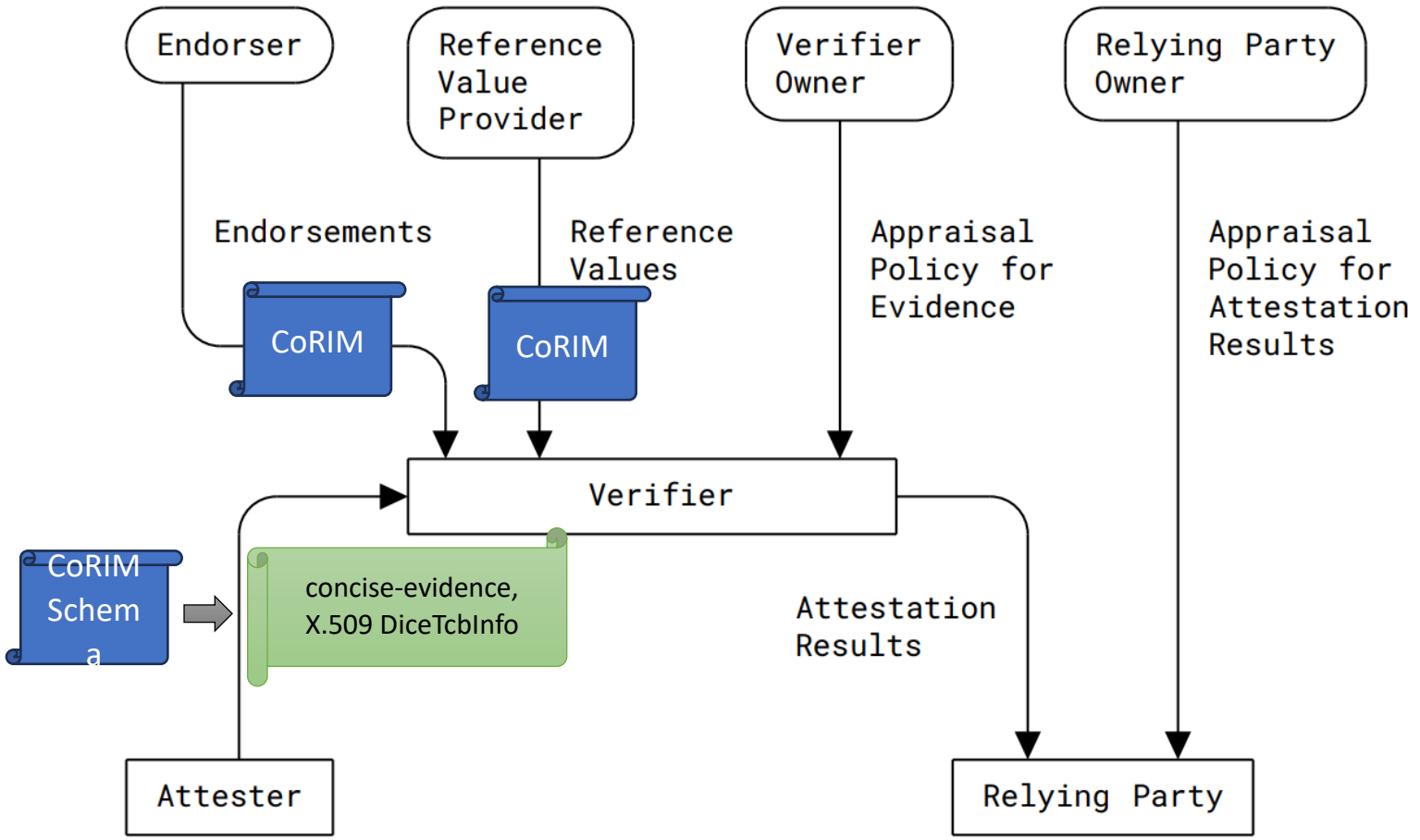
Concise Attestation Results using CoRIM Schema (CAR)

