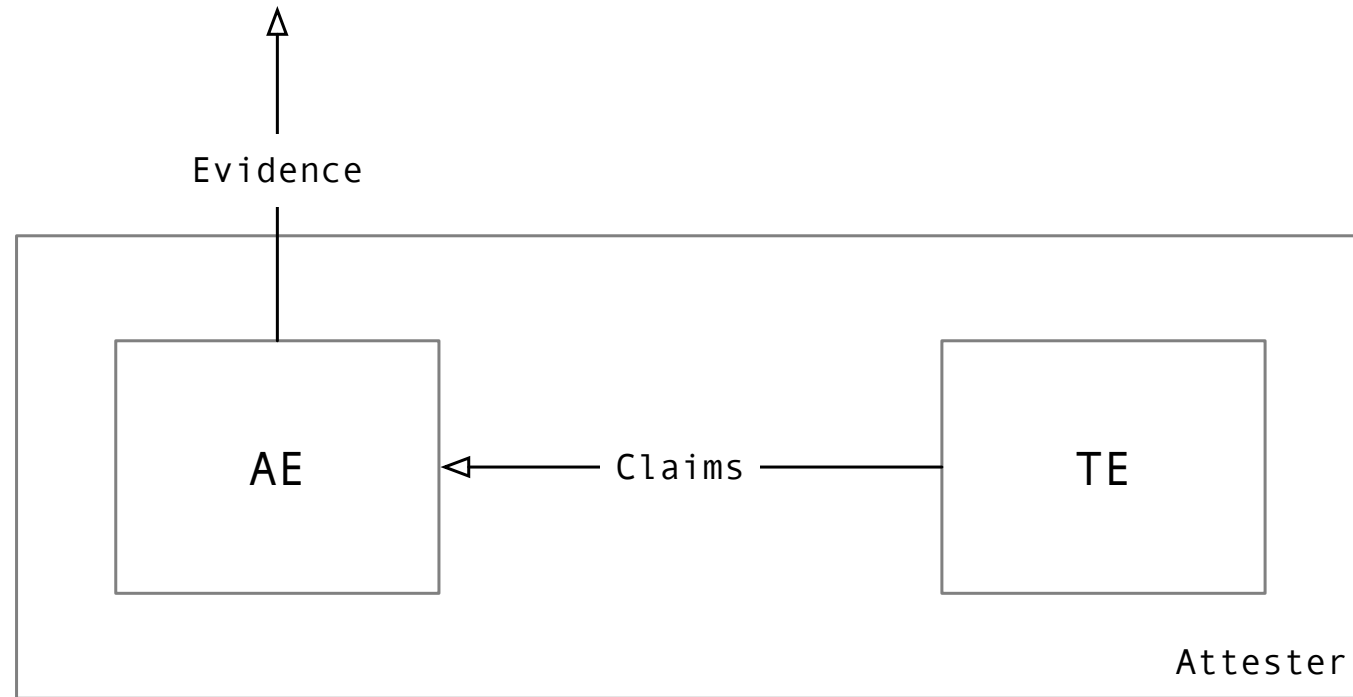


Cryptographic Identity Statement



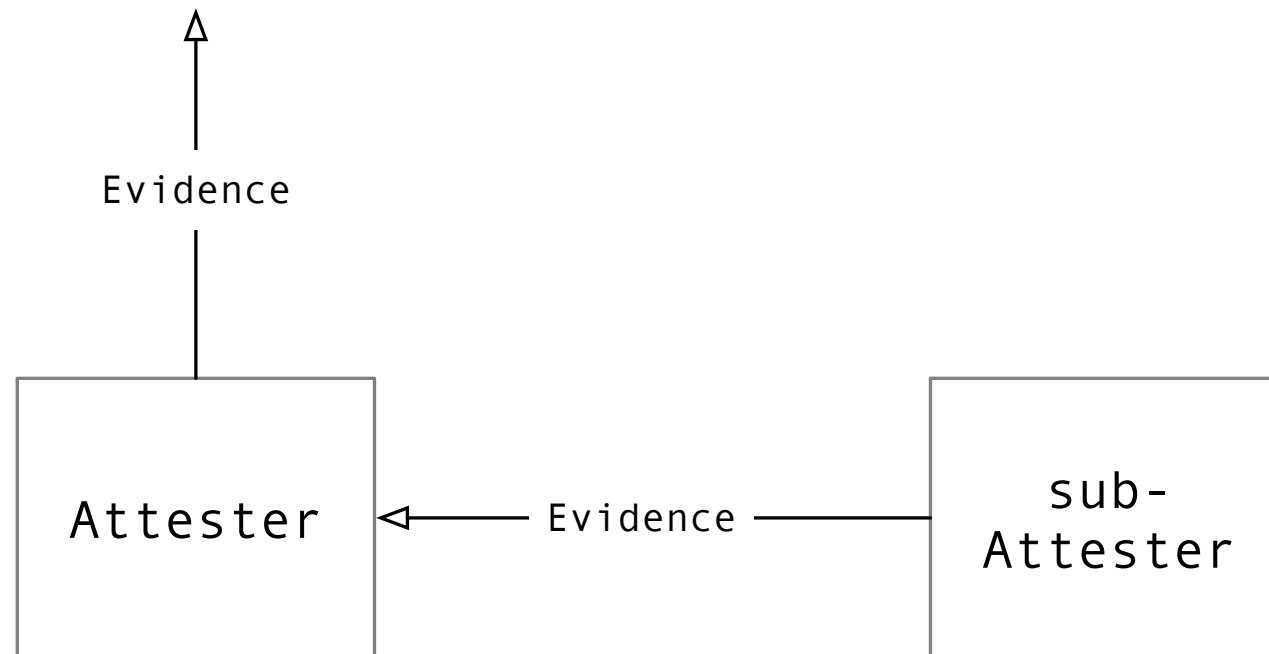
Composition Patterns

- Attester (de)composition
 - i.e., relationships between Attesting and Target Environments within an Attester



Composition Patterns

- Device layering
 - i.e., how different Attesters come together in a composite device



