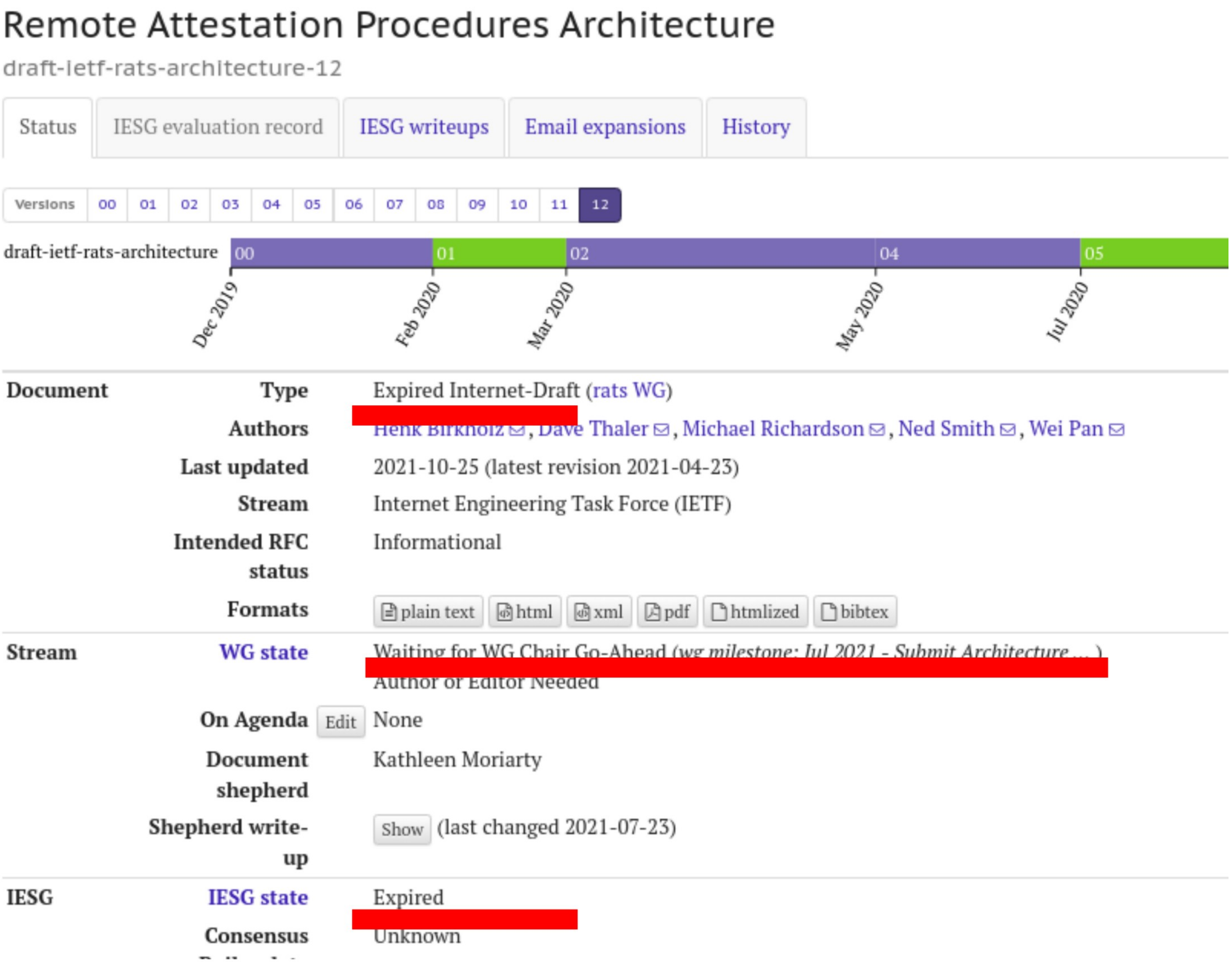
RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

12:05 : 12:15 **RATS Architecture and next steps**
(10 min) Michael Richardson (draft-ietf-rats-architecture-12)

RATS Architecture Status: no change since April



Remote Attestation Procedures Architecture

draft-ietf-rats-architecture-12

Document	**Type**	Expired Internet-Draft (rats WG)
	Authors	Henk Birkholz, Dave Thaler, Michael Richardson, Ned Smith, Wei Pan
	Last updated	2021-10-25 (latest revision 2021-04-23)
	Stream	Internet Engineering Task Force (IETF)
	Intended RFC status	Informational
	Formats	plain text html xml pdf htmlized bibtex
Stream	**WG state**	Waiting for WG Chair Go-Ahead *(wg milestone: Jul 2021 - Submit Architecture ...)* Author or Editor Needed
	On Agenda	None
	Document shepherd	Kathleen Moriarty
	Shepherd write-up	(last changed 2021-07-23)
IESG	**IESG state**	Expired
	Consensus	Unknown

Questions
Discussion

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

12:15 : 12:20 **Attestation Event Stream Subscription**
(5 min) Eric Voit (draft-ietf-rats-network-device-subscription-00)

Attestation Event Stream Subscription

draft-ietf-rats-network-device-subscription-00

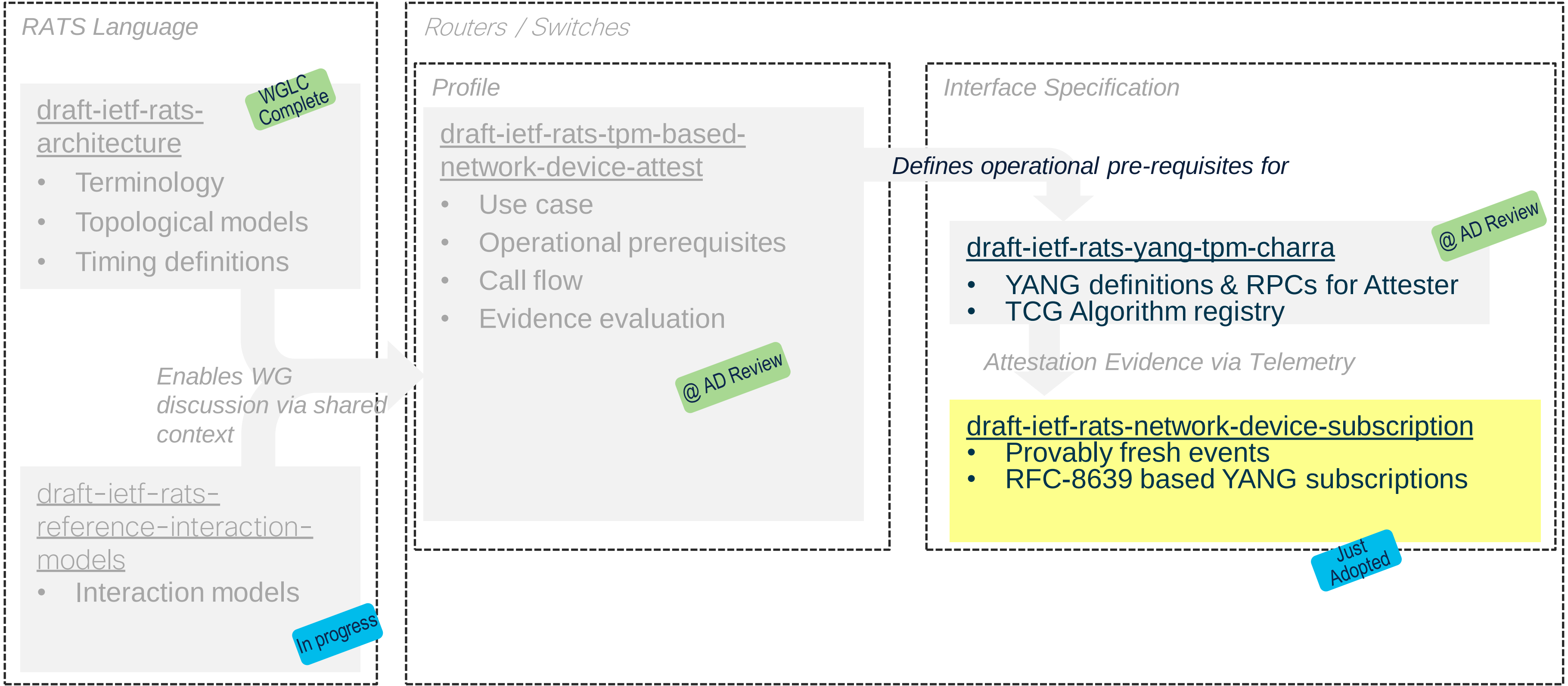
Henk Birkholz {henk.birkholz@sit.fraunhofer.de},

Eric Voit {evoit@cisco.com},

Wei Pan {william.panwei@huawei.com}

November 2021, RATS WG

Relationship to other RATS drafts



RATS Language

WGLC Complete

draft-ietf-rats-architecture
- Terminology
- Topological models
- Timing definitions

Enables WG discussion via shared context

draft-ietf-rats-reference-interaction-models
- Interaction models

In progress

Routers / Switches

Profile

draft-ietf-rats-tpm-based-network-device-attest
- Use case
- Operational prerequisites
- Call flow
- Evidence evaluation