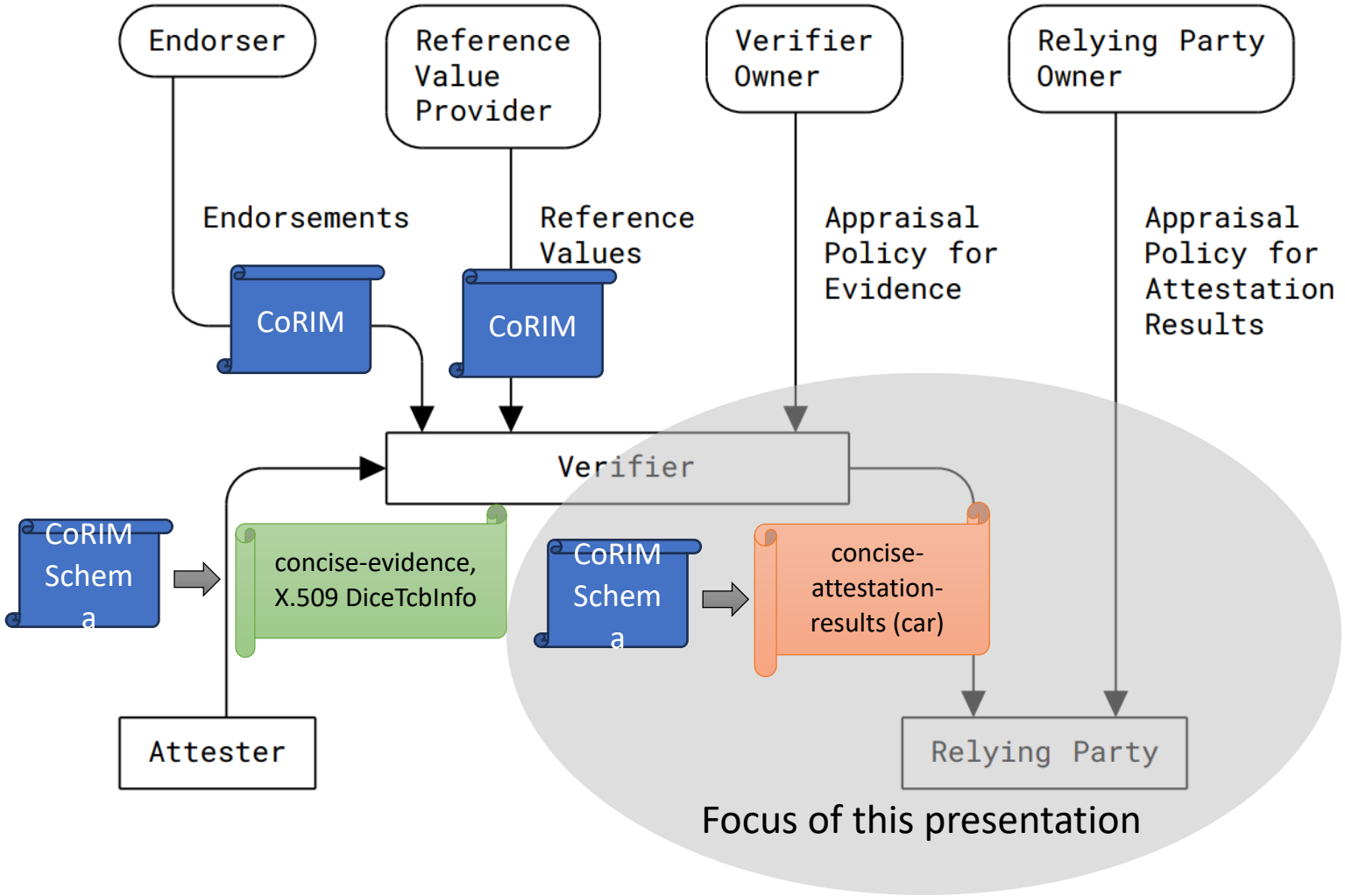
Ned Smith

Intel

RATS Architecture with CoRIM

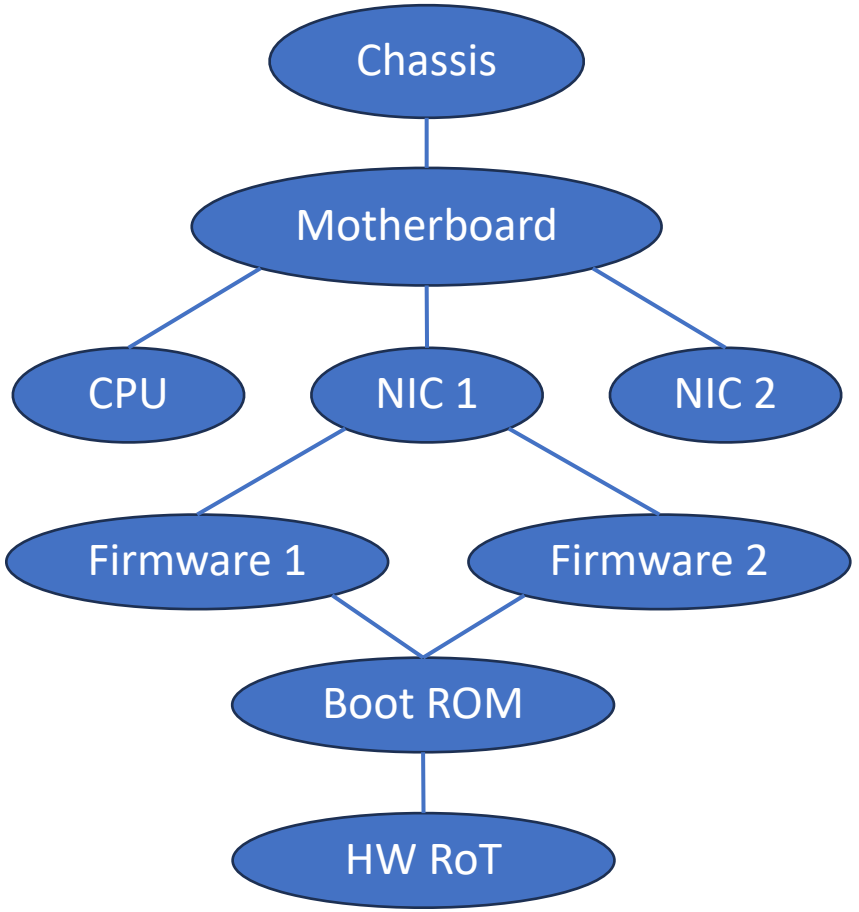


RATS Architecture with CoRIM for Attestation Results



