SPDM Support for IETF EAT

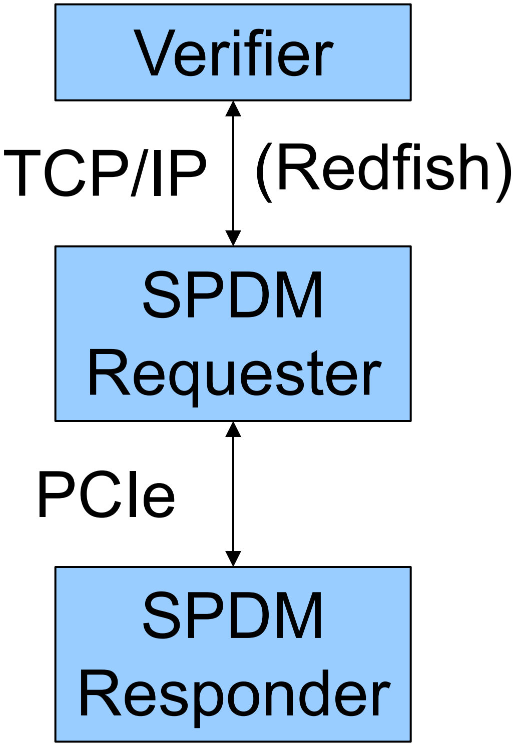
www.dmtf.org

Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF SPDM Working Group.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the DMTF website.
- This information is a summary of the information that will appear in the specifications. See the specifications for further details.

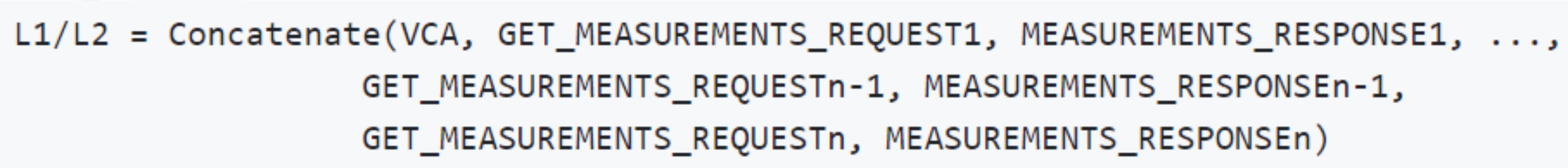
Simple Requester / Verifier / Responder Model

1. Requester retrieves Responder’s x.509 certificate chain.
2. Verifier and/or Requester generates a nonce.
3. Requester sends GET_MEASUREMENTS with nonce and requests all measurement indices to be present and response to be signed.
4. Responder replies with MEASUREMENTS response and signature over L1/L2 transcript.
5. Requester sends certificate chain, measurement transcript, and signature to Verifier for verification.



Measurement Transcript

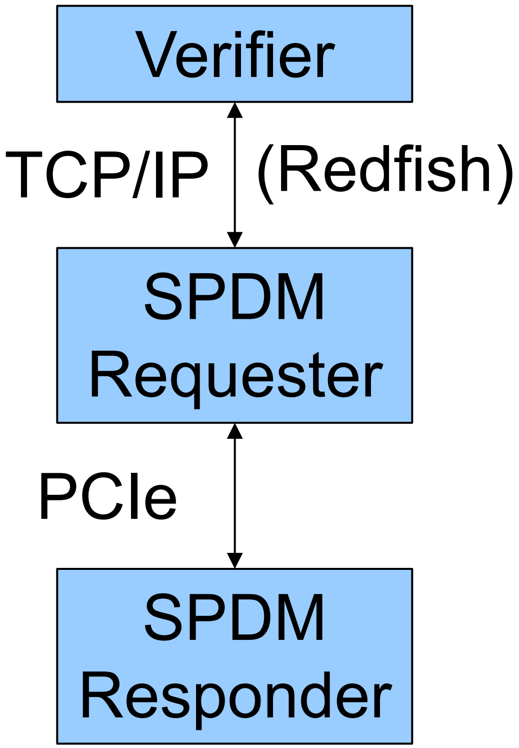
- The measurement transcript contains
  - GET_VERSION / VERSION
  - GET_CAPABILITIES / CAPABILITIES
  - NEGOTIATE_ALGORITHMS / ALGORITHMS
  - GET_MEASUREMENTS / MEASUREMENTS



```
L1/L2 = Concatenate(VCA, GET_MEASUREMENTS_REQUEST1, MEASUREMENTS_RESPONSE1, ...,
                    GET_MEASUREMENTS_REQUESTn-1, MEASUREMENTS_RESPONSEn-1,
                    GET_MEASUREMENTS_REQUESTn, MEASUREMENTS_RESPONSEn)
```

