COSE Hash Envelope

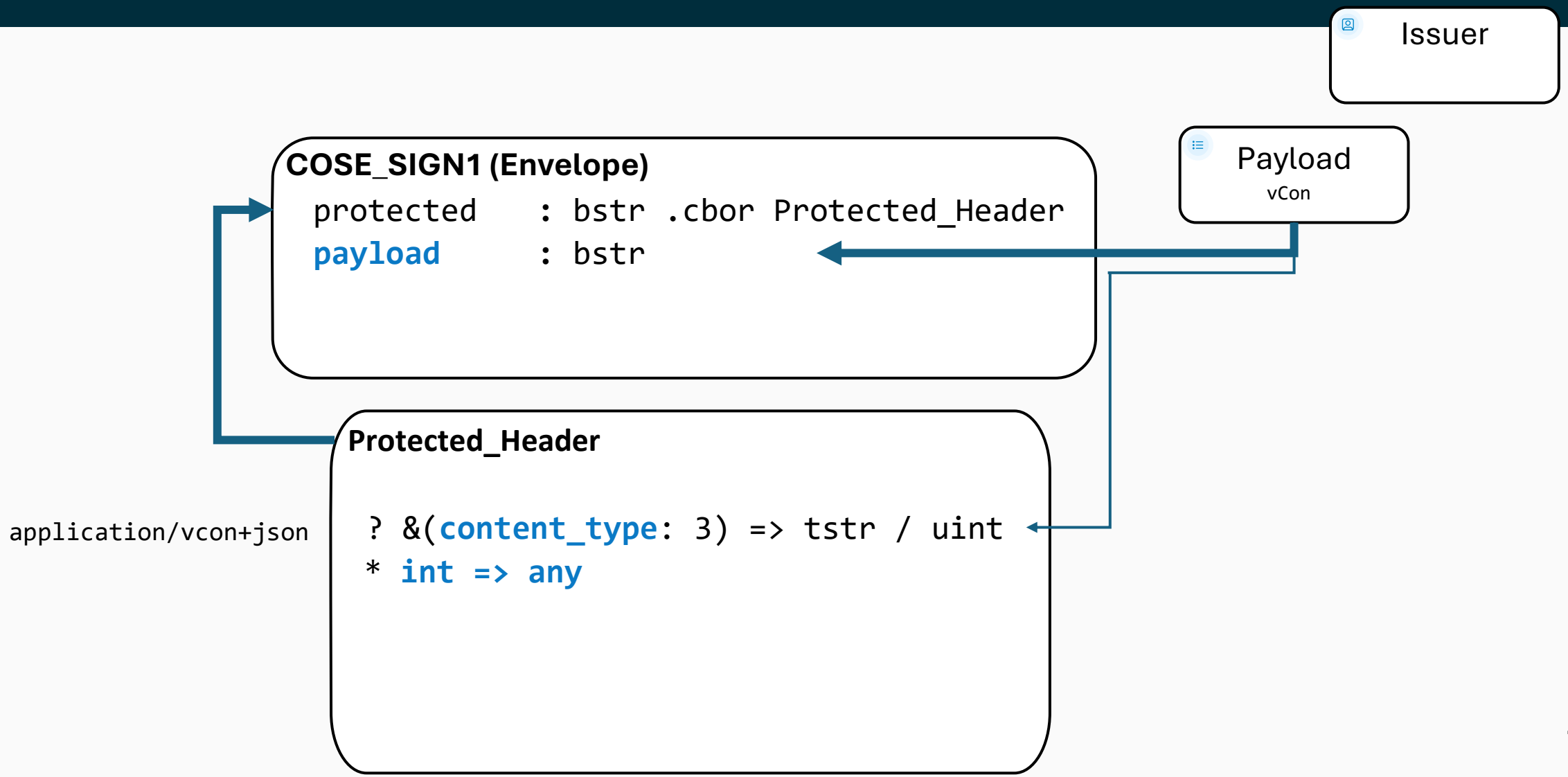
https://datatracker.ietf.org/doc/draft-ietf-cose-hash-envelope/