@ AD Review

Defines operational pre-requisites for

Interface Specification

draft-ietf-rats-yang-tpm-charra
- YANG definitions & RPCs for Attester
- TCG Algorithm registry

@ AD Review

Attestation Evidence via Telemetry

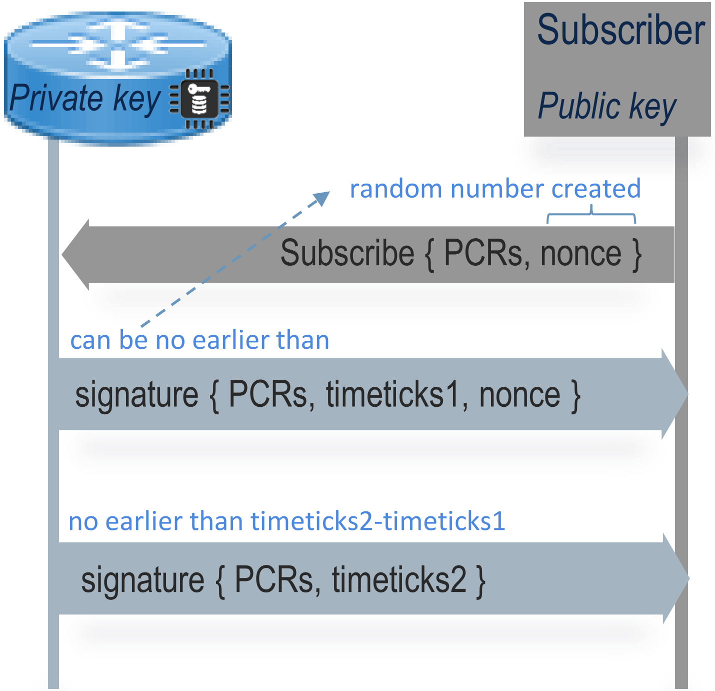
draft-ietf-rats-network-device-subscription
- Provably fresh events
- RFC-8639 based YANG subscriptions

Just Adopted

Purpose & Scope

- Defines how to subscribe to a stream of attestation related Evidence on TPM-based network devices.
  - When subscribed, a Telemetry stream of verifiably fresh YANG notifications are pushed to the subscriber.
  - Notifications are generated for the Evidence going into TPM PCRs, and when the PCRs are extended.

- Result
  - Verifier is pushed new verifiably fresh Evidence whenever PCRs change.


Private key
Subscriber
Public key
random number created
Subscribe { PCRs, nonce }
can be no earlier than
signature { PCRs, timeticks1, nonce }
no earlier than timeticks2-timeticks1
signature { PCRs, timeticks2 }

Contents

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

12:20 : 12:40 **A CBOR Tag for Unprotected CWT Claims Sets**
(10 min) Carsten Bormann (draft-ietf-rats-uccs-01)

draft-ietf-rats-uccs-01

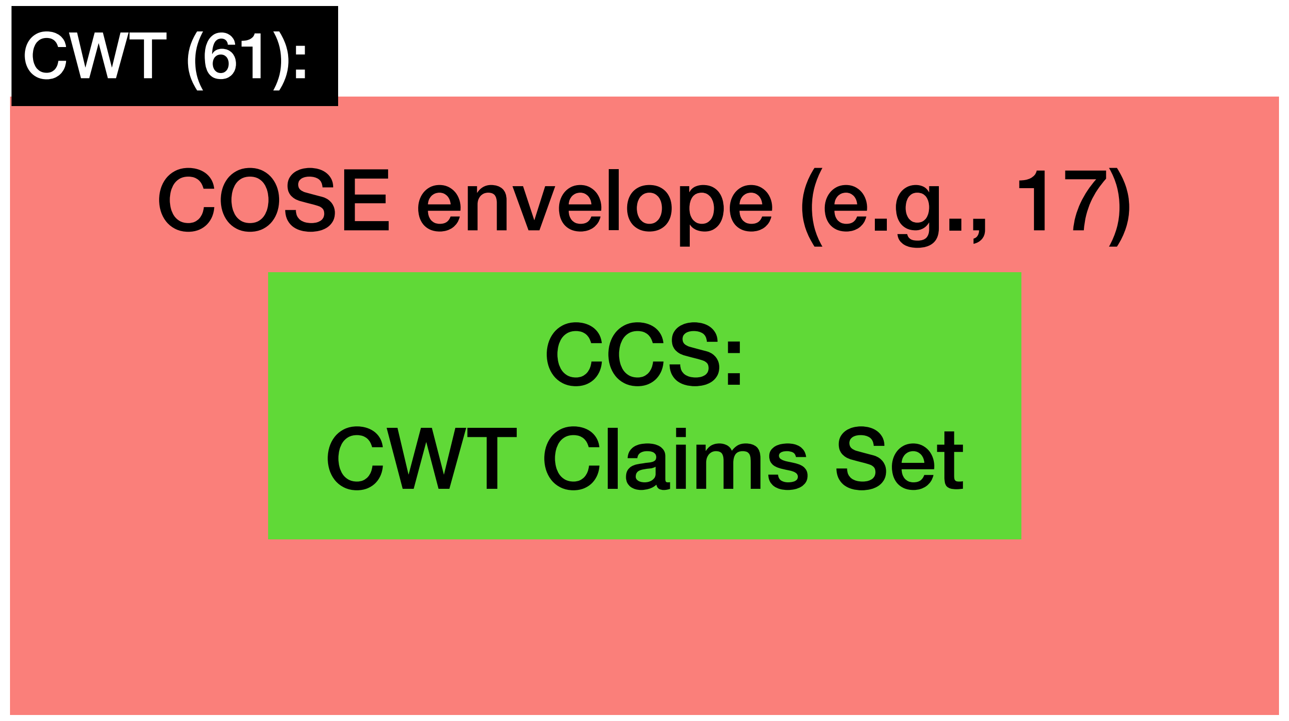
A CBOR Tag for Unprotected CWT Claims Sets

Carsten Bormann, 2021-11-08 • IETF 112

CWT, CCS, and UCCS

- RFC 8392 defines CWT:
  - CWT = COSE armor around CCS (tag 61)
  - CCS is similar to a JWT claims set
    (RFC 7519, RFC 8726):
    - key/value set (map) of “claims”
    - **together** form an assertion
- UCCS = Unprotected CCS (tag 601*)

CWT:

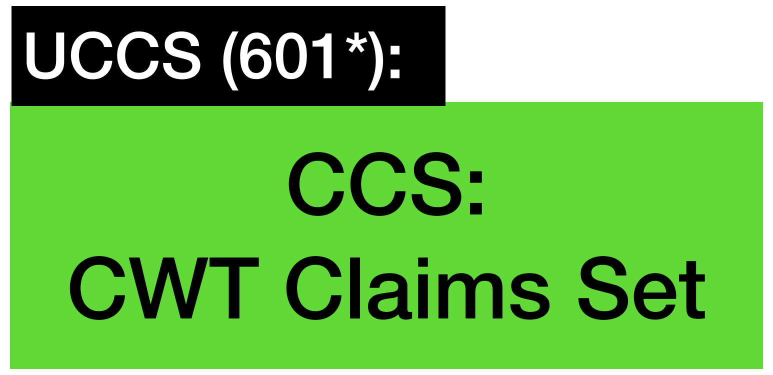


CWT (61):

COSE envelope (e.g., 17)

CCS:
CWT Claims Set

UCCS:



UCCS (601*):

CCS:
CWT Claims Set

*) Tag 601 proposed, but not yet assigned.

Why does UCCS need a specification?

- Actually: no. Could just register the tag and refer to RFC 8392.
- Better: yes.
  - Write up the area of application: UCCS is **not** a replacement for CWT.
    - Security considerations.
    - Relationship to RATS concepts, likely usage in RATS.
      What are the RATS requirements on a secure channel carrying a UCCS?

While we are at it…

- RFC 8392 (CWT) predates completion of RFC 8610 (CDDL).
  Now could provide CDDL spec for CCS.
  (Proposal is in a UCCS repo branch.)

  - (Note that CDDL for COSE is in RFC 8152 [yes, that predates RFC 8610, too] and RFC 9052-to-be.)

- Grander plans for unification between JWT (JCS) and CWT (CCS):
  Probably not.
  And if yes anyway, not here.