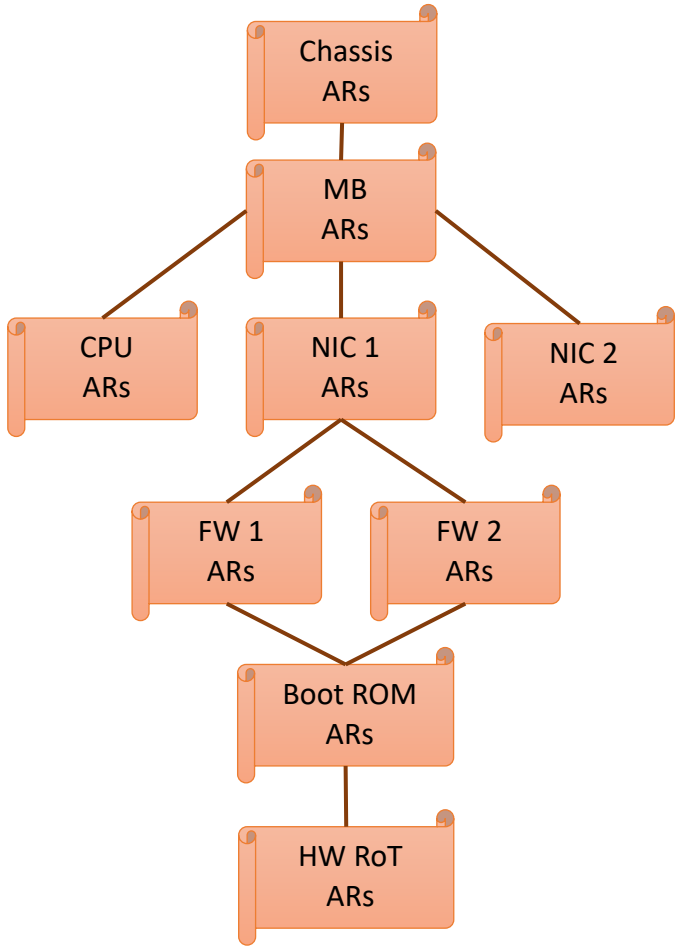
Attestation Results may benefit from Attester compositional context

Example Attester Composition



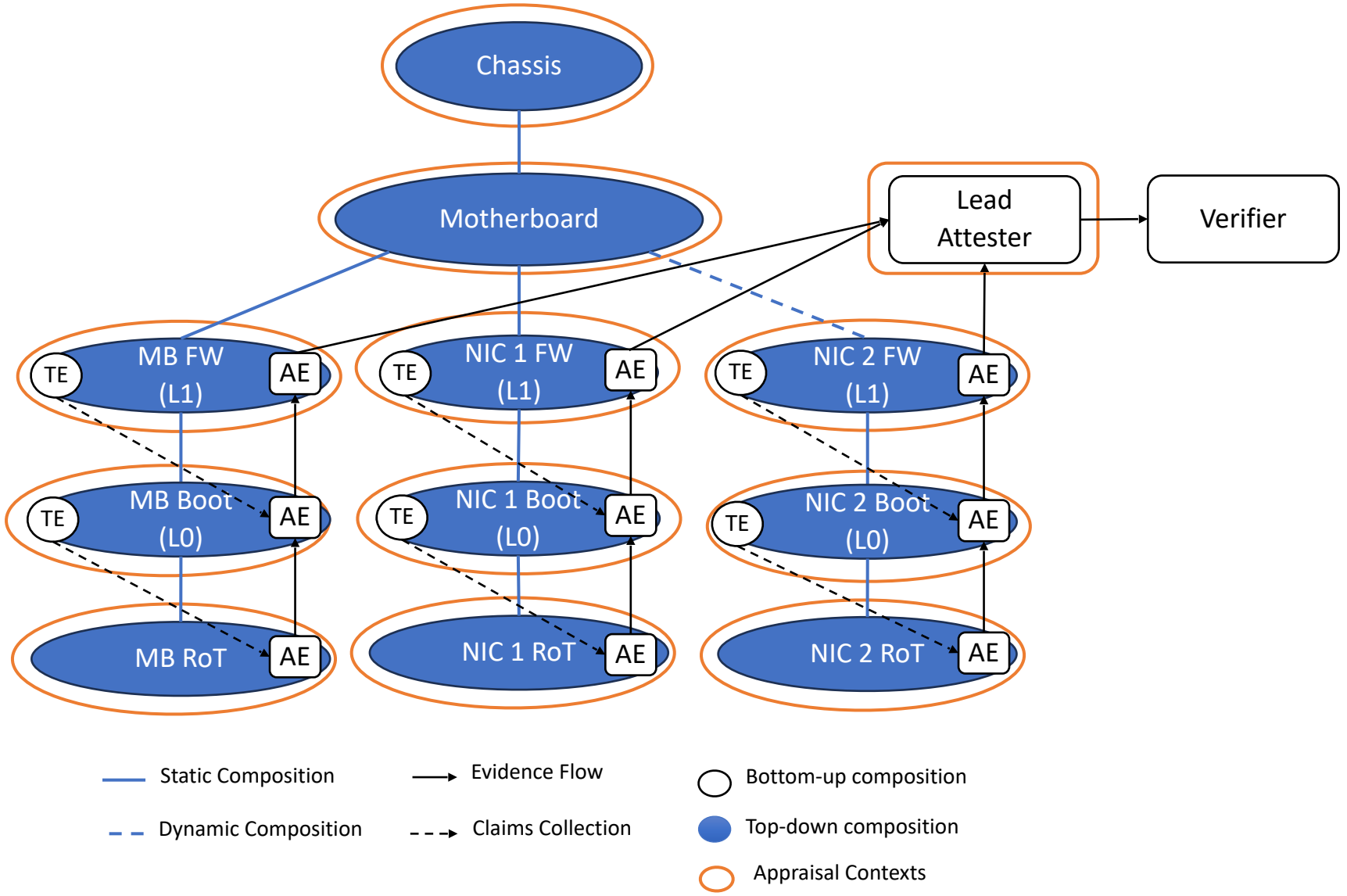
Etc...