COSE Hash Envelope: Overview

- Defines COSE header parameters for signaling a payload as an output of a hash function.
- Enables faster validation as access to the original payload is not required for signature validation.
- Hints of the detached payload's content format and location

Current State - Attached

To Be Signed Bytes



Signed Bytes

COSE_SIGN1 (Envelope)

```
protected   : bstr .cbor Protected_Header
payload     : bstr
signature   : bstr
```

Issuer

Payload
vCon

Protected_Header

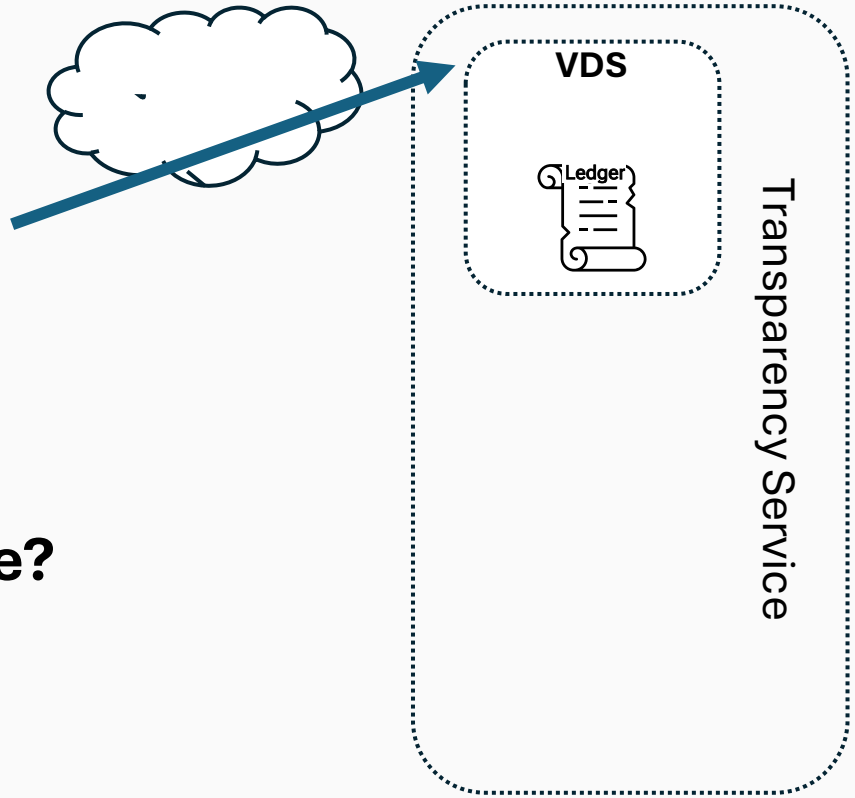
```
? &(content_type: 3) => tstr / uint
* int => any
```

application/vcon+json

Registering On a Verifiable Data Structure (VDS)

COSE_SIGN1 (Envelope)

```
protected   : bstr .cbor Protected_Header
payload     : bstr
signature   : bstr
unprotected : Unprotected_Header
```

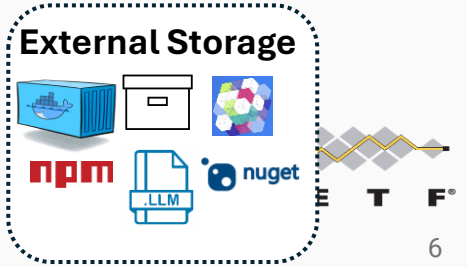


How large is the COSE_Sign1 Envelope?