Next Steps

- Accept or reject the idea to add CDDL for CCS
- One more round of editing to address more of Thomas Fossati’s review
- WGLC then

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

12:40 : 12:55 **Entity Attestation Token r11 changes**
(15 min) Laurence Lundblade (draft-ietf-rats-eat-11)

EAT Change in -11 draft

Laurence Lundblade

IETF 112 November 2021

Planned Contents of an EAT - The Claims

HW Identification

OEM, model, version…
Unique device identification

SW Identification - CoSWID

Author, package, version…
Measurement

Security Characterization

High-level OS, TEE, secure element, TPM…

Nonce and Timestamps

Freshness, prevent replay

Identify Verifier Input

Endorsements, key ID, reference values…

Context, Purpose, Profile

Intended use cases, profile claim

GPS Location

Public Keys

Attestation of private keys on the device (e.g., Android key store)

Running State

Boot and debug state

Measurement of Running SW

Runtime integrity check

Submodules

HW subsystems, TEE, SW process and apps…

Nested EATs

One signed EAT inside another

Formal Device Certifications

For example, Common Criteria certification; format is GP's DLOA

Verification Results

Overall Verification Results, Measurement Results

Level of Completion in EAT Draft

🟩🟩🟩🟩
- Ready for last call, no open issues
- Near completion, reviewed
- Draft text
- Proposed, Interest in

➡ Progress & change since IETF 111. In draft -11

➡ 🟩🟩🟩🟩 **HW Identification**
OEM, model, version…
Unique device identification

➡ 🟩🟩🟩🟩 **SW Identification - CoSWID**
Author, package, version…
Measurement

➡ 🟩🟩🟩🟩 **Security Characterization**
High-level OS, TEE, secure element, TPM…

🟩🟩🟩🟩 **Nonce and Timestamps**
Freshness, prevent replay

🟩🟩🟩🟩 **Identify Verifier Input**
Endorsements, key ID, reference values…

🟩🟩🟩🟩 **Context, Purpose, Profile**
Intended use cases, profile claim

🟩🟩🟩🟩 **GPS Location**

🟩🟩🟩🟩 **Public Keys**
Attestation of private keys on the device (e.g., Android key store)

➡ 🟩🟩🟩🟩 **Submodules**
HW subsystems, TEE, SW process and apps…

➡ 🟩🟩🟩🟩 **Nested EATs**
One EAT inside another, Detached Bundles

🟩🟩🟩🟩 **Running State**
Boot and debug state

➡ ⬜🟩🟩🟩 **Measurement of Running SW**
Runtime integrity check

➡ 🟩🟩🟩🟩 **Formal Device Certifications**
For example, Common Criteria certification; format is GP’s DLOA

➡ ⬜🟩🟩🟩 **Verification Results**
Overall Verification Results, Measurement Results

EAT work needed beyond claims

- Rework introduction and related with respect to RATS Architecture
  - Use Architecture terminology: “Attester”, “Verifier”…
  - Remove most of the architecture-related text currently in EAT

- More examples

- Should a verification procedure be included

DONE

Important changes in the -11 draft (since IETF 111)

- Consistent terminology with RATS Architecture, CWT and JWT
- Remove operating model procudures; rely on RATS Architecture, CWT and JWT instead
- Add a simple software name and software version claim as alternate to CoSWID
- Add DLOAs claim
- Add SW Results claim
- Improved OEMID Claim – It is only for HW, allows PEN to be used, allows randomly generated ones to be used
- Many more, and much improved examples (includes CoSWID examples, DEB example, measurements example)
- Adds universal CDDL for a Claims-Set as used by EAT, CWT, JWT and UCCS (details in following slides)
- Defines UJCS, the JSON equivalent of UCCS
- Clarifications and improvements of nesting one EAT inside another (details in following slides)
- Added Detached EAT Bundles (DEBs) a means of signing detached Claims-Sets

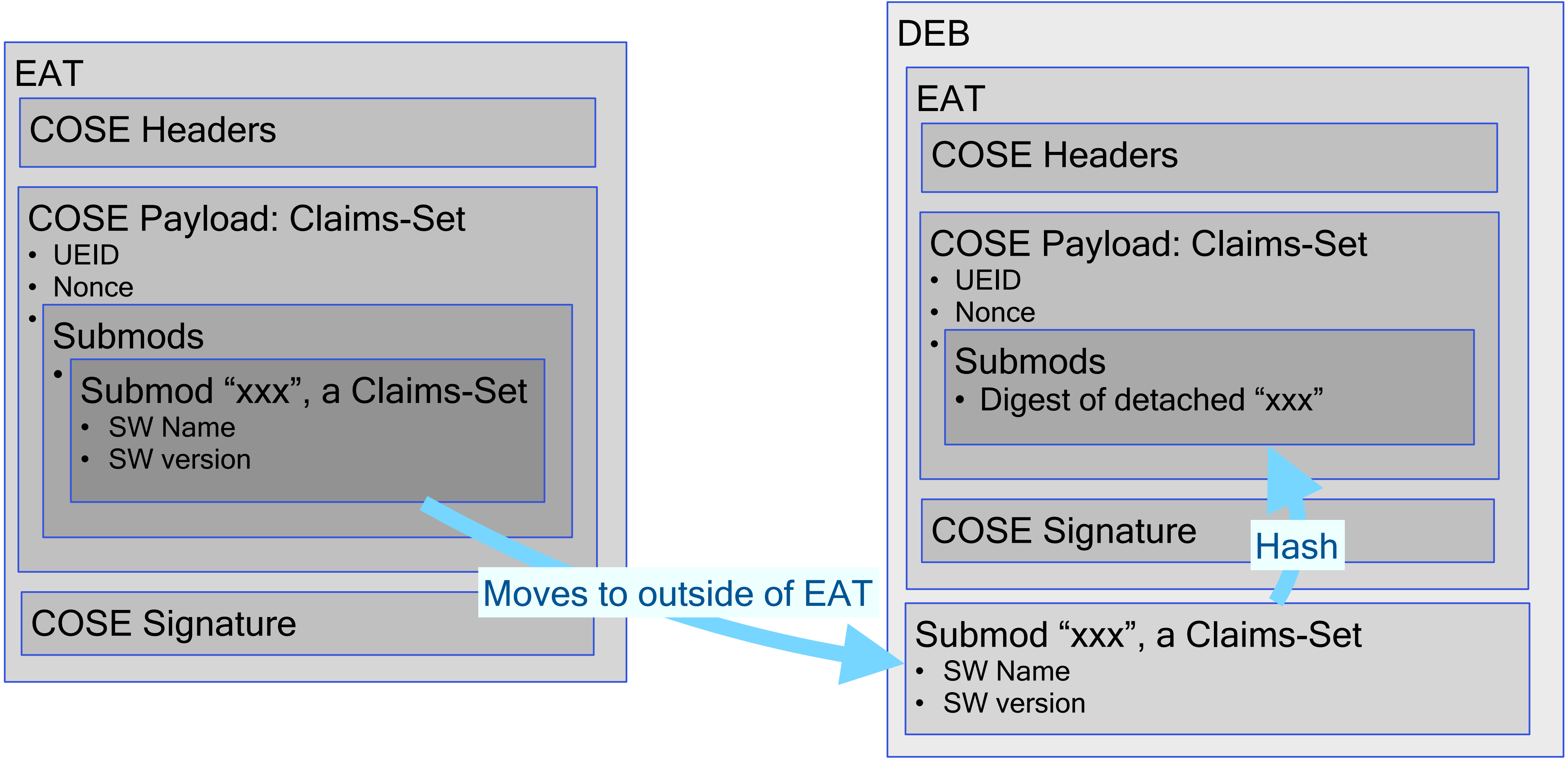
RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

12:55 : 13:25 **EAT Topics: CDDL for Claims-Sets & Nesting CWT in JWT**
(30 min) Laurence Lundblade

DEB – Detached EAT Bundle & Detached Claims-Set

Allows submodule to be a digest of Claims-Set outside of the EAT
DEB one way to bundle the EAT and the detached Claims-Set
Useful for building an EAT-based Attestation HW block (has something kind of like PCRs in a TPM)



CDDL for a Claims-Set for CBOR and JSON

UJCS

Nested EATs of different Encodings

CDDL for CBOR and JSON

- There is general agreement that CDDL can be used to define stuff that can encode in JSON and CBOR
  - Appendix E of CDDL RFC says how to do it
  - Many protocol-defining drafts do this now
  - Consensus in email discussion

- CBOR and JSON will coexist long term
  - CBOR for use cases requiring compactness
  - JSON because backends and B2B are broadly JSON

`Claims-Set` is Central and Useful

- `Claims-Set` – A group of label-value pairs that pertain to a device, a subsystem, a result, a transaction…
- Central to CWT and JWT
- `Claims-Set` is a convenient unit of conveyance between roles and actors in a scheme like RATS or other
- Main structure that is signed and/or encrypted (COSE/JOSE payload)

===> Very Useful to have CDDL for a `Claims-Set`

- Then can define most individual claims in CDDL
- Protocols that need a construct like a `Claims-Set` can just use it off-the-shelf, even non-attestation protocols
- Write CDDL once for either JSON or CBOR

Further…

- Nest one `Claims-Set` in another
- Even a CBOR `Claims-Set` in a JSON `Claims-Set` and vice versa

CDDL for Claims-Set

```
Claims-Set = {
    * $$claims-set-claims,
    * Claim-Label .feature "extended-label" => any
}

Claim-Label = int / text
```

Thanks, Carsten

The central definition of a Claims-Set. Has a CDDL socket into which all claims plug. Can be referred to as the COSE/JOSE payload for CWT and JWT or the main body of UCCS / UJCS.

```
$$claims-set-claims //= (sub-label => text)
```

Definition of a text string claim for both CBOR and JSON

```
sub-label = 2
```

CBOR integer label for above claim

```
sub-label = "sub"
```

JSON text string label for above claim

CDDL for the 7 claims in CWT and JWT

```
$$claims-set-claims //= (iss-label => text)
$$claims-set-claims //= (sub-label => text)
$$claims-set-claims //= (aud-label => text)
$$claims-set-claims //= (exp-label => ~time)
$$claims-set-claims //= (nbf-label => ~time)
$$claims-set-claims //= (iat-label => ~time)

$$claims-set-claims //= (cti-label => bytes)
```

This CDDL works for both JSON and CBOR, JWT and CWT (almost)

Labels, `iss-label, sub-label`,… are not shown. They are integer for CBOR, text for JSON.