Attestation Results follow Attester Composition



Etc...

Example System with Appraisal Contexts



Appraisal Assumptions

- Attestation Results describe the current (actual) state of the Attester.
- RP Owner and RP agree on which Attestation Results are required.
- RP negotiates with Verifier on which Attestation Results are relevant and in which format.
- Attestation Results may be comprised of:
  - AR summary result (e.g., 1-bit verified/not-verified) or multi-bit (e.g., AR4SI). Verifier asserts claims based on its appraisal state.
  - The Verifier may represent Attester's actual state; a.k.a., as contained in ACCEPTED-CLAIMS set (with appropriate redaction to satisfy Attester's privacy policy)

Concise Attestation Results (CAR) Design

- Define a new CoRIM tag type for Attestation Results
  - `tagged-concise-ar-tag = #6.TBA(bytes .cbor concise-ar-tag)`
  - `$concise-tag-type-choice /= tagged-concise-ar-tag`
- Copy relevant Verifier ACCEPTED-CLAIMS into the AR tag
- Verifier originates AR Claims
  - The Verifier may assert new claims about the Attester (or any of its subcomponents)
    - E.g.: AR4SI
    - AR4SI claims summarize the actual state of the Attester.
- Assumptions about the ACCEPTED-CLAIMS set
  - The Attester device composition is represented by ACCEPTED-CLAIMS
  - Only the *current state* of the Attester is represented
  - Inputs to the Verifier that realized ACCEPTED-CLAIMS are available for logging / audit purposes and may be conveyed using a CAR tag.

Explanation of CAR Schema

- Concise-ar-tag
  - **tag-id** – identifies an instance of a CAR tag
  - **profile** – a profile identifier that qualifies ‘normal’ behavior
  - **ar-triples** – the triples that describe the current state of the Attester once appraisal is complete. ar-triples are a subset of CoRIM triples. The exact subset is TBD. Since the Verifier’s accepted claims describes the actual state of the Attester, `reference-triples` is omitted.

```
concise-ar-tag = {
      &(tag-id: 0) => tag-identity-map
    ? &(profile: 1) => $profile-type-choice
      &(ar-triples: 2) => ar-triples-map
    * $$concise-ar-tag-extension
}
```

Note: The AR4SI I-D suggests the trustworthiness of the Verifier is important to the Relying Party and that this is achieved through evaluation of the Verifier’s software. The `concise-evidence` schema can be used to satisfy this requirement. The verifier can bundle Evidence and Attestation Results in a common Conceptual Message.

Explanation of CAR Schema – ar-triples-map

- CoMID triples as Attestation Results (a.k.a., `ar-triples-map`)
  - **endorsed-triples** – The Verifier's ACCEPTED-CLAIMS set. Contains Attester current state.
  - **dependency-triples** – ACCEPTED-CLAIMS set may have trust dependencies that have been verified.
  - **membership-triples** – ACCEPTED-CLAIMS set may describe Attester composition. Membership captures the composition hierarchy.
- On-the-fence triples
  - **coswid-triples** – Software packages that are installed on a target environment are described using coswid.

Explanation of CAR Schema – AR Claims

- The Verifier may assert claims about the Attester.
  - The CoMID `measurement-values-map` is extended to include AR4SI claims.
  - The Verifier creates claims under its own authority.
  - The appraisal log is a claim about appraisal integrity.
  - AR4SI claims can be asserted within the appropriate grouping context.
- AR4SI Claims
  - **timestamp** – the time the Verifier asserted AR4SI claims
  - **status** – the ar4si trust tier (none, affirming, warning, contraindicated).
    - A trust tier status is applied to the top of a composition hierarchy.
    - It aggregates status derived from the status of each of its sub-components.
  - **trust-vector** – the ar4si trust vector - a codepoint from -128 to 127 describing a results condition
  - **policy-id** – the appraisal policies used by the Verifier
  - **ar-log** – proof of appraisal integrity