Protected Header	~1k
Unprotected Header	0
Signature	~1k

2k

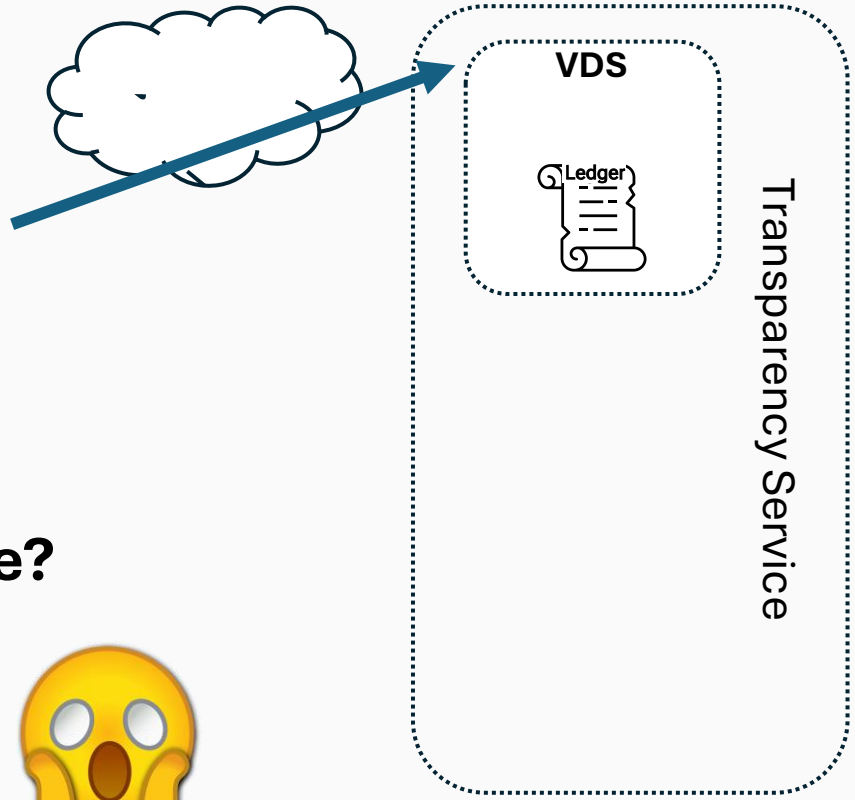
- Is ***size*** the constraint?
- Is the ***payload*** already stored somewhere else?
- Do we need to continually pass the payload for signature checking?



Registering On a Verifiable Data Structure (VDS)

COSE_SIGN1 (Envelope)

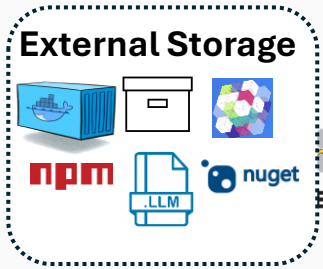
```
protected   : bstr .cbor Protected_Header
payload     : bstr
signature   : bstr
unprotected : Unprotected_Header
```



How large is the COSE_Sign1 Envelope?