Issue with validation using the cddl tool for byte string claims. In CBOR they are bytes. In JSON they are text fields with b64 content

CDDL for UCCS (Unprotected CWT Claims Sets)

```
UCCS-Message = UCCS-Tagged-Message / UCCS-Untagged-Message

UCCS-Tagged-Message = #6.601(UCCS-Untagged-Message)

UCCS-Untagged-Message = Claims-Set
```

It is just a CBOR map of claims that may or may not be a CBOR tag.

CDDL for UJCS (Unprotected JWT Claims Sets, draft-ietf-rats-eat-11)

```
UJCS-Message = Claims-Set
```

JSON has no equivalent of a CBOR tag, so UJCS is nothing but a Claims-Set encoded in JSON

UJCS is currently defined and described in draft-ietf-rats-eat-11

The EAT authors are open to it staying in EAT or moving to UCCS (which would require renaming UCCS)

Why UJCS is important

JSON is far more widely use than CBOR, so if UCCS is important, isn’t UJCS important?

Back ends and B2B

- Primarily and hugely JSON today
- Have many mechanisms in place for integrity, authenticity and privacy (usually TLS)
  - Security added by JWT is not necessary, not deployed, awkward

JWT’s {"alg":"none"} is awkward and adds implementation overhead compared to UJCS

Attestation Results going from Verifier to Relying Party are usually B2B

- JSON is highly appropriate
- Already have security mechanisms (no need for JWT)

Standardizing UJCS

Not much work…

The CDDL is simple (previous slides)

The security considerations from UCCS can be exactly re used

Having UCCS without UJCS is awkward

Going to/from CBOR claims sets to/from JSON Claims-Sets needs more code

- Needs a library to encode/decode JWT {“alg":"none"}

Makes all the nesting constructs in EAT (submodules, detached Claims-Sets) more complex

Today, people send JSON maps of label/value pairs all day long without JWT {“alg”:"none"}

Not really any logical reason why CBOR Claims-Sets can be sent fully in the clear and JSON Claims-Sets must have the JWT {“alg”:"none"} construct

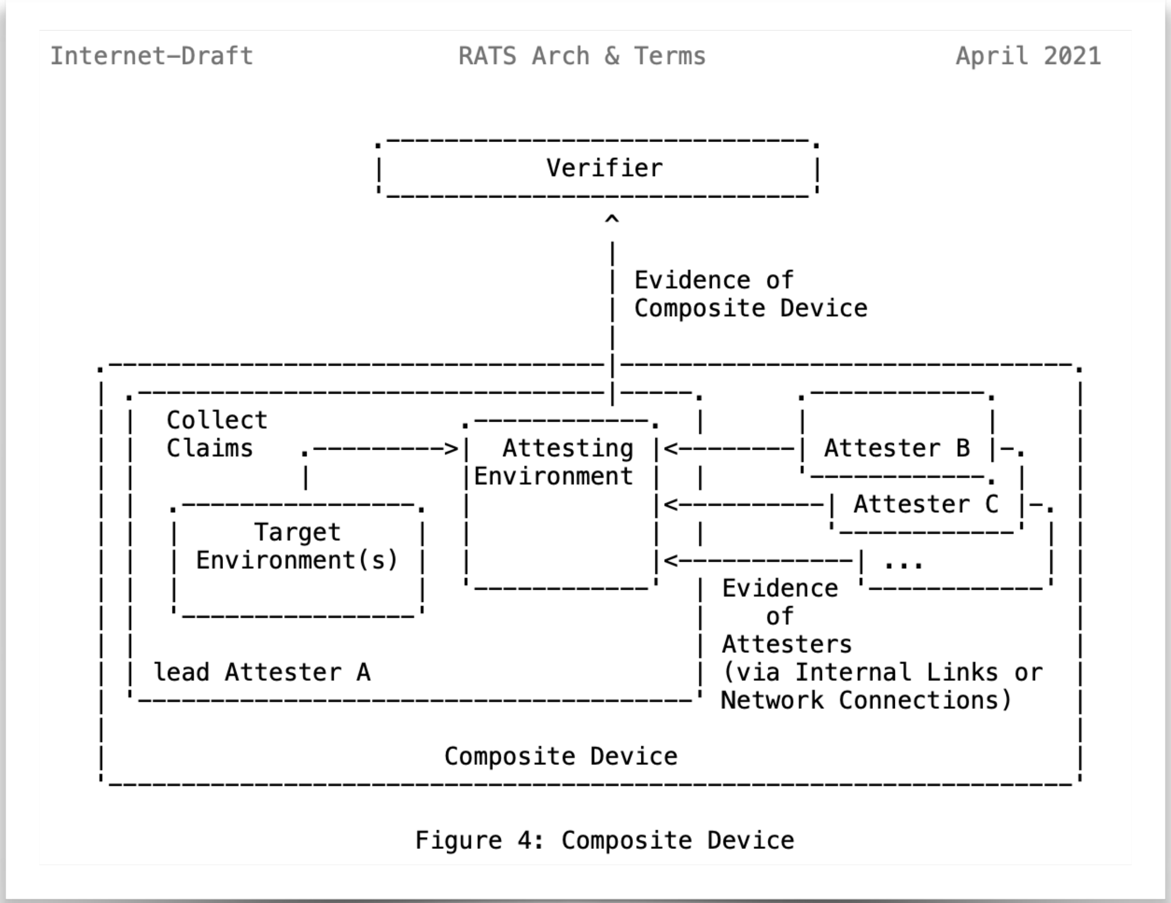
Mixed Encoding Nested Tokens

Q: Why nest CBOR-encoded tokens in JSON-encoded tokens? (and vice versa)

A: Composite Devices & Attesters

- No guarantee or requirement that off-the-shelf Attesters that make up a composite device all use the same encoding
- Nested composite evidence might be signed (COSE or JOSE) or not signed (UCCS or UJCS) depending on use case

Mixed nested encoding is only allowed when nesting tokens. You can't mix claim encoding within a token.



Internet-Draft RATS Arch & Terms April 2021

```
                   .-----------------------------.
                   |           Verifier          |
                   '-----------------------------'
                                   ^
                                   |
                                   | Evidence of
                                   | Composite Device
                                   |
.----------------------------------|-------------------------------.
| .--------------------------------|-----.      .------------.     |
| |  Collect             .------------.  |      |            |     |
| |  Claims   .--------->|  Attesting |<--------| Attester B |-.   |
| |           |          |Environment |  |      '------------. |   |
| |  .----------------.  |            |<----------| Attester C |-. |
| |  |     Target     |  |            |  |        '------------' | |
| |  | Environment(s) |  |            |<------------| ...        | |
| |  |                |  '------------'  | Evidence '------------' |
| |  '----------------'                  |    of                   |
| |                                      | Attesters               |
| | lead Attester A                      | (via Internal Links or  |
| '--------------------------------------' Network Connections)    |
|                                                                  |
|                       Composite Device                           |
'------------------------------------------------------------------'
```

Figure 4: Composite Device

All the EAT Token Formats

All-in-all, there are 6 token formats

Any one can be nested inside the other as a nested token submodule

EAT draft 11 specifies how:

- CBOR tags and byte string wrapping is used when surrounding token is CBOR
- Base64 encoding and a simple JSON structure is used when the surrouding token is JSON. Here it is in CDDL that will always be encoded in JSON format:

```
Nested-Token = [
   type : "JWT" / "CBOR" / "UJCS" / "DEB",
   nested-token : JWT-Message /
                  B64URL-Tagged-CBOR-Token /
                  UJCS-Message /
                  DEB-JSON-Message
]
```

Format	Signed / Encrypted	Encoding
CWT	Yes, COSE	CBOR
JWT	Yes, JOSE No with {“alg”:"none"}	JSON
UCCS	No	CBOR
UJCS	No	JSON
DEB encoded in CBOR	Indirectly through CWT	CBOR
DEB encoded in JSON	Indirectly through JWT	JSON

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

13:25 : 13:30 **EAT Open Issues**
(5 min) Giri Mandyam

EAT: Open issues
IETF 112

Summary

- Only one issue currently classified as LC blocking
- Recommend immediate Last Call

Last Call Blocking
Issue 15: should/must consistency

- All normative language must be review before LC completion
- There has been no additional feedback or review regarding usage of should/must/SHOULD/MUST language in spec since issue was opened
- Issue has been open since 07/15/2019
- Recommend closing issue
  - LC/AD/IESG reviews may turn up additional issues with normative language – can consider during comment resolution

Status of Unclassified Issues

- 2 issues are currently unclassified (neither LC Blocking or ‘wontfix’)
- Issue 131: Fill in list for IANA of all to-be-registered claims
  - Should not be LC blocking
- Issue 135: Say that submodules relate to target environments
  - Related to RATS Arch. relation to EAT document
  - Recommend not addressing prior to LC – comments from WG will determine whether it is required to address

