EAT
Entity Attestation Token

CCC Attestation SIG

2022-06-07

Scoping

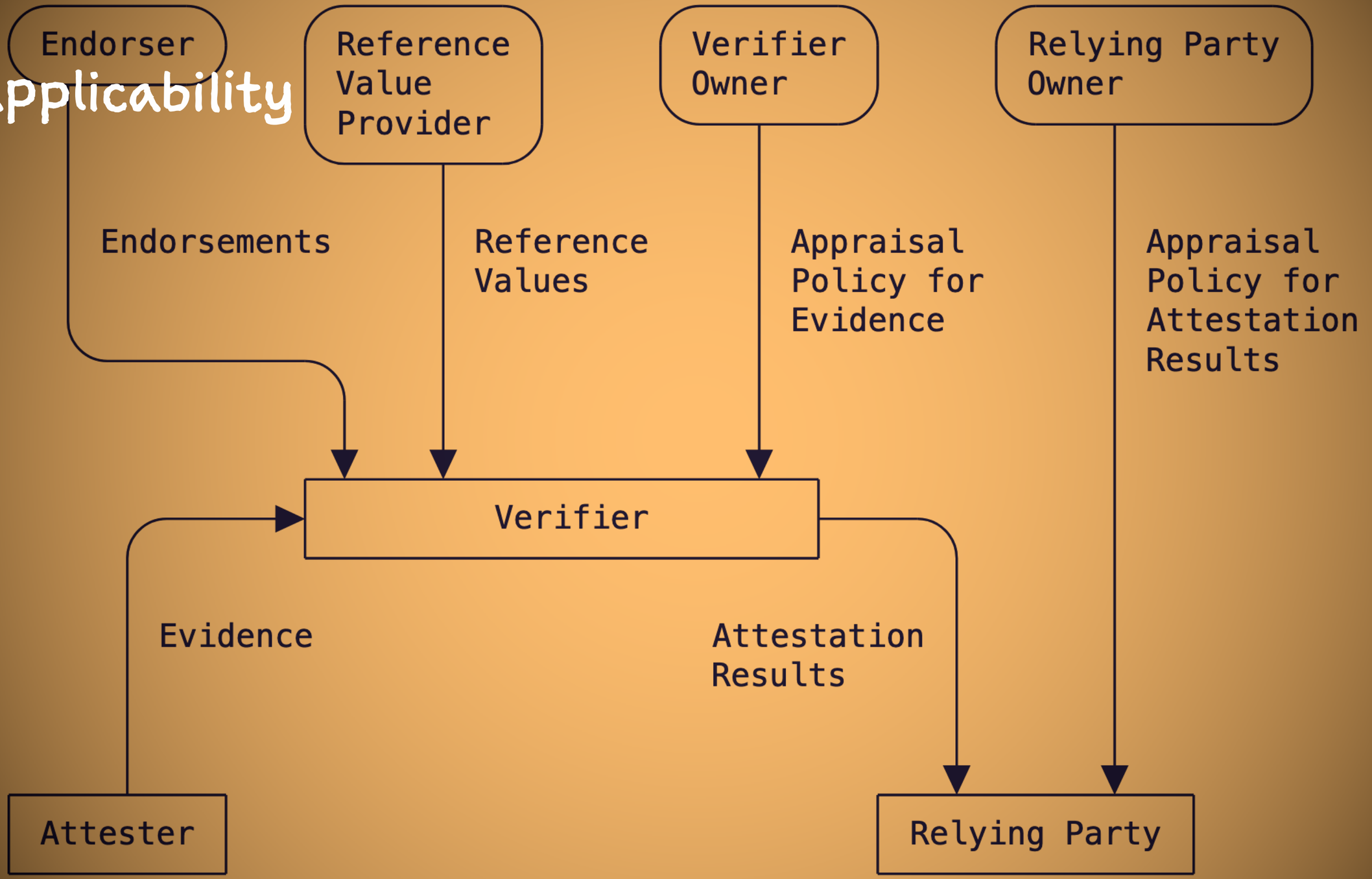
What EAT **is not:**

- an off-the-shelf interoperable message format

What EAT **is:**

- a flexible **framework** that provides building blocks for creating and consuming RATS conceptual messages