Protected Header	~1k
Unprotected Header	0
Signature	~1k
Payload	1k-50gb

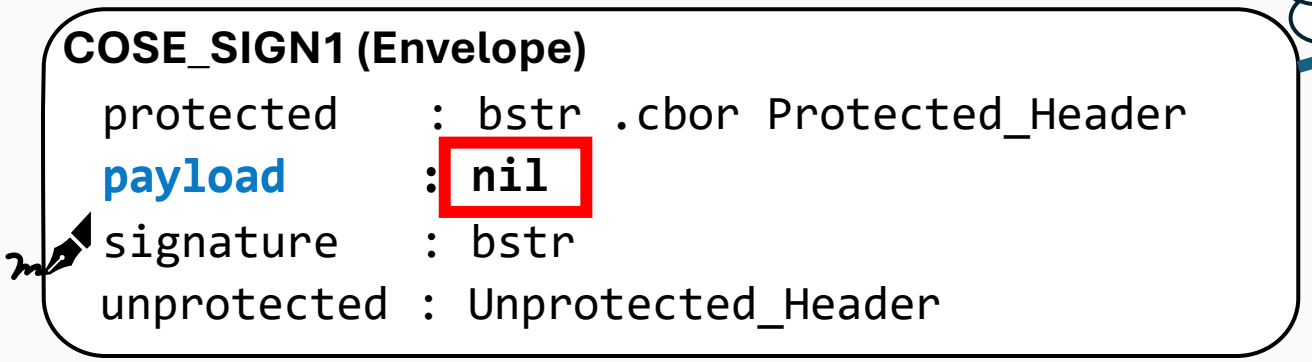
~50.002gb

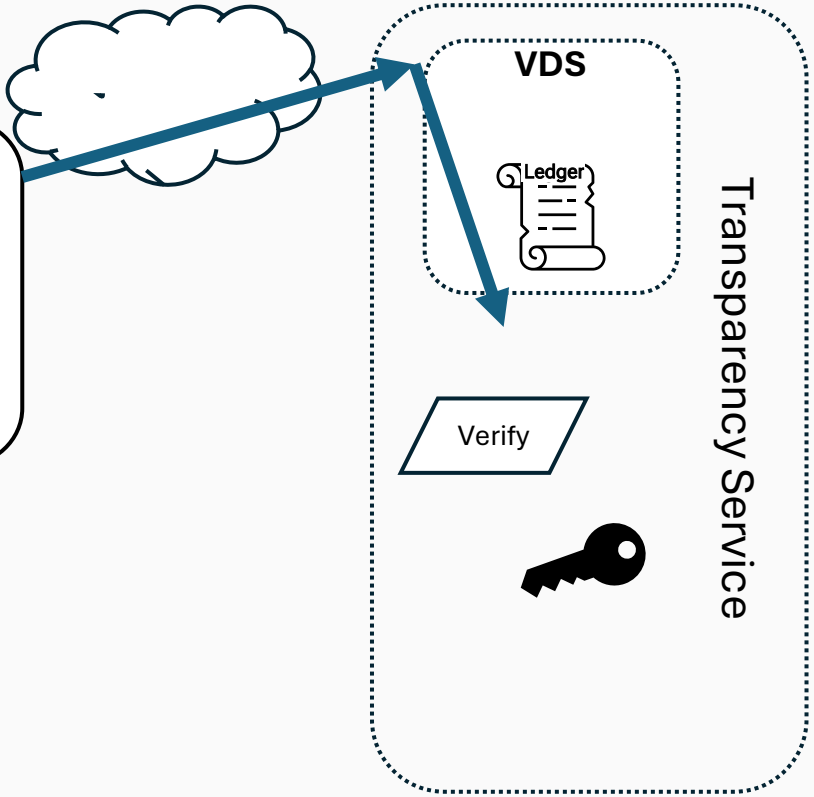


- Is ***size*** the constraint?
- Is the ***payload*** already stored somewhere else?
- Do we need to continually pass the payload for signature checking?