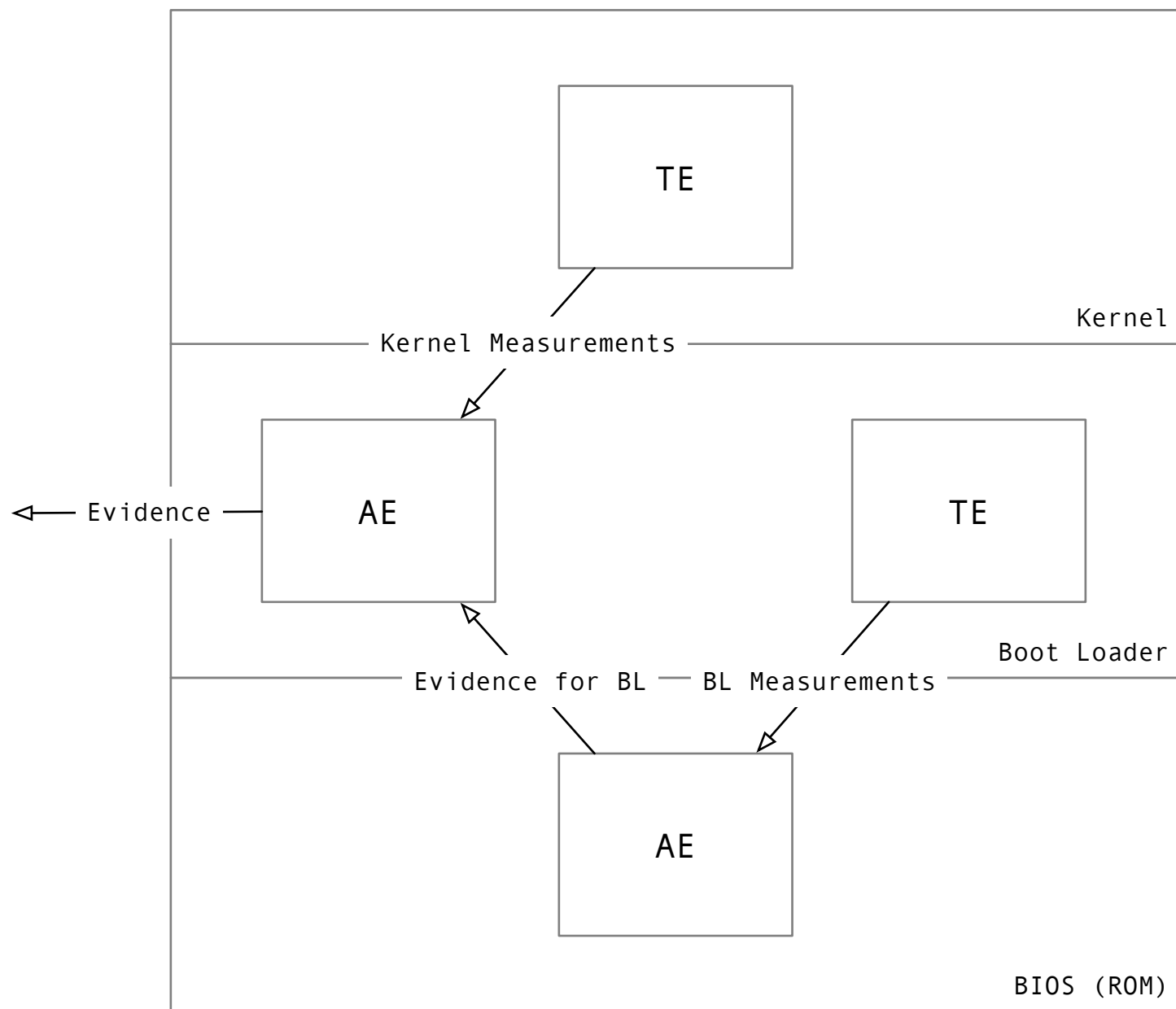
Composition Patterns

It turns out that both can be expressed with the same statement:

Attesting Environment {class-id} retrieves {"claims"/"evidence"} by {"active"/"passive"} collection over {"trusted"/"untrusted"} path from Environment {class-id}

where the “object” Environment could be either a Target