Applicability



Some Context

- IETF->Security Area->Remote ATtestation ProcedureS (RATS) WG
- Internet Draft, targeting the Standards Track
- Adopted June 2019, currently at rev -13 (released May 2022)
- It's currently in its 2nd WGLC

Bookmarks

Published revisions:

→ https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat

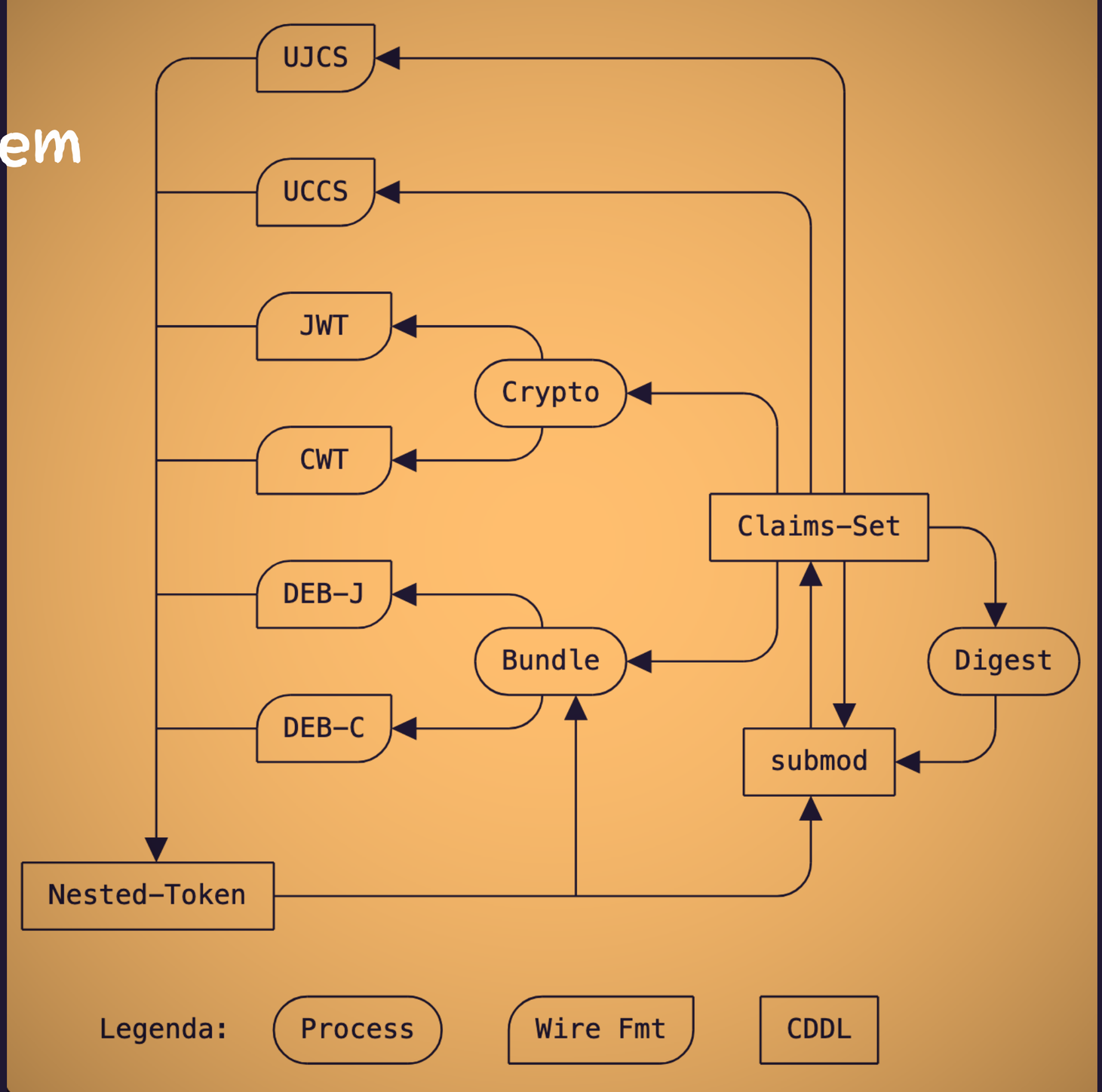
Editor copy:

→ https://github.com/ietf-rats-wg/eat

EAT building blocks

- A type system
  - Claims-set & a few aggregation types
- A number of pre-defined "claims"
  - readily reusable pieces of semantics with a codepoint attached
- CBOR and JSON serialisations
- Cryptographic envelopes based on COSE and JOSE

Type System


UJCS
UCCS
JWT
Crypto
CWT
Claims-Set
DEB-J
Bundle
Digest
DEB-C
submod
Nested-Token
Legenda:
Process
Wire Fmt
CDDL

Vocabulary (already registered)

See CWT Claims and JWT Claims registries @ IANA

code-point	claim name	semantics
10	nonce	Challenger input
256	ueid	Unique Entity ID
257	sueids	Semi-permanent UEIDs
258	oemid	Hardware OEM ID
259	hwmodel	Hardware Model
260	hwvers	Hardware Version
262	secboot	Secure Boot
263	dbgstat	Debug Status
264	location	The geographic location
265	eat_profile	EAT profile
266	submods	The section containing submodules

Vocabulary (pending publication)

code-point	claim name	semantics
?	seclevel	Security Level
?	uptime	Uptime
?	bootseed	Per-boot unique seed
?	intuse	Intended use
?	dloas	DLOAs
?	swname	Software Name
?	swversion	Software Version
?	manifests	Manifests
?	swevidence	Software Evidence
?	measres	Measurement Results
?	odometer	Odometer

Serialisation(s) and security envelopes

- CBOR, JSON
- CWT, JWT

Profiling

- A profile defines the **exact shape** of a EAT message by constraining the parameters that prod/cons need to understand in order to interoperate
- The profile claim carries in-band information about the specific profile
- Code-point 265 (aka `profile`), type is OID or URI
  - Note: no need to register a profile name (as long as one owns a Domain Name or an OID arc)