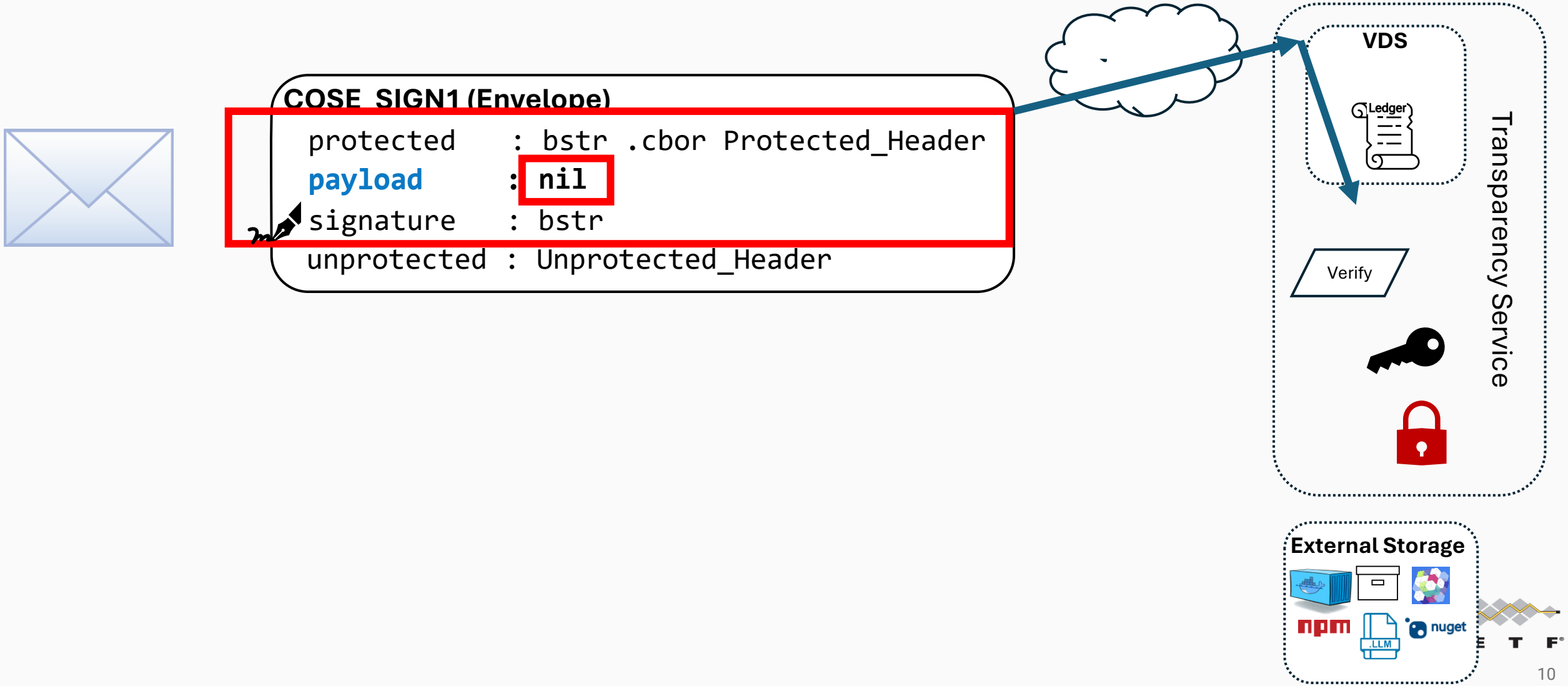
COSE Detached Payloads

Detached Payloads

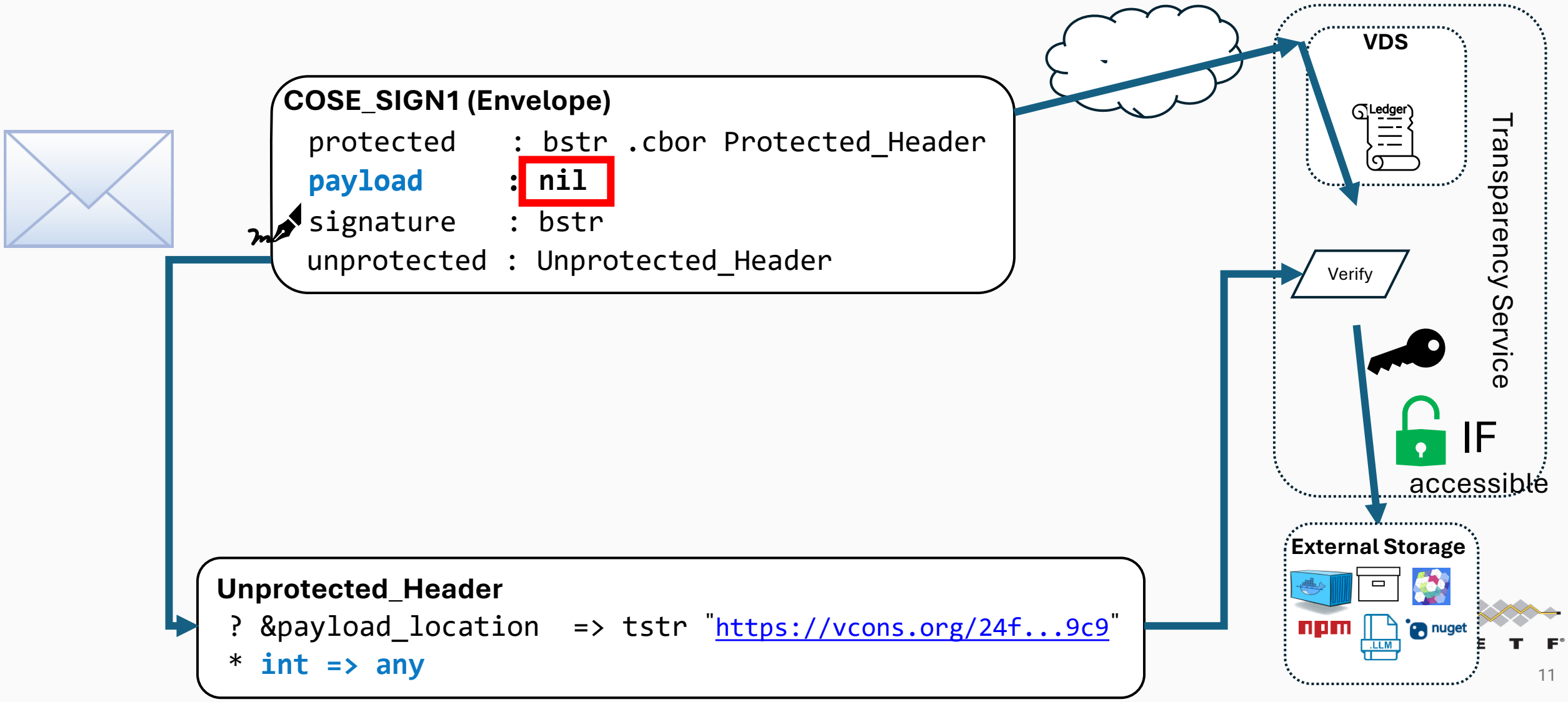




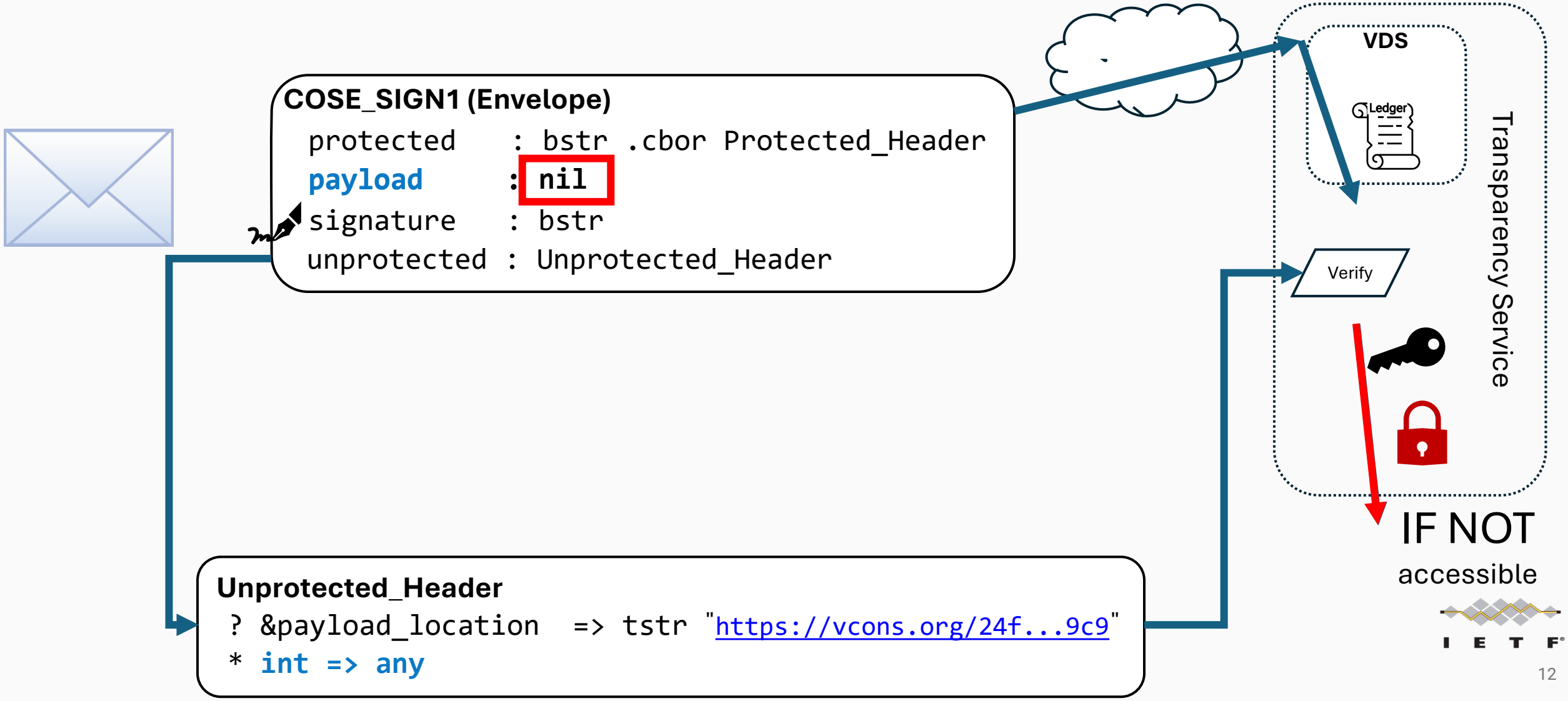