Simple Requester / Verifier / Responder Model

- Ultimately the Verifier’s domain of responsibility is assessing the security state of the Responder and the Responder’s target environment.
- Verifier is not particularly interested in SPDM traffic between Requester and Responder.
- Verifier may have to translate the DMTF measurement format to something more amenable for evaluation with a reference manifest.



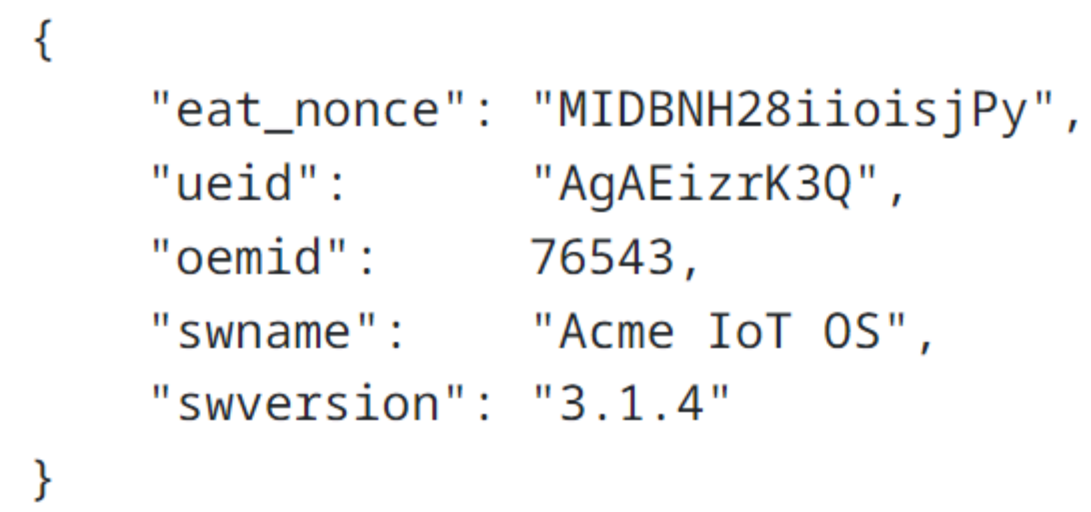
IETF Entity Attestation Token

- EAT is a draft RFC “that describes state and characteristics of an entity”.
- Developed as part of IETF RATS working group.
- Can be encoded in JSON and/or CBOR as a web token.
  - For SPDM devices assume CBOR.
- Can include composability and nesting of claims.
  - If a router has three NICs then three NIC sub-EATs can be embedded in the router EAT.
  - In contrast to the “flat” DMTF measurement specification.

IETF Entity Attestation Token

- Types of EAT claims include
  - Nonce
  - Identifiers
  - Device model information
  - Debug status
  - Location information
  - Uptime
  - Measurements
    - ‘The "measurements" claim contains descriptions, lists, evidence or measurements of the software that exists on the entity or any other measurable subsystem of the entity (e.g. hash of sections of a file system or non-volatile memory).’
    - Catch-all for device-specific claims that can be in various formats such as
      - Evidence CoSWID
      - CoRIM / CoMID

IETF Entity Attestation Token



```
{
    "eat_nonce": "MIDBNH28iioisjPy",
    "ueid":      "AgAEizrK3Q",
    "oemid":     76543,
    "swname":    "Acme IoT OS",
    "swversion": "3.1.4"
}
```

Example EAT encoded as JSON.

(Proposal) SPDM Support for EAT

Add a new bit to MeasurementSpecification.