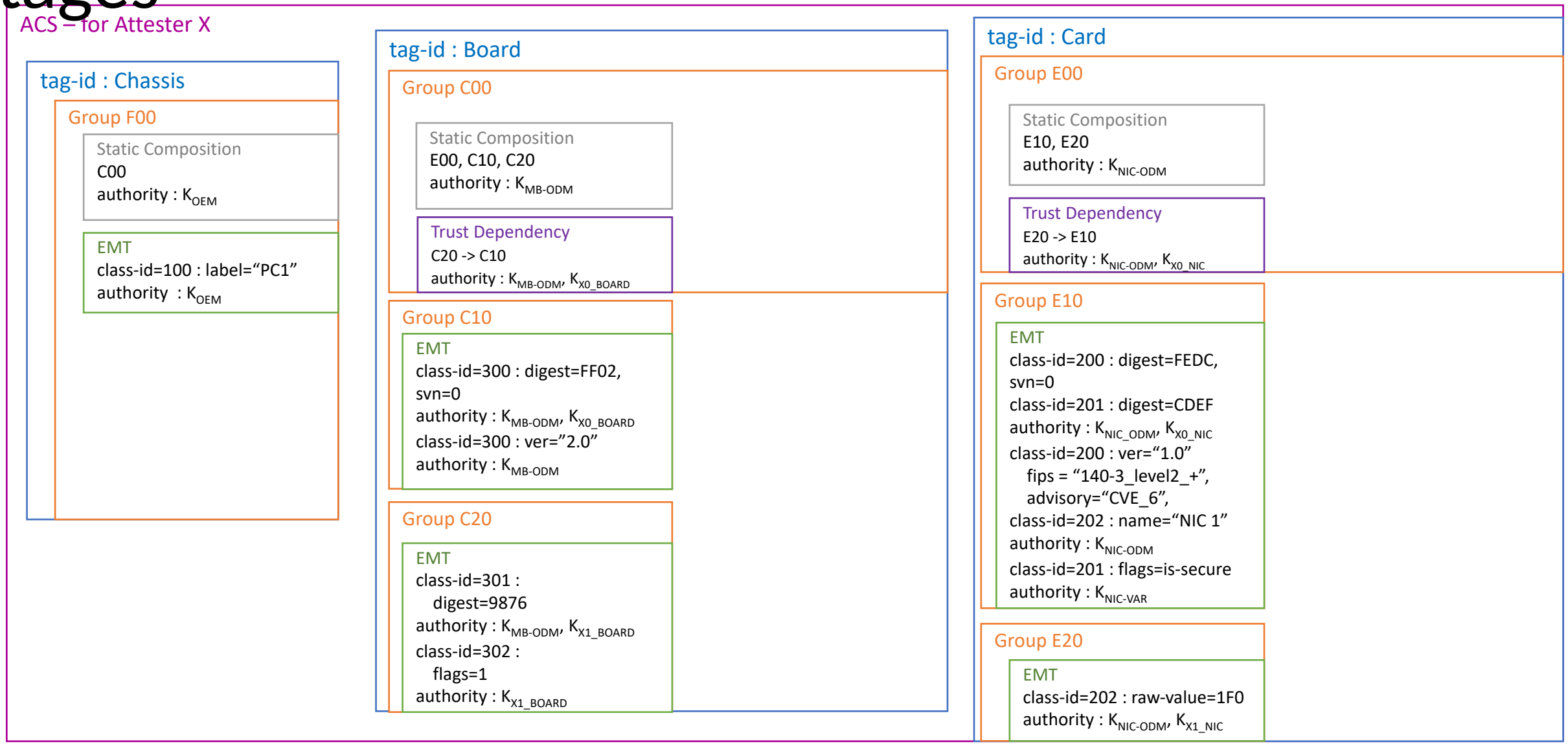
```
$$measurement-values-map-extension //= (
  ? &(timestamp: -1) => ~time
  ? &(status: -2) => $ar4si.trust-tier
  ? &(vector: -3) => ar4si.trustworthiness-vector
  ? &(policy-id: -4) => text ; format TBD
  ? &(ar-log: -5) => bytes ; format TBD
)
```