Environment or another Attesting Environment in a sub-Attester.

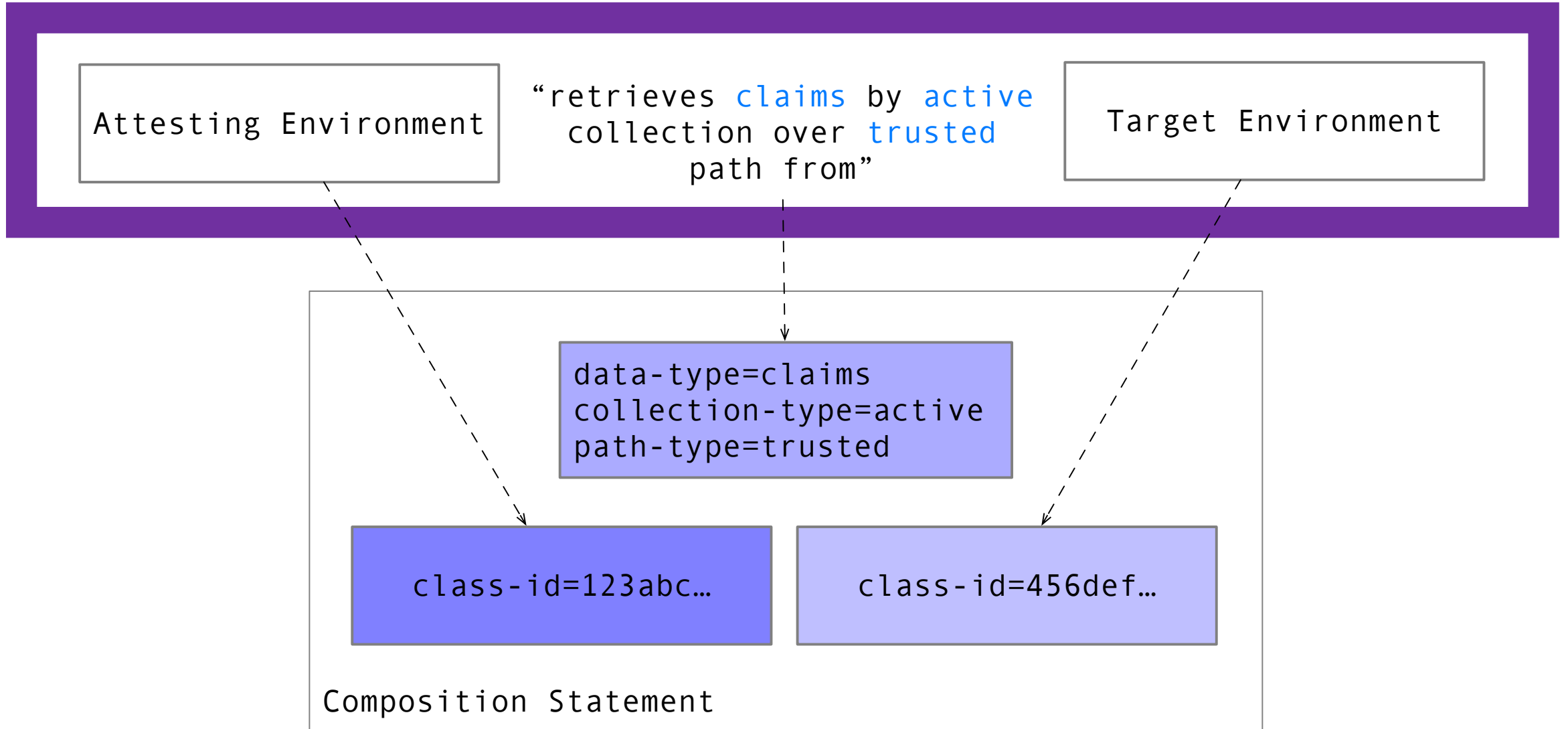
Note: There is also a separate statement to describe the environments that compose a certain Attester. (This is effectively just a grouping overlay on top of a device decomposition that can be fully described by the statement above.)