Detached Payloads



Detached Payloads



Detached Payloads

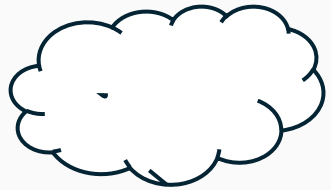


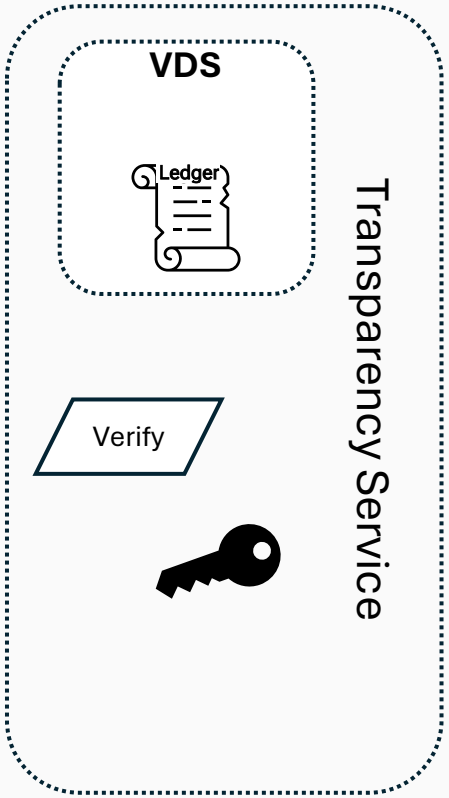
COSE Hash Envelope

Hashed Payloads

COSE_SIGN1 (Envelope)

```
protected   : bstr .cbor Protected_Header
payload     : h'935b5a91...e18a588a'
signature   : bstr
unprotected : Unprotected_Header
```