Note: The complete CDDL is in Backup

Verifier Stages

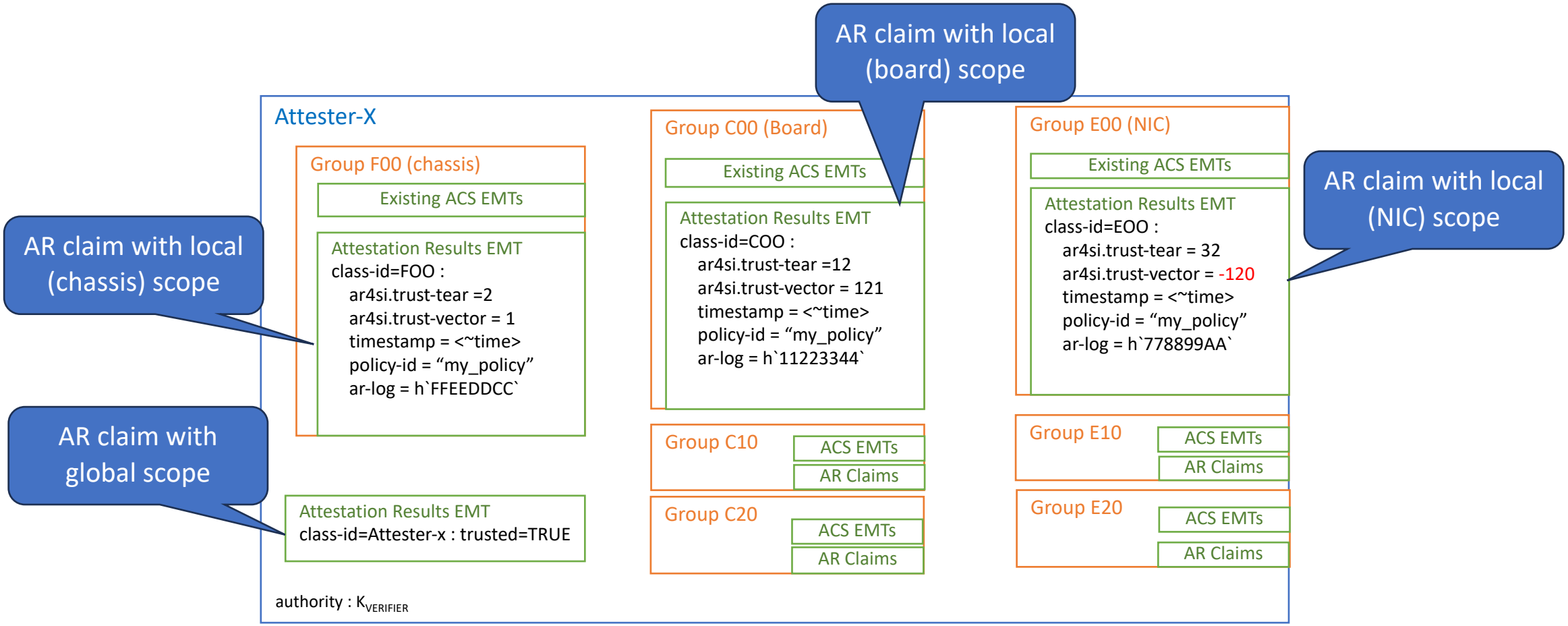
- **Stage 1** : Add Evidence to ACS with Attester authority
- **Stage 2**: Match Reference Values with ACS and add RVP authority
- **Stage 3**: Add direct Endorsements to ACS and add Endorsed Values under Endorser authority
- **Stage 4**: Match Conditional Endorsements with ACS and add Endorsed Values under Endorser authority
- **Stage 5**: Apply appraisal policies to ACS by trimming claims without proper authority
- **Stage 6**: Assess ACS and add Verifier asserted claims under Verifier authority

Accepted Claims Set (ACS) After Verifier Stages



Verifier Asserted Claims

- Verifier asserted claims are added to the ACS



Accepted Claims Set (ACS) Final State

ACS – for Attester X

tag-id : Chassis

Group F00

Static Composition
C00
authority : K_{OEM}

EMT
class-id=100 : label="PC1"
authority : K_{OEM}

EMT
class-id=FOO :
ar4si.trust-tear =2
ar4si.trust-vector = 1
timestamp = <~time>
policy-id = "my_policy"
ar-log = h`FFEEDDCC`
authority : $K_{VERIFIER}$

EMT
class-id=attester-x : trusted=TRUE
authority : $K_{VERIFIER}$

tag-id : Board

Group C00

Static Composition
E00, C10, C20
authority : K_{MB-ODM}

Trust Dependency
C20 -> C10
authority : K_{MB-ODM}, K_{X0_BOARD}

EMT
class-id=COO :
ar4si.trust-tear =12
ar4si.trust-vector = 121
timestamp = <~time>
policy-id = "my_policy"
ar-log = h`11223344`
authority : $K_{VERIFIER}$

Group C10

EMT
class-id=300 : digest=FF02,
svn=0
authority : K_{MB-ODM}, K_{X0_BOARD}
class-id=300 : ver="2.0"
authority : K_{MB-ODM}

Group C20

EMT
class-id=301 :
digest=9876
authority : K_{MB-ODM}, K_{X1_BOARD}
class-id=302 :
flags=1
authority : K_{X1_BOARD}

tag-id : Card

Group E00

Static Composition
E10, E20
authority : $K_{NIC-ODM}$

Trust Dependency
E20 -> E10
authority : $K_{NIC-ODM}$, K_{X0_NIC}

EMT
class-id=EOO :
ar4si.trust-tear = 32
ar4si.trust-vector = -120
timestamp = <~time>
policy-id = "my_policy"
ar-log = h`778899AA`
authority : $K_{VERIFIER}$

Group E10

EMT
class-id=200 : digest=FEDC,
svn=0
class-id=201 : digest=CDEF
authority : K_{NIC_ODM}, K_{X0_NIC}
class-id=200 : ver="1.0"
fips = "140-3_level2_+",
advisory="CVE_6",
class-id=202 : name="NIC 1"
authority : $K_{NIC-ODM}$
class-id=201 : flags=is-secure
authority : $K_{NIC-VAR}$

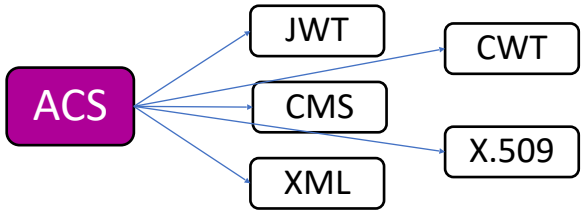
Group E20

EMT
class-id=202 : raw-value=1F0
authority : $K_{NIC-ODM}$, K_{X1_NIC}

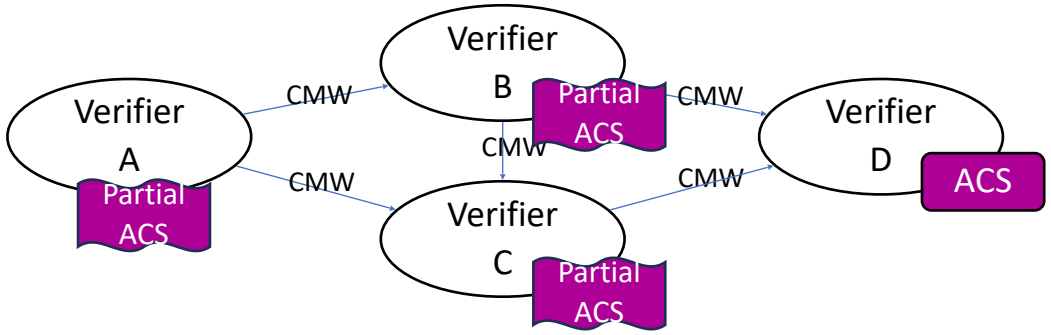
Note: Verifier may construct an Attestation Results message from the ACS

Why Add Verifier Claims to the ACS?