RATS-ARCH, Layered Attester

- BIOS retrieves claims by active collection over trusted path from Boot Loader
- Boot Loader retrieves evidence by active collection over trusted path from BIOS
- Boot Loader retrieves claims by active collection over trusted path from Kernel

Composition Statement

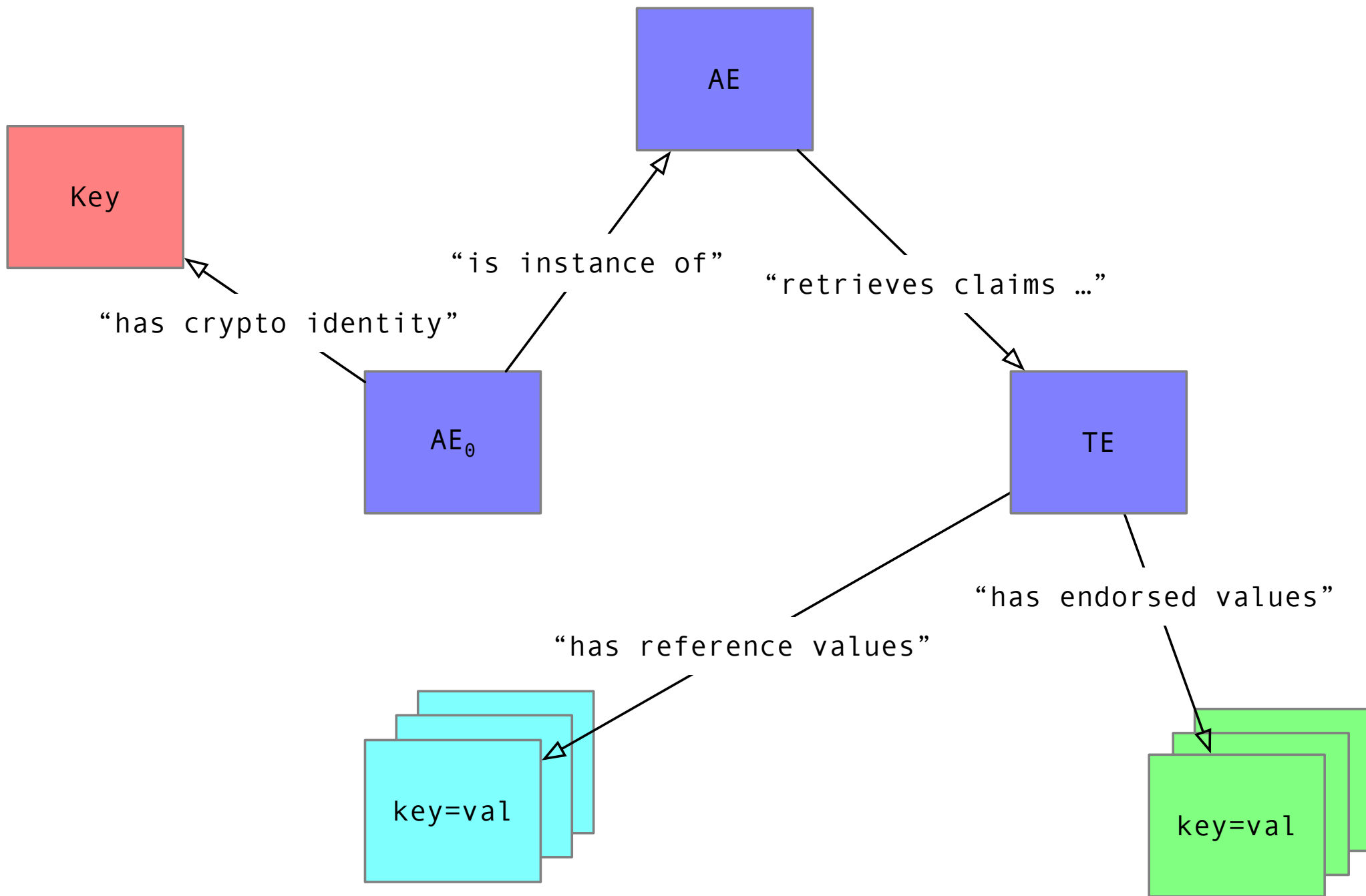


And a few more...

- Attester's PK has certification path *x5chain*
- A and B are aliases for Attester
- Attester is a member of Group
- *<insert your statement here, the format is extensible>*