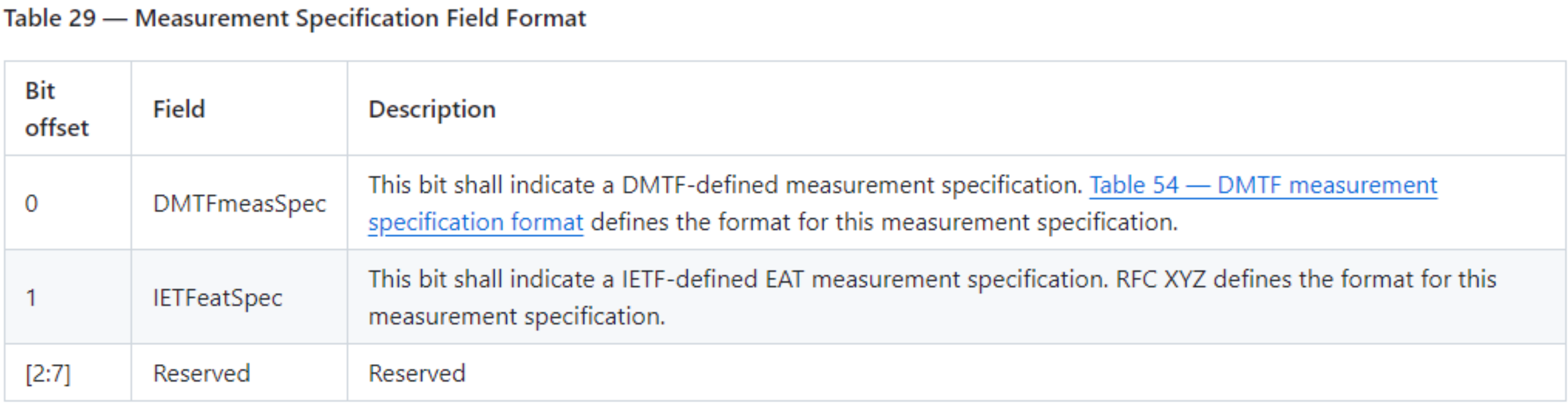


Table 29 — Measurement Specification Field Format

Bit offset	Field	Description
0	DMTFmeasSpec	This bit shall indicate a DMTF-defined measurement specification. Table 54 — DMTF measurement specification format defines the format for this measurement specification.
1	IETFeatSpec	This bit shall indicate a IETF-defined EAT measurement specification. RFC XYZ defines the format for this measurement specification.
[2:7]	Reserved	Reserved

Should the EAT always be CBOR or is JSON also allowed?