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

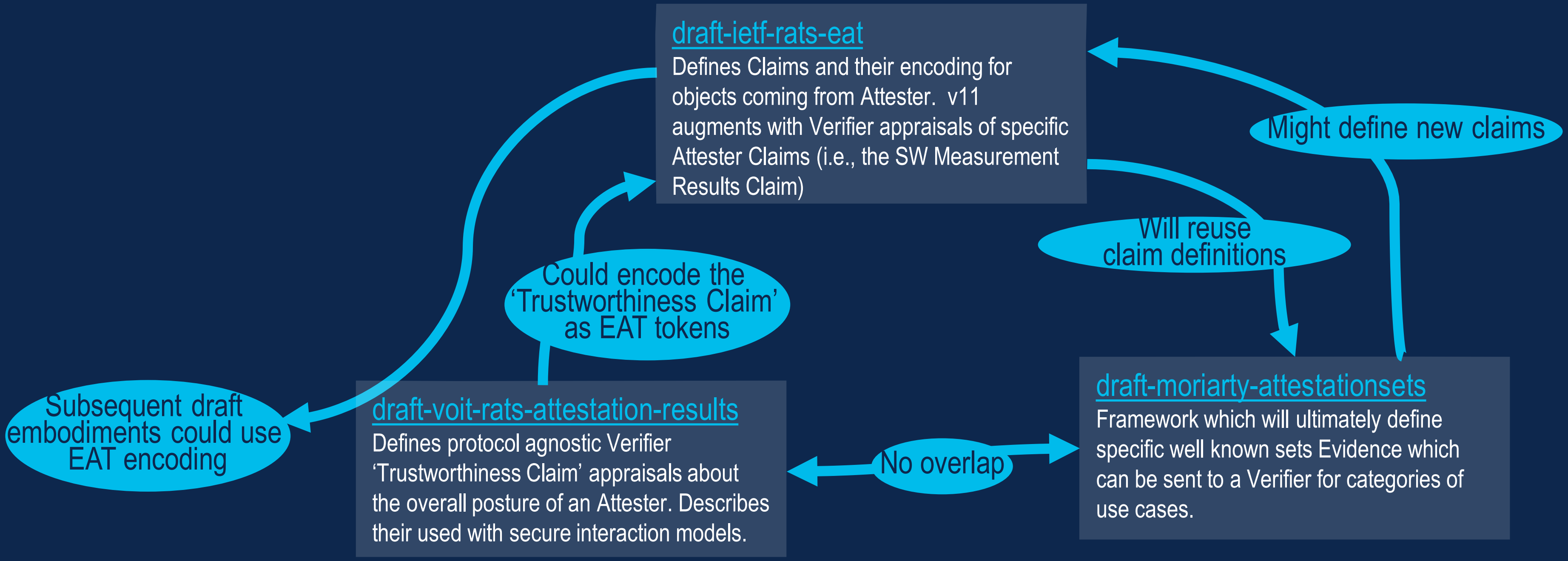
13:30 : 13:40 **WGLC for EAT**
(10 min) RATS Chairs

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

13:40 : 13:50 **Attestation Results for Secure Interactions**
(10 min) Eric Voit (draft-voit-rats-attestation-results-02)

Normative Intersections



draft-ietf-rats-eat

Defines Claims and their encoding for objects coming from Attester. v11 augments with Verifier appraisals of specific Attester Claims (i.e., the SW Measurement Results Claim)

Might define new claims

Will reuse claim definitions

Subsequent draft embodiments could use EAT encoding

Could encode the ‘Trustworthiness Claim’ as EAT tokens

draft-voit-rats-attestation-results

Defines protocol agnostic Verifier ‘Trustworthiness Claim’ appraisals about the overall posture of an Attester. Describes their used with secure interaction models.

No overlap

draft-moriarty-attestationsets

Framework which will ultimately define specific well known sets Evidence which can be sent to a Verifier for categories of use cases.

Attestation Results for Secure Interactions

draft-voit-rats-attestation-results-02

IETF 112, November 2021, RATS WG

Eric Voit
Cisco
evoit@cisco.com

Henk Birkholz
Fraunhofer SIT
henk.birkholz@sit.fraunhofer.de

Thomas Hardjono
MIT
hardjono@mit.edu

Thomas Fossati
Arm Limited
Thomas.Fossati@arm.com

Vincent Scarlata
Intel
vincent.r.scarlata@intel.com

Summary

@ IETF 111
WG requested
document
content
realignment

- Part 1: Information Element definitions for Attestation Results (AR) generated by Verifier to support Secure Interactions between Attester and Relying Party

- Part 2: End-to-end implementation options: (a) Background check, (b) AR Augmented Evidence

- Implementations:
  - Trusted Path Routing (Proprietary – Cisco)
  - Veraison (Open Source, aspiration = Confidential Compute Consortium adoption)

- Ask: WG Adoption after intersections discussed

  draft-ietf-rats-eat
  draft-voit-rats-attestation-results
  draft-moriarty-attestationsets

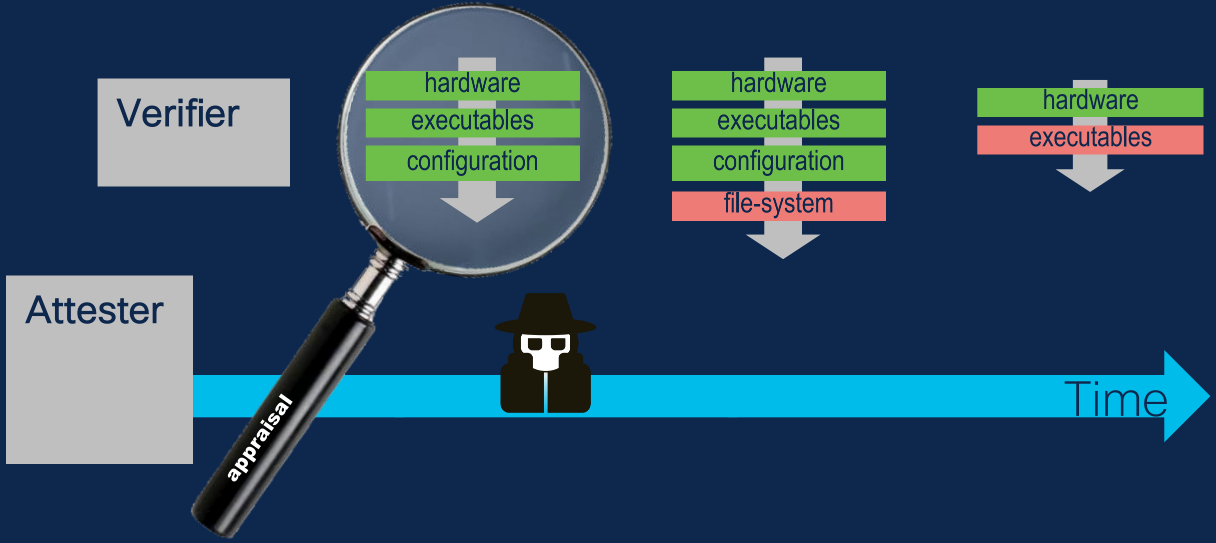
Part 1

Remote Attestation in a Heterogenous World

- Many types of Attesting Environments (AE)
- Relying Party cannot support ∞ language permutations
  - And a mix and match across L1 ↔ L7 platforms is coming if IETF RATS succeeds
- Relying Party needs shared definitions/structures for Verifier Appraisals
  - Will help scale and Interop
  - Reduce transcoding/mapping between sequentially bound sets of Attesters
  - Could be encoded in EAT, YANG, CDDL, etc...

Verifier Appraisal

- Zero to many Trustworthiness Claims assigned during an appraisal cycle.



Part 1

Trustworthiness Claims, simplified since IETF 111

Identity	**instance-identity**	Recognition of Attester via a private key signature which could only have come from that instance of the Attesting Environment
Integrity	**hardware**	Recognition of expected hardware and firmware based on their code fingerprints
	executables	Recognition of runtime files, scripts, and other objects loaded into runtime memory
	sourced-data	Evaluation of the integrity of data objects loaded into memory
	file-system	Recognition of all file system objects which may be utilized
	configuration	Evaluation of the configuration, and conclusions on the exposure of known vulnerabilities
Confidentiality	**runtime-opaque**	Accessibility of Attester objects in memory from outside the Attester but within same physical host
	storage-opaque	Does Attester encrypt its persistent storage

Proposed Encodings of Trustworthiness Claims

Part 1

Identity	**instance-identity**	Recognition of Attester via a private key signature which could only have come from that instance of the Attesting Environment	2: Recognized, affirmed 96: Not recognized, but should be 97: Recognized, contraindicated
Integrity	**hardware**	Recognition of expected hardware and firmware based on their code fingerprints	2: Only genuine/supported Authentic 32: Authentic, but known security bugs 96: Recognized, contraindicated 97: Not recognized, but should be
	executables	Recognition of runtime files, scripts, and other objects loaded into runtime memory	2: Recognized, only genuine/supported 32: Recognized, but known security gaps 33: Some objects loaded not recognized 96: Recognized, contraindicated
	sourced-data	Evaluation of the integrity of data objects loaded into memory	2: comes from affirmed Attesting sources 32: does not come from affirmed 96: Recognized, contraindicated
	file-system	Recognition of all file system objects which may be utilized	2: Recognized, affirmed 32: Some analyzed files not recognized 96: Recognized, contraindicated
	configuration	Evaluation of the configuration, and conclusions on the exposure of known vulnerabilities	2. Known and approved config 3. No known vulnerabilities exposed 32: Known security risk exposed 96: Unsupportable configuration
Confidentiality	**runtime-opaque**	Accessibility of Attester objects in memory from outside the Attester but within same physical host	2: TEE encryption, opaque to device root 32: Target inaccessible by peer Apps 96: Contraindicated or compromised
	storage-opaque	Does Attester encrypt its persistent storage	2: All objects needing privacy encrypted 32: Not all objects need privacy encrypted 96: Secrets are stored unencrypted

Encoded using signed 8-bit integer, intended to simplify RP based Policy evaluation

Affirming (Values 2 to 31): The Verifier affirms the Attester support for this aspect of trustworthiness

Warning (Values 32 to 95): The Verifier warns about this aspect of trustworthiness

Contraindicated (Values 96 to 127): The Verifier asserts the Attester is explicitly untrustworthy regarding this aspect. (99 is always signature verification error.)