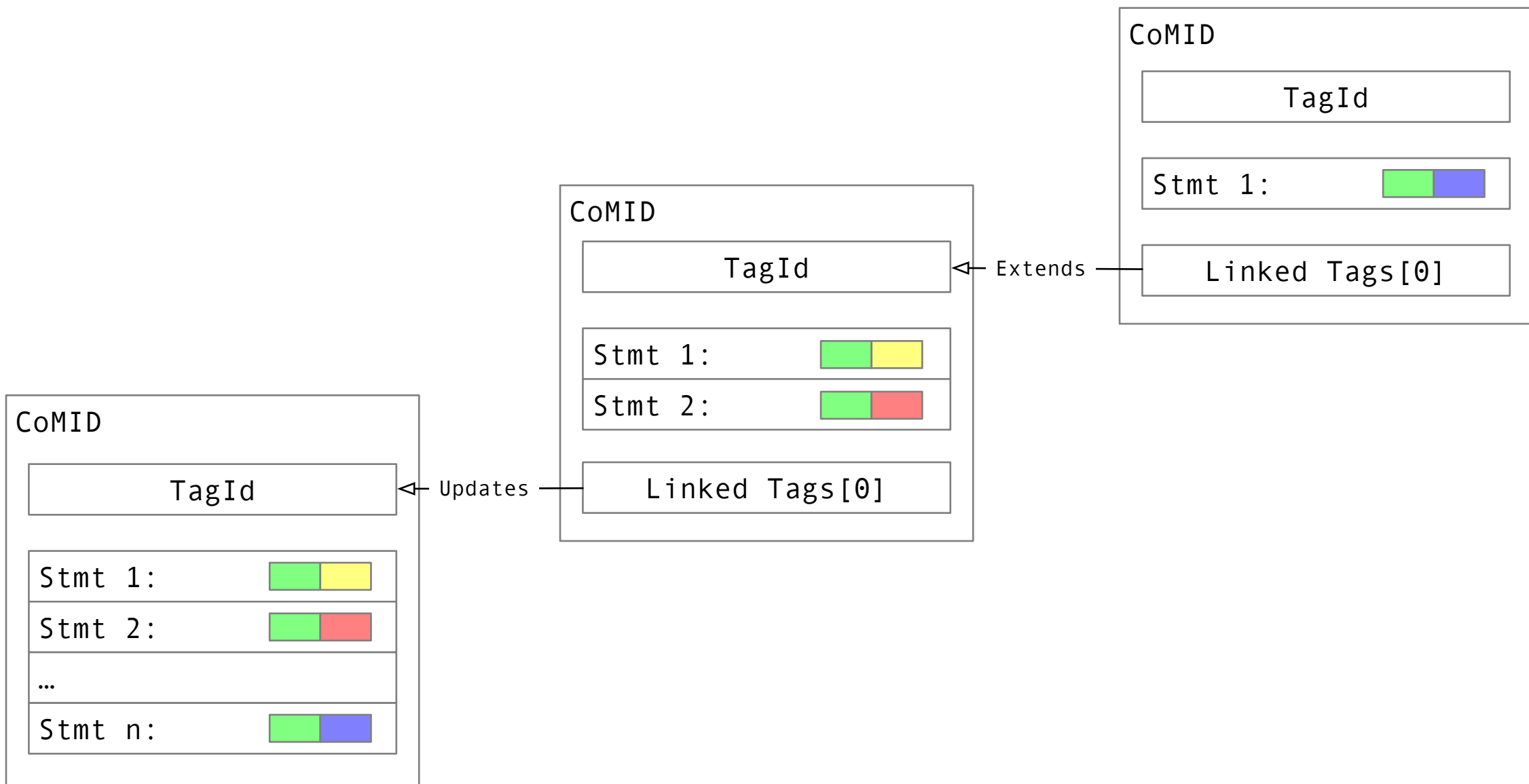
Pulling all together

Navigating the sea of triples allows a Verifier to construct a comprehensive device/attester description that it can use as the backdrop against which its Appraisal Policy for Evidence is evaluated