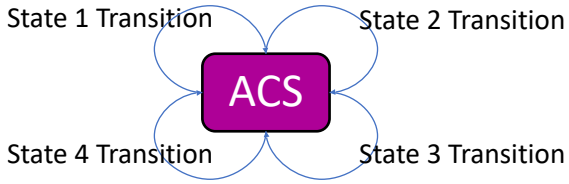
Common oracle for multiple AR formats



Leverage for Verifier MESH deployments



Processing Correctness / Consistency



Comparison to EAT Attestation Results (EAR)

EAR

1.Must be an EAT token

2. Uses EAT profiles

3. ‘Issued at’ freshness claim

4.Verifier identity / trust def’n is new

5.Leverages AR4SI

6.Raw evidence / audit trail is new

7.EAT submods used as outer container class

8.AR4SI trust tier, vector, and policy have EAT submod scope

9.Uses existing EAT `nonce` claim

CAR

1.Uses CoRIM container; can map to any other format (CWT/JWT/EAT)

2.Uses CoRIM schema

3.Freshness extension for CoMID `measurement-values-map`

4.Follows existing CoRIM authority model

5.Leverages AR4SI

6.Raw evidence / audit trail already in ACS

7.CoRIM `tag` as outer container class

8.AR4SI trust tier, vector, and policy have flexible scope

9.Uses existing CoRIM / COSE recentness mechanisms

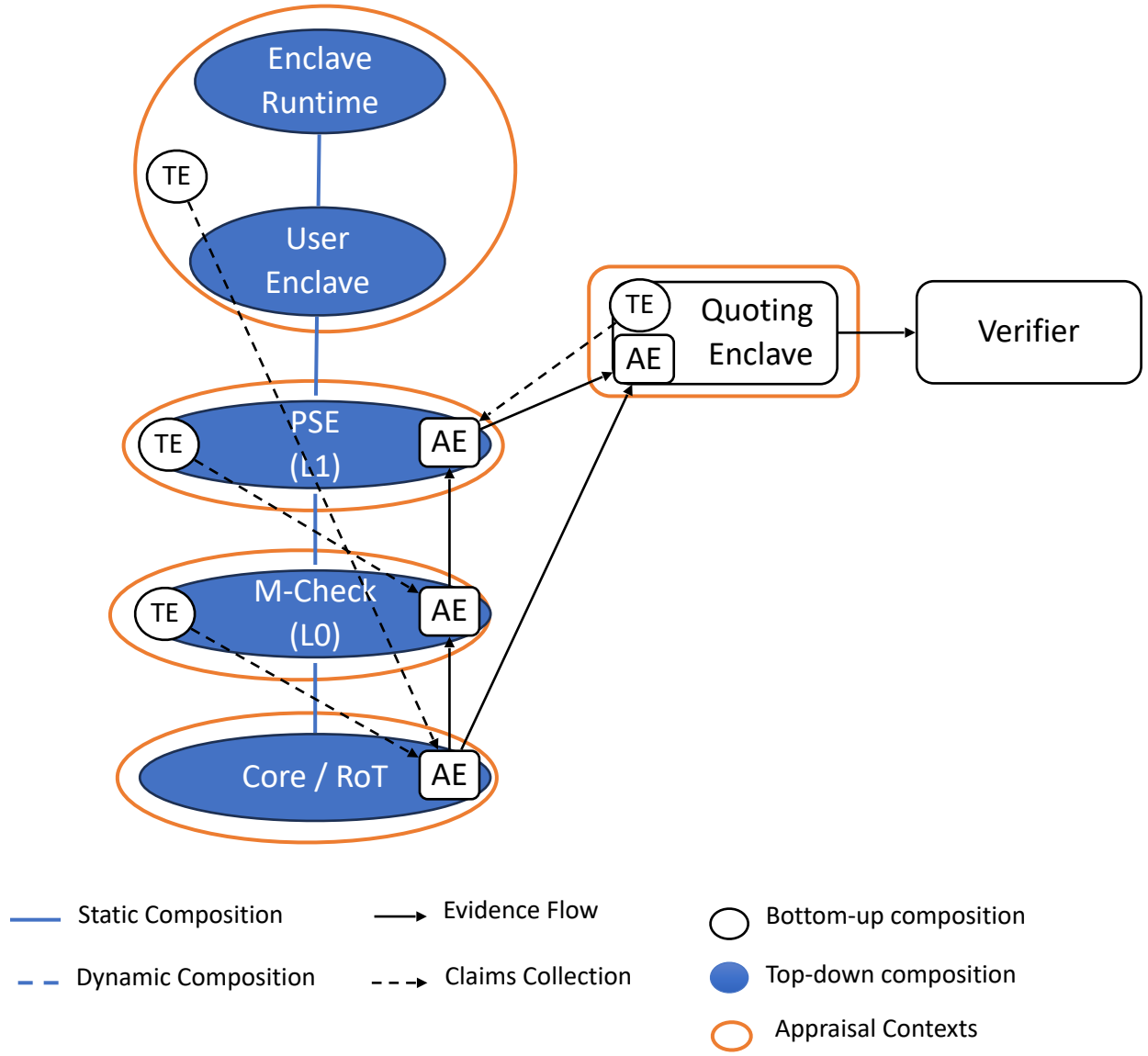
10. Supports advanced use models with ease