None (Values 0, 1, & -1): The Verifier makes no assertions about this Trustworthiness Claim. (0 is no claim, 1 is wrong evidence delivered, -1 is processing error.)

Values under -1: vendor allocatable

Values -2 to -32

Values -33 to -96

Values -97 to -128

Normalizing Trustworthiness Claims (Informational /Appendix)

Part 1

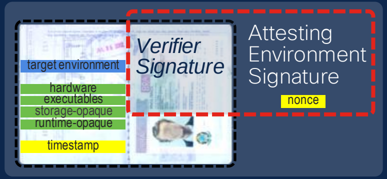
	Trustworthiness Claim	Protection Technologies		
		Process-based	VM-based	HSM-based
Identity	instance-identity	Optional	Optional	Optional
Integrity	hardware	Implicit	Chip dependent	If PCR check ok
	executables	Optional	Optional	If PCR check ok
	sourced-data	Optional	Optional	Optional
	file-system	Optional	Optional	Insufficient
	configuration	Optional	Optional	Optional
Confidentiality	runtime-opaque	Implicit	Implicit	Very limited support
	storage-opaque	Implicit	Chip dependent	Very minimal space

Normalized Trustworthiness Claims ≠ the same Relying Party policy disposition

- Even with Normalized Trustworthiness Claims, Attesters need not be treated equivalently by the Relying Party
    - Variance in underlying protections of SGX, TrustZone, SEV, TPM, etc. could mean different disposition via the Appraisal Policy for Attestation Results.
    - Each Verifier, or Verifier version, or Verifier appraisal of a specific type of Attester may be trusted differently for different claims

Attestation Results Augmented Evidence

- Evidence the Relying Party might Action bundled by Attester
- Signatures protect from manipulation



Verifiable Identity instance(s) + Trustworthiness Claims of the Verifier + Verifiable Freshness

Attester	chip vendor chip type target environment target developer instance
Verifier	verifier id verifier developer

- Categories defined in this draft
- Specific objects to be defined in other drafts

Identity	instance-identity
Integrity	hardware executables configuration file-system sourced-data
Confidentiality	runtime-opaque storage-opaque

Defined in this draft

Random Number	nonce
Synchronized Clocks	timestamp tuda sync token
Epoch	epoch id

- Categories defined in draft-ietf-rats-architecture Section 10

Trustworthiness Claim Delivery

Based on draft-ietf-rats-architecture: Passport Model

```
  .----------------.
  | Attester       |
  | .-------------.|
  | | Attesting   ||                 .----------.     .---------------.
  | | Environment ||                 | Verifier |     | Relying Party |
  | '-------------'|                 |     A    |     |  / Verifier B |
  '----------------'                 '----------'     '---------------'
        time(VG)                           |                 |
          |<------Verifier PoF-------time(NS)                |
          |                                |                 |
 time(EG)(1)------Evidence------------>|                     |
          |                          time(RG)                |
          |<------Attestation Results-(2)                    |
          ~                                ~                 ~
        time(VG')?                         |                 |
          ~                                ~                 ~
          |<------Relying Party PoF-----------------(3)time(NS')
          |                                |                 |
time(EG')(4)------AR-augmented Evidence----------------->|
          |                                |   time(RG',RA')(5)
                                                        (6)
                                                         ~
                                                      time(RX')
```

Summary

@ IETF 111
WG requested
document
content
realignment

- Part 1: Information Element definitions for Attestation Results (AR) generated by Verifier to support Secure Interactions between Attester and Relying Party

- Part 2: End-to-end implementation options: (a) Background check, (b) AR Augmented Evidence

- Implementations:
  - Trusted Path Routing (Proprietary – Cisco)
  - Veraison (Open Source, aspiration = Confidential Compute Consortium adoption)

- Ask: WG Adoption after intersections discussed

  draft-ietf-rats-eat
  draft-voit-rats-attestation-results
  draft-moriarty-attestationsets

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

13:50 : 13:55 **Trusted Path Routing**
(5 min) Eric Voit (draft-voit-rats-trustworthy-path-routing-04)

Trusted Path Routing

draft-voit-rats-trustworthy-path-routing-04

IETF 112, November 2021, RATS WG

Eric Voit
Cisco
evoit@cisco.com

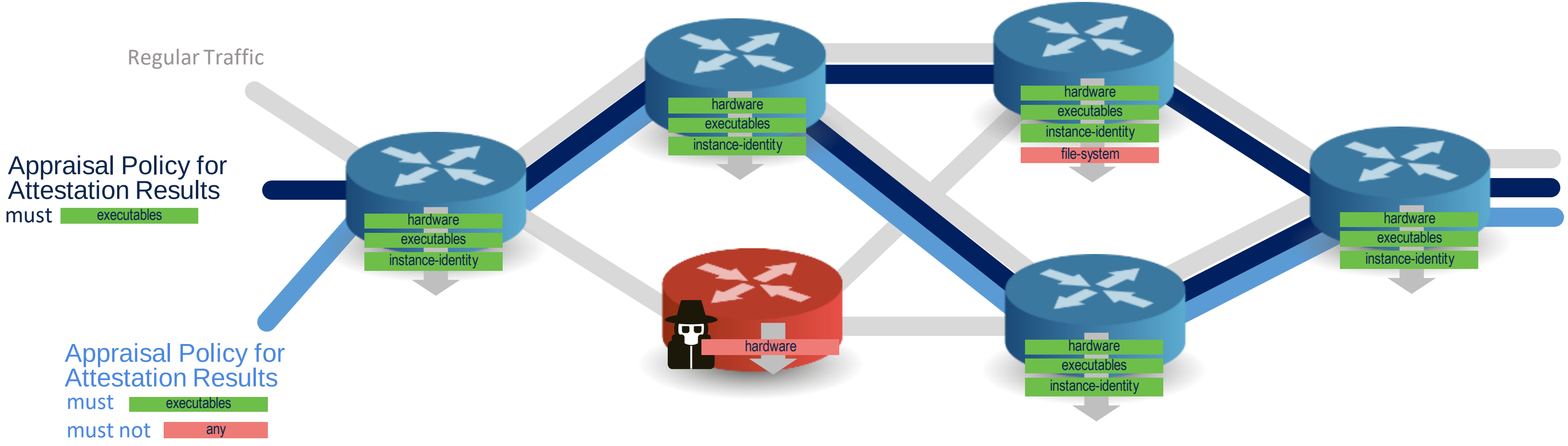
Chennakesava Reddy Gaddam
Cisco
chgaddam@cisco.com