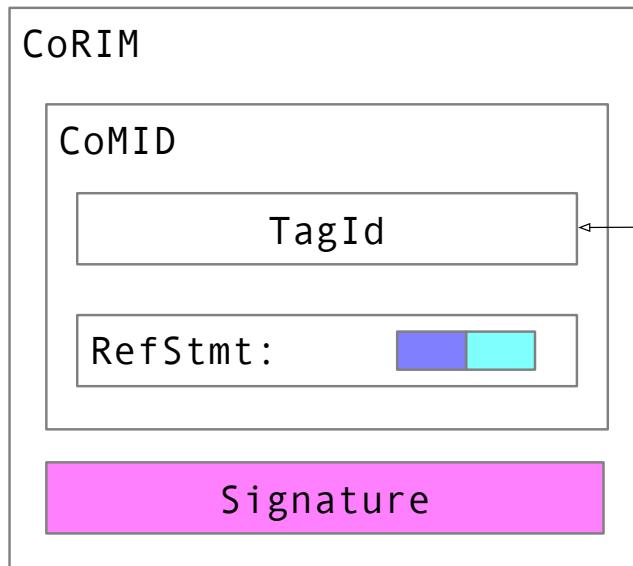
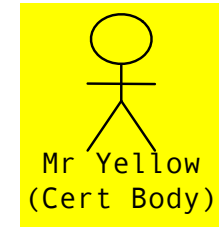
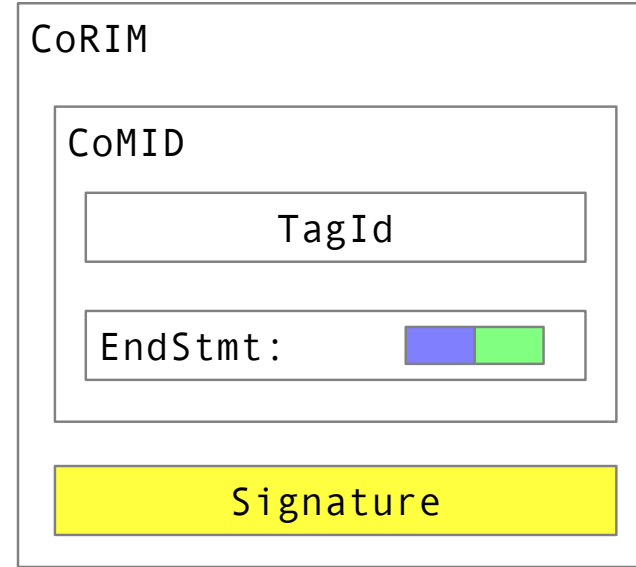
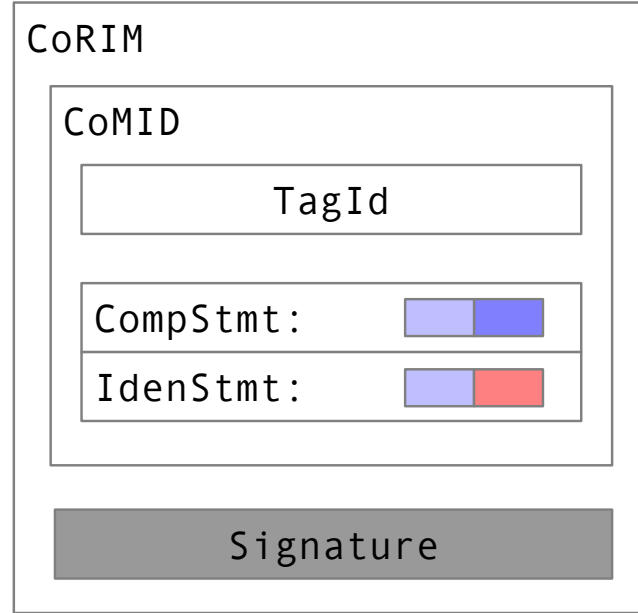
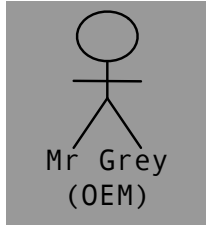
Grouping statements