Conclusion

- EAR and CAR are nearly the same, except that it doesn't allow reuse of CoMID schema
- What needs fixing?
  - EAR claims as `measurement-values-map` extensions
  - Representation of ACS using CoMID triples
    - e.g., endorsed-triple-record
  - Attestation Results as a CoRIM tag type

Backup

Example SGX System with Appraisal Contexts



EAR Schema in CDDL

```
EAR = {
  eat.profile => "tag:github.com,2023:veraison/ear"
  iat => int
  ear.verifier-id => ar4si.verifier-id
  ? ear.raw-evidence => ear-bytes
  eat.submods => { + text => EAR-appraisal }
  ? eat.nonce => eat.nonce-type
  * $$ear-extension
}
```

```
EAR-appraisal = {
  ear.status => $ar4si.trust-tier
  ? ear.trustworthiness-vector => ar4si.trustworthiness-vector
  ? ear.appraisal-policy-id => text
  * $$ear-appraisal-extension
}
```

CAR Schema in CDDL

```
tagged-concise-ar-tag = #6.5XXTBD(bytes .cbor concise-ar-tag)

$concise-tag-type-choice /= tagged-concise-ar-tag

concise-ar-tag = {
       &(tag-id: 0) => tag-identity-map
     ? &(profile: 1) => $profile-type-choice
       &(ar-triples: 2) => ar-triples-map
     * $$concise-ar-tag-extension
}

$$measurement-values-map-extension //= (
  ? &(timestamp: -1) => ~time
  ? &(status: -2) => $ar4si.trust-tier
  ? &(vector: -3) => ar4si.trustworthiness-vector
  ? &(policy-id: -4) => text
  ? &(ar-log: -5) => bytes ; format TBD
)
```

```
ar-triples-map = non-empty<{
  ? &(endorsed-triples: 0) =>
    [ + endorsed-triple-record ]
  ? &(dependency-triples: 1) =>
    [ + domain-dependency-triple-record ]
  ? &(membership-triples: 2) =>
    [ + domain-membership-triple-record ]
  ? &(identity-triples: 3) =>
    [ + identity-triple-record ]; tbd
  ? &(attest-key-triples: 4) =>
    [ + attest-key-triple-record ]; tbd
  ? &(coswid-triples: 5) =>
    [ + coswid-triple-record ]; tbd
  * $$ar-triples-map-extension
}>
```

Back

Common AR4SI Schema

```
$ar4si.trust-tier /= ar4si.trust-tier-none
$ar4si.trust-tier /= ar4si.trust-tier-affirming
$ar4si.trust-tier /= ar4si.trust-tier-warning
$ar4si.trust-tier /= ar4si.trust-tier-contraindicated
ar4si.trust-tier-none = 0
ar4si.trust-tier-affirming = 2
ar4si.trust-tier-warning = 32
ar4si.trust-tier-contraindicated = 96

ar4si.trustworthiness-vector = non-empty<{
    ? instance-identity => $ar4si.trustworthiness-claim
    ? configuration => $ar4si.trustworthiness-claim
    ? executables => $ar4si.trustworthiness-claim
    ? file-system => $ar4si.trustworthiness-claim
    ? hardware => $ar4si.trustworthiness-claim
    ? runtime-opaque => $ar4si.trustworthiness-claim
    ? storage-opaque => $ar4si.trustworthiness-claim
    ? sourced-data => $ar4si.trustworthiness-claim
}>
$ar4si.trustworthiness-claim = -128..127
```

Back

Conditional Endorsement Triple

```
conditional-endorsement-triple-record = [
  stateful-environment-record,
  ; endorsed values
  measurement-values-map
]
stateful-environment-record = [
  environment-map,
  measurement-map
]
```

Conditional Endorsement Series Triple

```
conditional-endorsement-series-triple-record = [
  stateful-environment-record
  ; order matters: the first matching record wins and halts matching
  [ + conditional-series-record ]
]
stateful-environment-record = [
  environment-map,
  measurement-map
]
conditional-series-record = [
  ; reference values to be matched against evidence
  refv: measurement-values-map
  ; endorsed values that apply in case revf matches
  endv: measurement-values-map
]
```

Status of Attestation Results Drafts in IETF

- I-D.ietf-rats-ar4si – defines Attestation Results status and various claims that aggregate Accepted Claims
  - Note: I-D. ietf-rats-ar4si doesn’t contain CDDL definitions
- I-D.fv-rats-ear – defines an Attestation Results conceptual message using ar4si and EAT (Entity Attestation Token)
  - Note: Not yet adopted by a WG