Guy Fedorkow
Juniper
gfedorkow@juniper.net

Henk Birkholz
Fraunhofer SIT
henk.birkholz@sit.fraunhofer.de

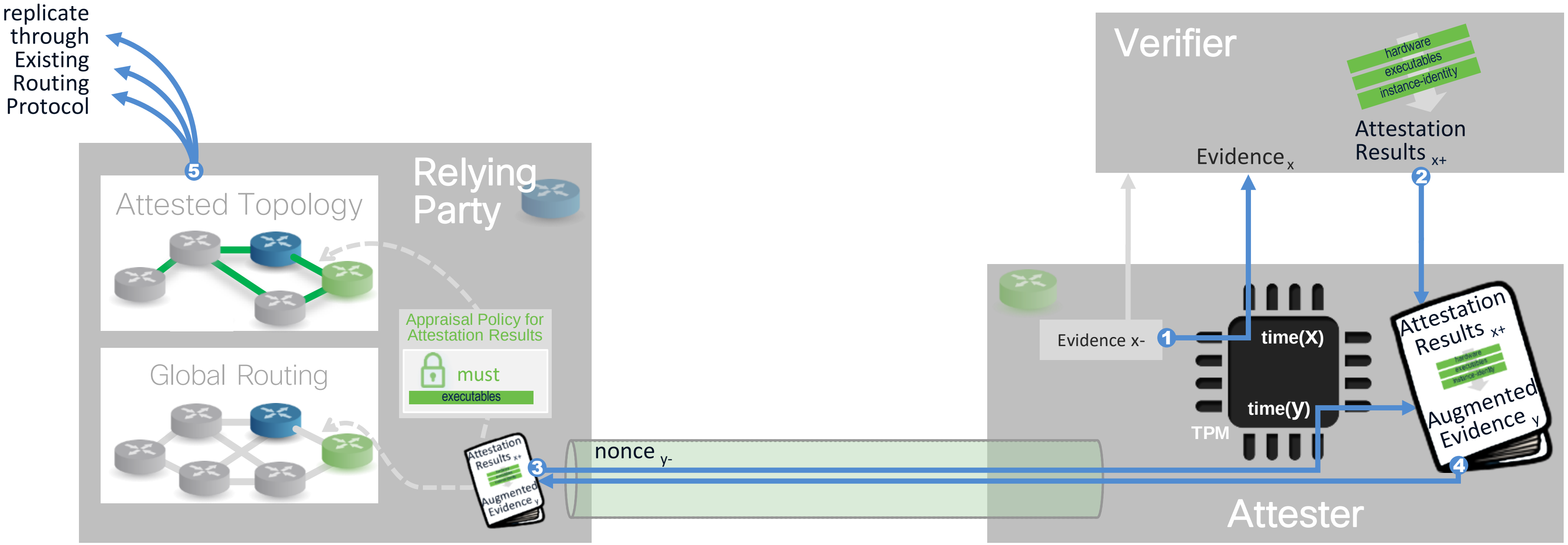
Trusted Path Routing

- Custom topologies dynamically maintained based on Attestation Results
- Instance of draft-voit-rats-attestation-results



Trusted Path Routing

- Link adjacencies added to Trusted Topology based on latest Relying Party’s appraisal of AR Augmented Evidence



Changed since last draft version

- Alignment to latest draft-voit-rats-attestation-results:
  - Trustworthiness Claims

Next Steps

- Continued alignment with draft-voit-rats-attestation-results
- Definition of EAP payload (separate draft)
- Not relevant to adopt until WG adopts draft-voit-rats-attestation-results (fully dependent)

RATS Agenda - Monday, November 8th – Session I

RATS Session 1, Room 7
Time zone: UTC, 2 hrs

13:55 : 14:00 **Scalable Remote Attestation for Systems, Containers, and Applications**
(5 min) Kathleen Moriarty (draft-moriarty-attestationsets-03)

Automation at Scale

Remote Attestation Sets

Kathleen M Moriarty
Center for Internet Security

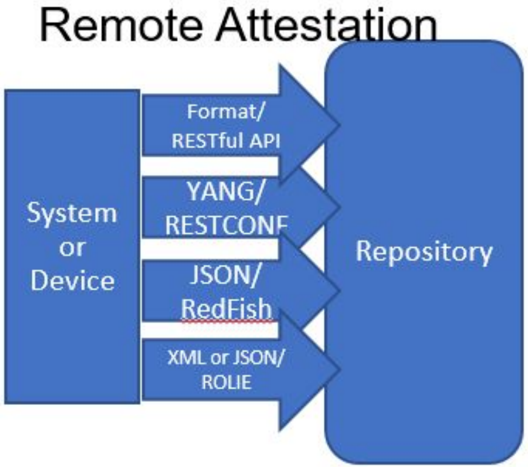
March 2021

Scaling Assessment

- Current posture assessment requires add-on tools to assess systems against expected policies and measurements. Current methods require expertise at each organization.
  - This requires distributed expertise to customize the current standards-based methods to access and collect assessments (e.g OVAL/XCCDF, SWIMA/NEA)
  - APIs are also used to gather information on software inventory or configuration data
- Trusted boot processes occur using attestation locally against a set of policies and measurements established by the vendor, aligned to both NIST SP 800-193 and TCG’s Reference Integrity Measurements
  - What if the local attestations were grouped as a set with log evidence to provide remote reporting? Could this simplify the model for assessment as it is provided and the local attestations must meet criteria for the boot process to continue in this example.

Attestation Local and Remote

- Attestation is essentially signed evidence from a root of trust (RoT)
- Attestations are verified to ensure the signer is trusted
- Evidence in attestations are matched against expected policies or measurements
- If expectations are not met, remediation occurs
- Zero Trust requires verification, identification, encryption, and logs
- Attestation provides verification to the subsequent processes, applications, modules, etc. before execution is permitted
- Attestation aligned to policy sets and are typically performed on system
- Remote attestation is shared through a RESTful interface



Local attestation data generated from boot and runtime measurements and configuration for all managed systems, how to scale remote?

Scaling Measured Trust: Attestation Sets

Attestation Sets to specified policy & measurements per component (e.g. NIST, TCG, CIS Benchmarks, etc.), remediated and verified per set on system.

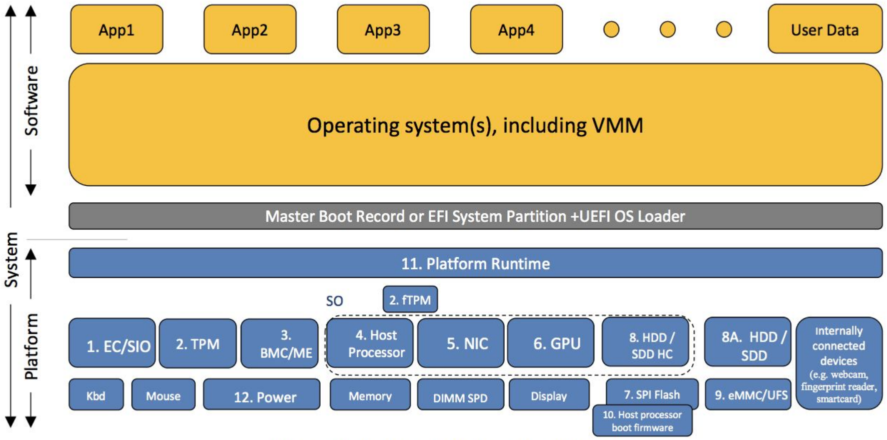
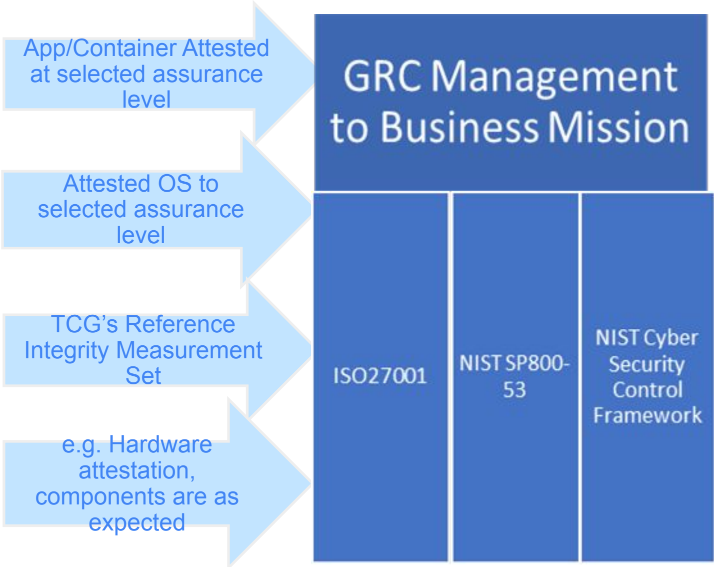


Figure 1: High-Level System Architecture

Image: NIST SP 800-193

Controls and Benchmarks verified locally using known frameworks, controls, or benchmarks (e.g. NIST, CIS Benchmarks, TCG, DISA STIGS, etc.)

Remote Attestation at Scale: **Attestations Aligned** to control frameworks



App/Container Attested at selected assurance level

Attested OS to selected assurance level

TCG's Reference Integrity Measurement Set

e.g. Hardware attestation, components are as expected

GRC Management to Business Mission

ISO27001

NIST SP800-53

NIST Cyber Security Control Framework

Attestation on set of locally verified attestations

Mapping to Control Frameworks and Risk Alignment

Attestation Set Draft Establishes a Registry

- Determine if the proposed information is the right set for reporting in a set
  - (Identifier, Attestation Set Name, Integrity Protected Log of attestation evidence verification for set, timestamp, other useful claims) Signed by Trusted Platform Module or software RoT
  - Establish a registry for the set names to enable remote attestations in sets
    - Levels may be needed in the case of Benchmark or assurance to hardening guides as decisions may vary for applications.
    - The set may contain the policy **or** measurement values from a standard such as NIST SP 800-193
    - The set may be aligned to all or part of a standard
    - The set may be complemented by other assessment types, but still having the goal of reducing the distributed assessment criteria and programming - the vendor would be responsible for built-in security and ongoing assurance automation
- Format: Entity Attestation Token (JWT or CWT)
- Protocol: RESTful interface (e.g. RedFish, ROLIE, etc.)

Thank You

Comments welcome and appreciated!