- CoMID (constrained Module Identifier) is the wrapper around a bunch of statements
- It allows grouping, identification, typed linking (e.g., *supersedes*, *updates*) with other CoMIDs, plus some further encoding optimization (e.g., if the statements subject is always the same it can be factored out)
- Grouping criteria are use-case specific. We can *suggest* a few (e.g., for handling FW updates), but we expect best practices to emerge with time and use

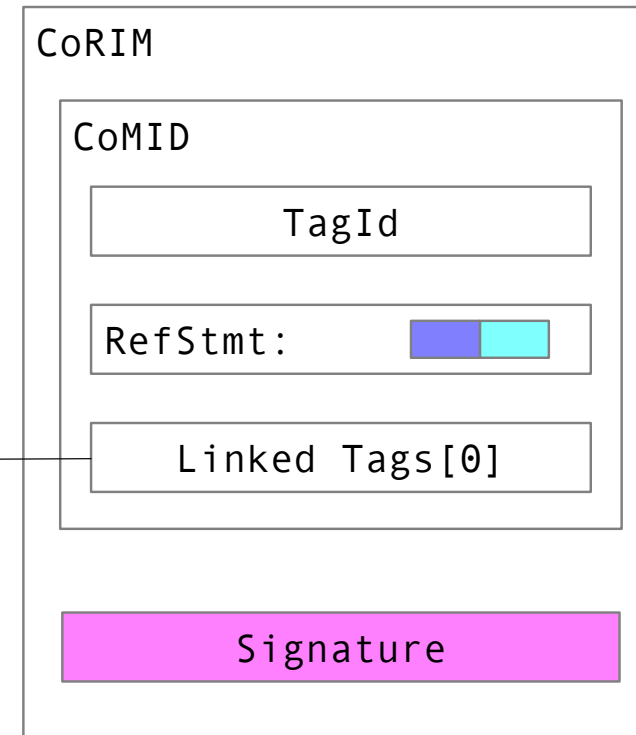


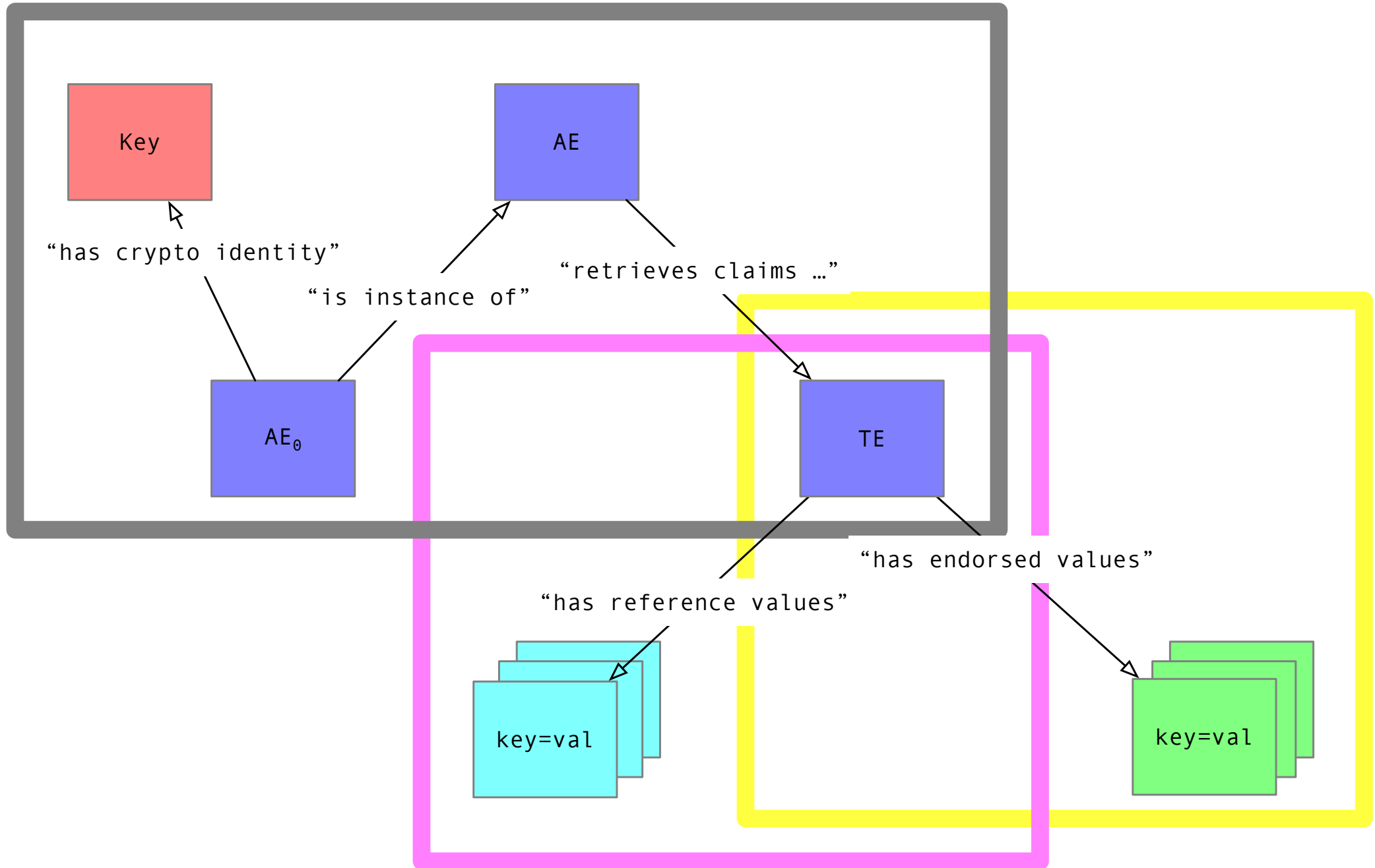
Grouping groups of statements

- CoMIDs (and CoSWIDs) are grouped into CoRIMs
- CoRIMs are signed by the relevant supply chain actor
- Used as the end-to-end conveyance payload (we don't define the transport)
- The outer signature augments the triples in the CoMID statements with provenance:
 - “Supply chain actor X says $\{CoMID\}$ ”
- Idealized supply chain (see next)



← supersedes





Final words

- Currently developed in TCG (DICE WG)
 - work-in-progress
 - Keep an eye on <https://github.com/ietf-rats/ietf-corim-cddl>
- Personnel:
 - Ned Smith (Intel)
 - Yogesh Deshpande and Thomas Fossati (Arm)
 - Henk Birkholz (Fraunhofer SIT)