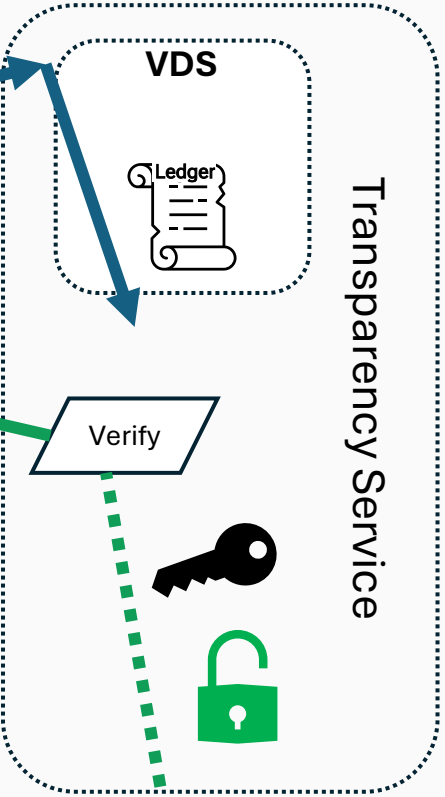
Protected_Header

```
? (typ: 16) : “application/hashed+cose”
? (payload_hash_alg : int) : -16 / SHA-256 /
? (content_type : tstr / int) : "application/vcon+json"
? (payload_preimage_content_type : tstr / int) : "application/vcon+json"
? (payload_location : tstr ) : "https://vcons.org/24f...9c9"
```

Hashed Payloads

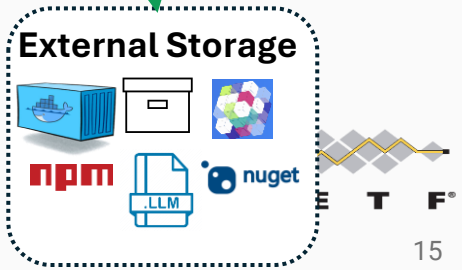
COSE_SIGN1 (Envelope)

```
protected   : bstr .cbor Protected_Header
payload     : h'935b5a91...e18a588a'
signature   : bstr
unprotected : Unprotected_Header
```



Protected_Header

```
? (typ: 16) : “application/hashed+cose”
? (payload_hash_alg : int) : -16 / SHA-256 /
? (payload_preimage_content_type : tstr / int) : "application/vcon+json"
? (payload_location : tstr ) : "https://vcons.org/24f...9c9"
```



HASH Envelope Protected Header

```
18(                                   / COSE Sign 1                          /
   [
    <<{                                   / Protected
      1:-35,                              / alg : ES384                      /
      4: h'75726e3a...32636573',          / kid                              /
      16: "application/example+cose",     / typ                              /
      TBD_1: -16                          / payload_hash_alg : sha-256       /
      TBD_2: "application/vcon+json"      / payload_preimage_content_type    /
      TBD_3: "https://vcons.org/24f...9c9"  / payload_location               /
    }>>
    {}                                / Unprotected                          /
    h'935b5a91...e18a588a',           / Payload (SHA-256 Hash of the vCon )  /
    h'15280897...93ef39e5'            / Signature                            /
   ]
)
```

Fetching the payload can be negotiated by the content type (**TBD_2**)

Example: vCon which may be GB in size, including transcriptions)

Incorporate Feedback Since IETF 120

- Use COSE_Hash_V #9 Closed
  - adoption of RFC 9596 by the COSE WG
  - #18 Label (content_type : 3) MUST NOT be used as it is easily confused with label (payload_preimage_content_type : TBD_2)
- Address COSE_Encrypt #16
  Remove COSE Encrypt Section #17
- -00 feedback from ilari #22
  - Pre 02 #27 – reference use of RFC8032 & FIPS-204
  - Explain the relationship between crypto layer pre-hash and protocol layer pre-hash #26

Reference Validations & Implementations

- vCon
  - github.com/vcon-dev/vcon-server/links/scitt
  - docs.datatrails.ai/developers/templates/scitt

- SCITT GitHub Actions
  - github.com/datatrails/scitt-action
  - github.com/digicert/scitt-action

COSE Hash Envelope Headers – Request for Pre-Allocation

3. Header Parameters

To represent a hash of a payload, the following headers are defined:

- **`payload_hash_alg: TBD_1`**
  the hash algorithm used to produce the payload.
- **`payload_preimage_content_type: TBD_2`**
  the content type of the bytes that were hashed to produce the payload.
- **`payload_location: TBD_3`**
  an identifier enabling a verifier to retrieve the bytes which were hashed to produce the payload.

https://datatracker.ietf.org/doc/draft-ietf-cose-hash-envelope/

Thank You