(Proposal) SPDM Support for EAT

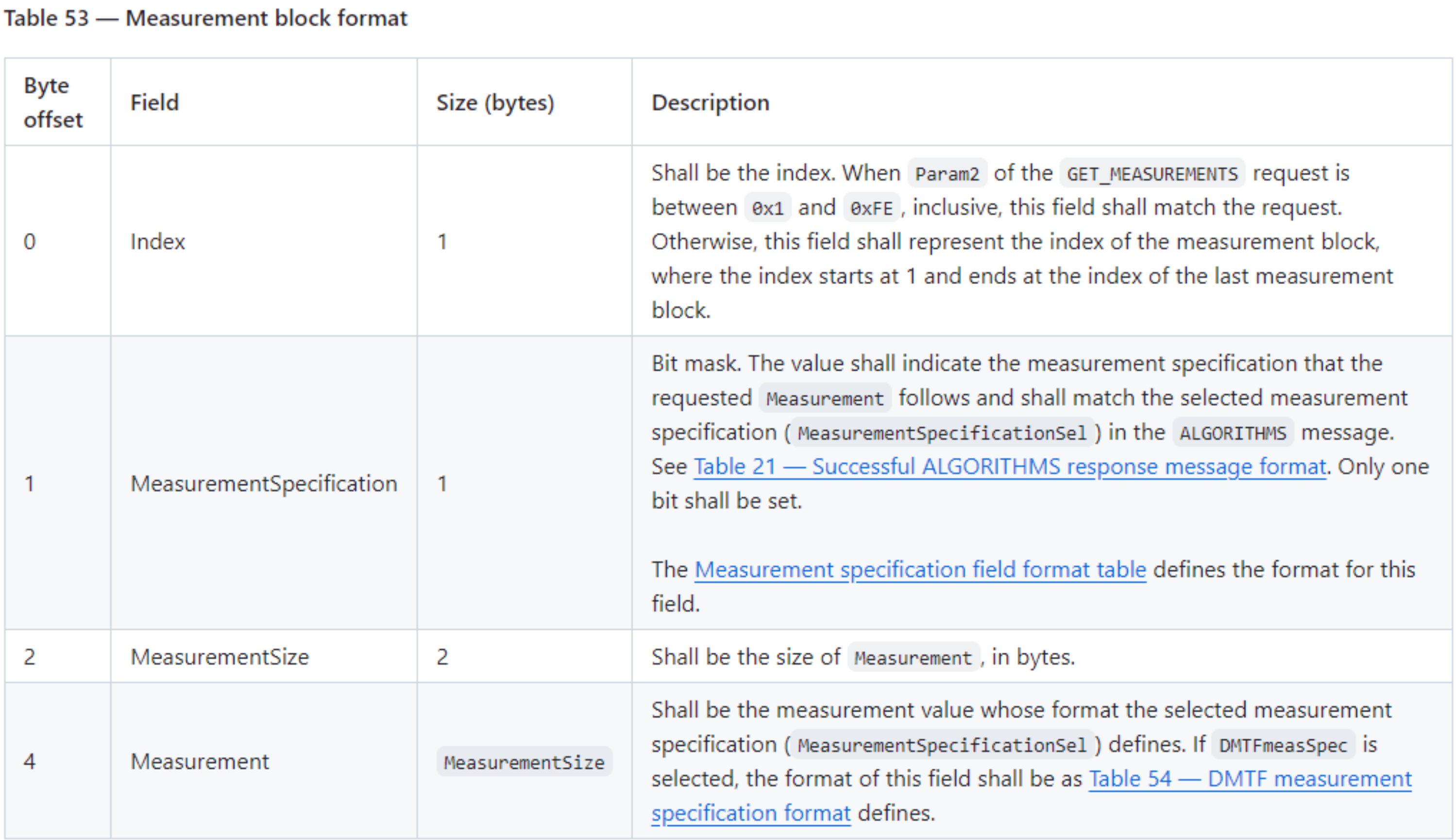


Table 53 — Measurement block format

Byte offset	Field	Size (bytes)	Description
0	Index	1	Shall be the index. When `Param2` of the `GET_MEASUREMENTS` request is between `0x1` and `0xFE`, inclusive, this field shall match the request. Otherwise, this field shall represent the index of the measurement block, where the index starts at 1 and ends at the index of the last measurement block.
1	MeasurementSpecification	1	Bit mask. The value shall indicate the measurement specification that the requested `Measurement` follows and shall match the selected measurement specification (`MeasurementSpecificationSel`) in the `ALGORITHMS` message. See Table 21 — Successful ALGORITHMS response message format. Only one bit shall be set. The Measurement specification field format table defines the format for this field.
2	MeasurementSize	2	Shall be the size of `Measurement`, in bytes.
4	Measurement	`MeasurementSize`	Shall be the measurement value whose format the selected measurement specification (`MeasurementSpecificationSel`) defines. If `DMTFmeasSpec` is selected, the format of this field shall be as Table 54 — DMTF measurement specification format defines.

EAT goes in the Measurement field.

EAT and SPDM Nonces

- The EAT Constrained Device Standard Profile requires a nonce and for the EAT to be signed.
- In SPDM a Requester only sends a nonce when the Responder is to sign the measurement transcript.
- Therefore if the measurement specification is EAT then Responder must support signing of measurements in its SPDM capabilities.
  - MEAS_CAP = 10b.
- EAT supports an array of nonces so both the Requester and Responder's nonce will appear in the EAT.

EAT and SPDM

- Entire EAT may be present in one measurement block or may be spread through multiple measurement blocks.
  - Maximum size of measurement block is 64 KiB.
- Besides requirements around the nonce, the EAT does not have to be consistent with SPDM capabilities or negotiated algorithms.
- Measurement extension / event log.
  - Considering forbidding GET_MEASUREMENT_EXTENSION_LOG if EAT is chosen.
  - The EAT itself may include the measurement extension log.

Feedback

- Please review the proposal and provide feedback.
  - Attend the SPDM Working Group if already a DMTF member.
  - Can also provide feedback via https://www.dmtf.org/standards/feedback.