Profile Examples

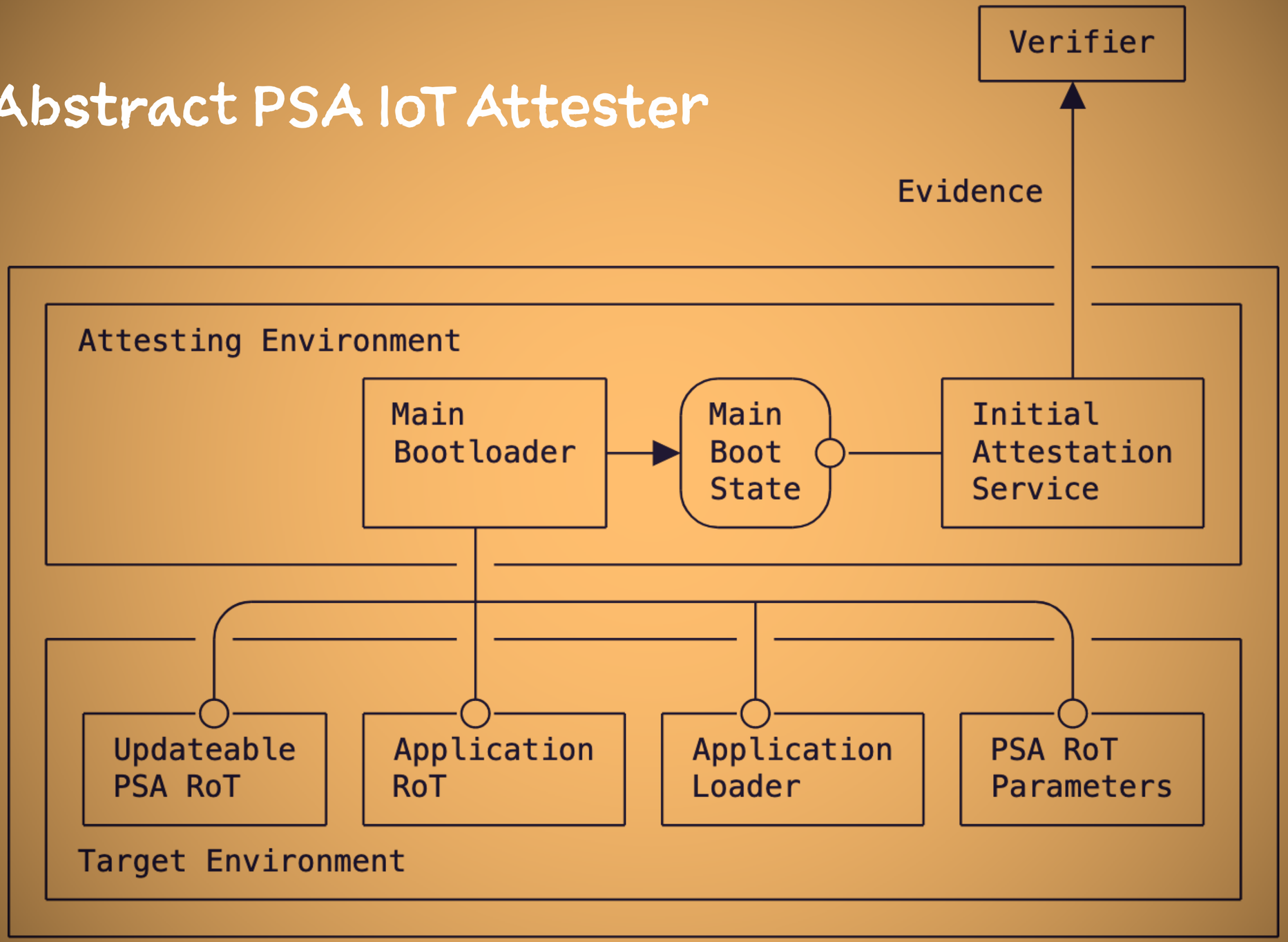
Some things we (Arm) have been working on:

* PSA - https://datatracker.ietf.org/doc/draft-tschofenig-rats-psa-token

* AISS - https://datatracker.ietf.org/doc/draft-tschofenig-rats-aiss-token

* Arm CCA (incoming)

The Abstract PSA IoT Attester



PSA Claims-Set

- Attester Identification (`EAT.euid`, `PSA.impl-id`)
- Target State (`EAT.boot-seed`, `PSA.security-lifecycle`)
- Software Inventory (`PSA.software-components`)
- Challenger Nonce (`EAT.nonce`)
- Profile identifier (`EAT.profile=="http://arm.com/psa/2.0.0"`)

PSA Encoding & Security

- CBOR
- COSE Sign1 / Mac0

Misc considerations

- Freshness model (nonce-based)
- Any expectation on the Verification side (any checks and endorsements needed)

(Pretend) PSA Claims-Set

```
{
  / eat-profile /                265: "http://arm.com/psa/2.0.0",
  / psa-client-id /             2394: 1,
  / psa-lifecycle /             2395: 12288,
  / eat-ueid /                   256: h'01A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3',
  / psa-implementation-id /     2396: h'505152535455565750515253545556575051525354555657505152535455565750515253545556575051525354555657',
  / psa-boot-seed /             2397: h'DEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEF',
  / psa-software-components /   2399: [
    {
      / measurement type /  1: "BL",
      / measurement value / 2: h'000102040001020400010204000102040001020400010204000102040001020400010204',
      / signer ID /         5: h'519200FF519200FF519200FF519200FF519200FF519200FF519200FF519200FF'
    },
    {
      / measurement type /  1: "PRoT",
      / measurement value / 2: h'0506070805060708050607080506070805060708050607080506070805060708',
      / signer ID /         5: h'519200FF519200FF519200FF519200FF519200FF519200FF519200FF519200FF'
    }
  ],
  / eat-nonce /           10: h'000102030001020300010203000102030001020300010203000102030001020300010203'
}
```