URL:	https://www.ietf.org/archive/id/draft-moriarty-attestationsets-03.txt
Status:	https://datatracker.ietf.org/doc/draft-moriarty-attestationsets/
Htmlized:	https://datatracker.ietf.org/doc/html/draft-moriarty-attestationsets
Htmlized:	https://tools.ietf.org/html/draft-moriarty-attestationsets-03

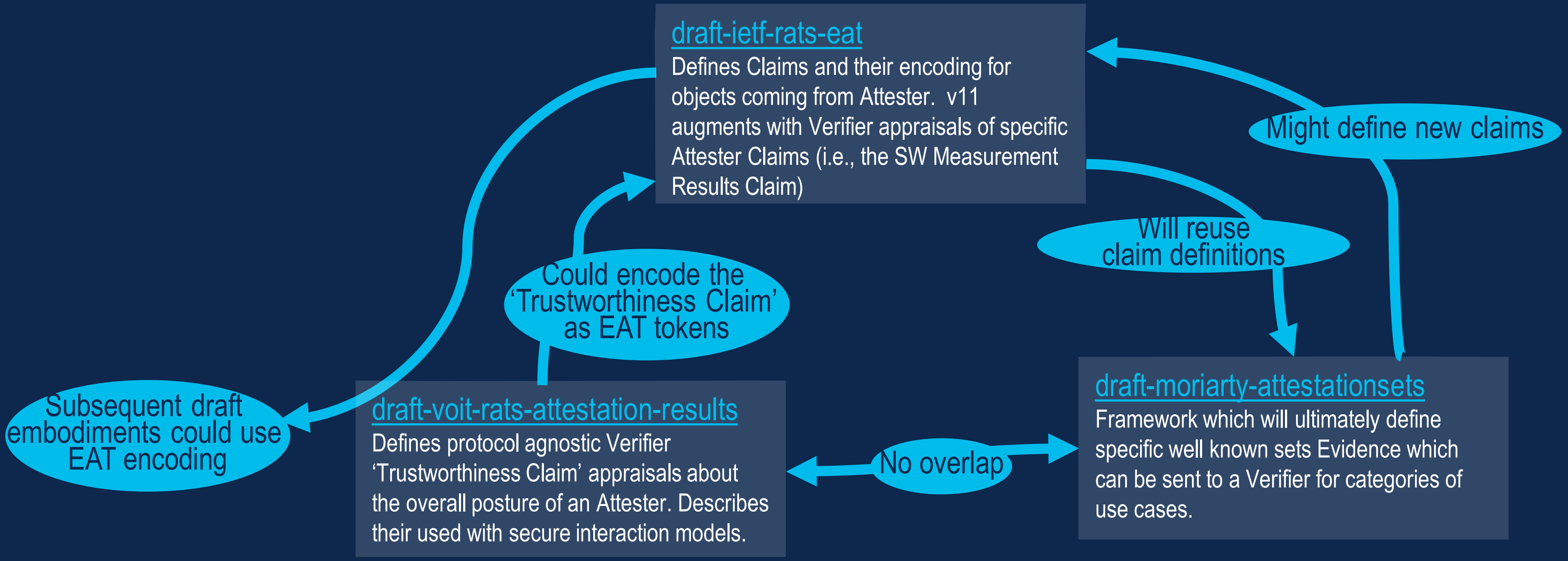
Thank You!

RATS Agenda – Friday, November 12th – Session II

Room 7, RATS Session 2
Time zone: UTC, 1 hr

14:35 : 14:50 **Overlap between Attestation Results, EAT and Attestation Sets**
(15 min) Eric Voit, Laurence Lundblade, Kathleen Moriarty, Giri Mandyam

Normative Intersections



draft-ietf-rats-eat

Defines Claims and their encoding for objects coming from Attester. v11 augments with Verifier appraisals of specific Attester Claims (i.e., the SW Measurement Results Claim)

Could encode the 'Trustworthiness Claim' as EAT tokens

Subsequent draft embodiments could use EAT encoding

draft-voit-rats-attestation-results

Defines protocol agnostic Verifier 'Trustworthiness Claim' appraisals about the overall posture of an Attester. Describes their used with secure interaction models.

No overlap

Will reuse claim definitions

Might define new claims

draft-moriarty-attestationsets

Framework which will ultimately define specific well known sets Evidence which can be sent to a Verifier for categories of use cases.

RATS Agenda – Friday, November 12th – Session II

Room 7, RATS Session 2
Time zone: UTC, 1 hr

14:50 : 14:55 **Direct Anonymous Attestation**
(5 min) Henk Birkholz (draft-birkholz-rats-daa-02)

RATS Agenda – Friday, November 12th – Session II

Room 7, RATS Session 2
Time zone: UTC, 1 hr

14:55 : 15:15 **Concise Reference Integrity Manifest**
(20 min) Henk Birkholz, Thomas Fossati (draft-birkholz-rats-corim-01)

IETF 112 RATS WG
Concise Reference Integrity Manifests

12 November 2021, Session II, notinmadrid

https://datatracker.ietf.org/doc/draft-birkholz-rats-corim/

Henk Birkholz <henk.birkholz@sit.fraunhofer.de>,
Thomas Fossati <thomas.fossati@arm.com>,
Yogesh Deshpande <yogesh.deshpande@arm.com>,
Ned Smith <ned.smith@intel.com>,
Wei Pan <william.panwei@huawei.com>,

Internet Engineering Task Force

Production by Meetecho

Quick Recap on CoRIM

- Mission Statement: a "sea of triples" to describe Attesters to Verifiers
- Initial cut includes:
  - Reference values
  - Verification key material
  - Endorsed values (e.g., certification status of a module)
- Also, eventually:
  - Representation of allowed/expected hierarchical composition of modules in an Attester
  - A module's firmware life-cycle (i.e., update/patch)
  - Anything else! – bring your own triple to the group and we'll do design team sessions

Extensibility and widely available codepoints!

(see RATS Architecture Figure 9: Multiple Attesters and Relying Parties with Different Formats)

CoRIM Applicability

- TCG DICE (by definition, especially to Layered Attestation)
- ARM PSA Token, an EAT profile (see draft-fdb-rats-psa-endorsements)
- Concise TPM-based Evidence in enterprise setting

Flexibility and Interoperability!

Specs Status

- Information model described in TCG's "DICE Endorsements Architecture" (under ballot, not yet public, a matter of weeks)
- Data Model specified in https://www.ietf.org/archive/id/draft-birkholz-rats-corim-01.html
  - Bleeding edge CDDL @ github.com/ietf-rats/ietf-corim-cddl

Implementation Status

Go packages (Apache 2.0 license, closely tracking upstream spec):

  - https://github.com/veraison/corim/corim
    - Low-level CoRIM manipulation – CBOR, JSON (bespoke) codecs
  - https://github.com/veraison/corim/comid
    - Low-level CoMID manipulation – CBOR, JSON (bespoke) codecs
- https://github.com/veraison/swid
  - CBOR (CoSWID, draft-ietf-sacm-coswid) and JSON (bespoke)
  - XML (SWID, ISO/IEC 19770-2:2015, NISTIR-8060),
- github.com/veraison/corim/cocli
  - Command Line Interface to deal CoRIMs, CoMIDs and CoSWIDs, for the (supply chain) end user

CoRIM & the RATS Charter Scope – Charter Goals

- Current charter's **goals addressed by CoRIM**
  - CoRIM **standardizes formats for describing assertions about system components** in the form of reference values, endorsed values, and environment endorsements based on their environment identity. These **assertions are directly associated with Evidence** as they are used in the appraisal procedures conducted by Verifiers in order to generate Attestation Results
  - CoRIM **content is protected using COSE signing capabilities**
  - CoRIMs are **intended to be consumed by Verifiers** (and not Relying Parties) and they suppliy the data inputs that enable a Verifier's appraisal procedures. The inputs originate from supply chain entities. CoRIMs **do not supply Appraisal Policies for Verifiers** in support of their appraisal procedures.
  - CoRIMs are specified **in collaboration with several supply chain stakeholders** that provide solutions for Attesting Environments designs and **in cooperation with the TCG**

CoRIM & the RATS Charter Scope – Charter Deliverables

- Current charter's **program of work defined deliverables addressed by CoRIM**
  - CoRIM involves the "system component providers" (e.g., OEM or ODM) by enabling them to provide conceptual message content, such as reference values about the Attester, endorsed values about the Attester and requirements (i.e., identity identifiers) on signing key material of the Attester, which is content of **deliverable two**.
  - CoRIM specifies a manufacturer's, OEM's, and others supply chain entities' requirements on providing information about system components characteristics of an Attester (described in, e.g., use case 2.1, 2.3, or 2.4), which is content of **deliverable three.**
  - CoRIM also standardizes a corresponding to data model the implement and secure the defined information model using a COSE like manifest similar to SUIT, which is content of **deliverable four**.

Next Step: Call of for WG Adoption (WGAC)

- Editor's version work items are documented in:
  - https://github.com/ietf-rats/draft-birkholz-rats-corim/issues
    - 8 open, 34 closed
  - https://github.com/ietf-rats/ietf-corim-cddl/
    - 26 open, 63 closed
- The editor's version is now in a fairly stable state
- It's the output of eleven months of thrice-weekly design meetings involving multiple Attesting Environments manufactures and various cross-SDO inputs and corresponding consensus
- The authors think this document is ready for adoption and in alignment with the current RATS charter

RATS Agenda – Friday, November 12th – Session II

Room 7, RATS Session 2
Time zone: UTC, 1 hr

15:15 : 15:30 **Open Mic**
(15 min) RATS Chairs

Thank You!