(Pretend) PSA Evidence

```
18(
  [
    / protected /   h'A10126',
    / unprotected / {},
    / payload /     h'AA19010978168747470 3A2F2F61726D2E636F6D2F7073612F322E302E3019095A0119095B19300019095C5820505152535455565750515253545556575051525354555657
505152535455565750515253545556575051525354555671 9095D5820DEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEFDEADBEEF19095E
6D313233343536373839303132331 9095F82A30162424C025820000102040001020400010204000102040001020400010204000102040001020405582051920 0
FF519200FF519200FF519200FF519200FF519200FF519200FF519200FFA3016450526F5402582005060708050607080506070805060708050607080506070805
060708050607080558205192 00FF519200FF519200FF519200FF519200FF519200FF519200FF519200FF519200FF0A5820000102030001020300010203000102
030001020300010203000102031901005821 01A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A3A0A1A2A319096078186874747073 3A2F2F
7073612D76657269666965722E6F7267',
    / signature /   h'E3B80C143403ECB744B1D6EF732872A1A3E682783E939F72A3CEF6BF74EF4BC5E7065725FF5C948770B673C5896D3F796F55D144FC
B456BEA832EB13E8258DB8'
  ]
)
```

Do it yourself

- Proprietary Claims only (plus maybe `EAT.profile`)
- Standard Claims
- A mix of proprietary and standard (a la PSA)

Proprietary Claims

- No registration needed
- For CWT just grab code-points < -65536
- For JWT be careful with your choice (use a prefix with low collision chances)

Standard Claims (CWT)

Section 9.1 of RFC8392

"Registration requests are evaluated using the criteria described in the Claim Key instructions in the registration template below after a three-week review period on the cwt-reg-review@ietf.org mailing list, on the advice of one or more Designated Experts [...]"

CWT Registration Policies

Range	Registration Procedures
Integer values from -256 to 255	Standards Action
Integer values from -65536 to -257	Specification Required
Integer values from 256 to 65535	Specification Required
Integer values greater than 65535	Expert Review
Strings of length 1	Standards Action
Strings of length 2	Specification Required
Strings of length greater than 2	Expert Review

Standard Claims (JWT)

Section 10.1 of RFC7519

"Values are registered on a Specification Required basis after a three-week review period on the jwt-reg-review@ietf.org mailing list, on the advice of one or more Designated Experts."

Using EAT with REST APIs

- EAT media types? In a parallel document:

EAT type	Media Type
EAT CWT	application/eat-cwt
EAT JWT	application/eat-jwt
EAT CBOR DEB	application/eat-deb+cbor
EAT JSON DEB	application/eat-deb+json
EAT UCCS	application/eat-ucs+cbor
EAT UJCS	application/eat-ucs+json

- Carry profile as an (optional) media type parameter

Example

-> request

```
POST /challenge-response/v1/session/1234567890 HTTP/1.1
Host: verifier.example
Accept: application/eat-jwt; profile=tag:ar4si.example,2021
Content-Type: application/eat-cwt; profile=tag:evidence.example,2022

[ CBOR-encoded EAT w/ profile=tag:evidence.example,2022 ]
```

<- response

```
HTTP/1.1 200 OK
Content-Type: application/eat-jwt; profile=tag:ar4si.example,2021

[ JSON-encoded EAT w/ profile=tag:ar4si.example,2021 ]
```

Parting Words