RESTful Attested Resources

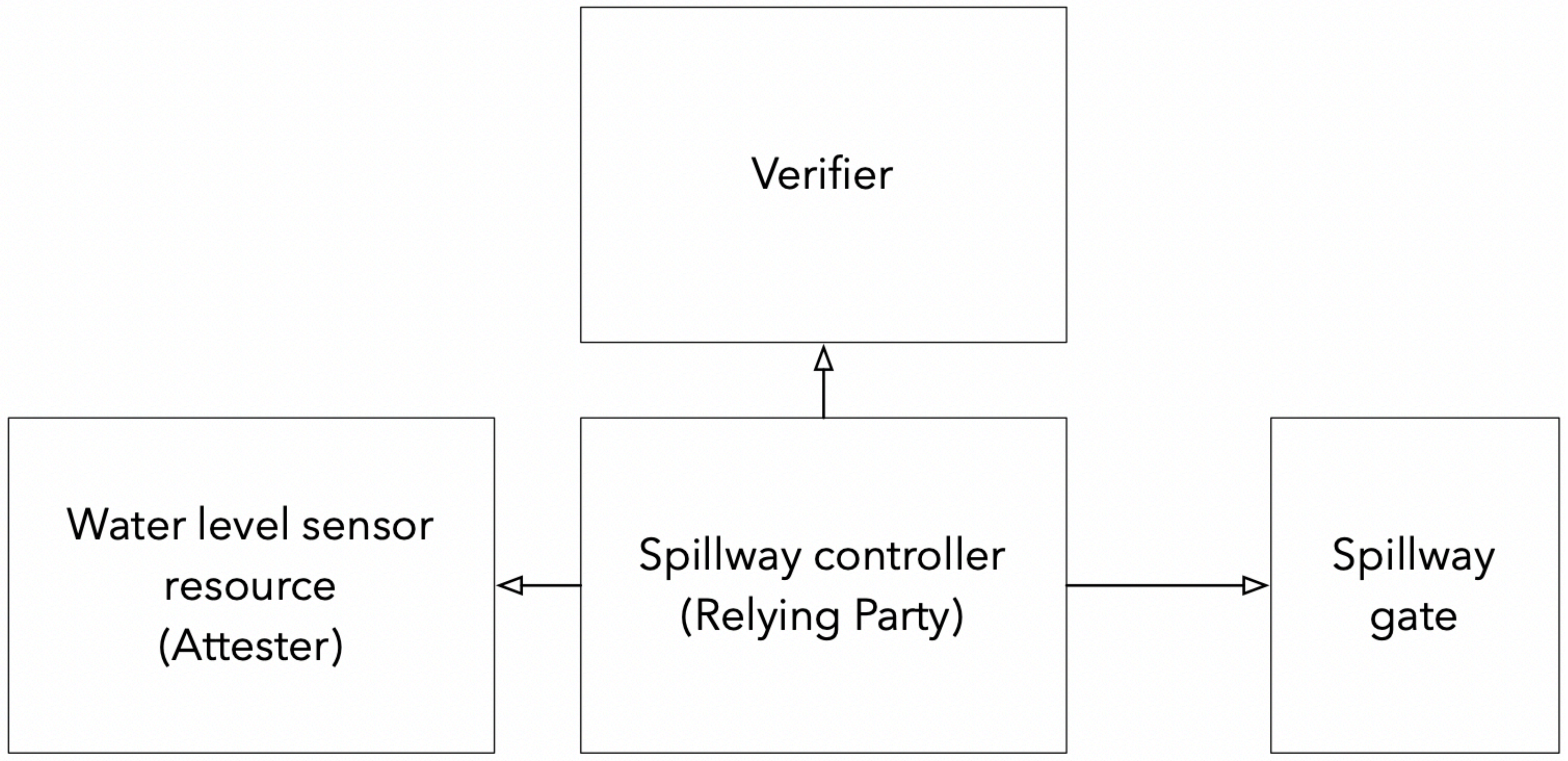
`draft-shaw-rats-rear-00`

Goal

- Present the main ideas in draft-shaw-rats-rear that might be useful to others when looking at how to instantiate the RATS architecture using the IETF toolbox

Use cases

- Critical infrastructure systems, IIoT
- E.g., dam
  - Objective: control inflow/outflow balance by regulating the spillway / overflow channel
  - Water level sensor (Attester)
  - Dam's spillway gate controller (RP)
    - Needs to trust the water level sensor (verify evidence)


Verifier
Water level sensor resource (Attester)
Spillway controller (Relying Party)
Spillway gate

Attested Resource

- All in one :
  - Resource representation
  - Evidence about the hosting platform security state
  - Freshness indicator

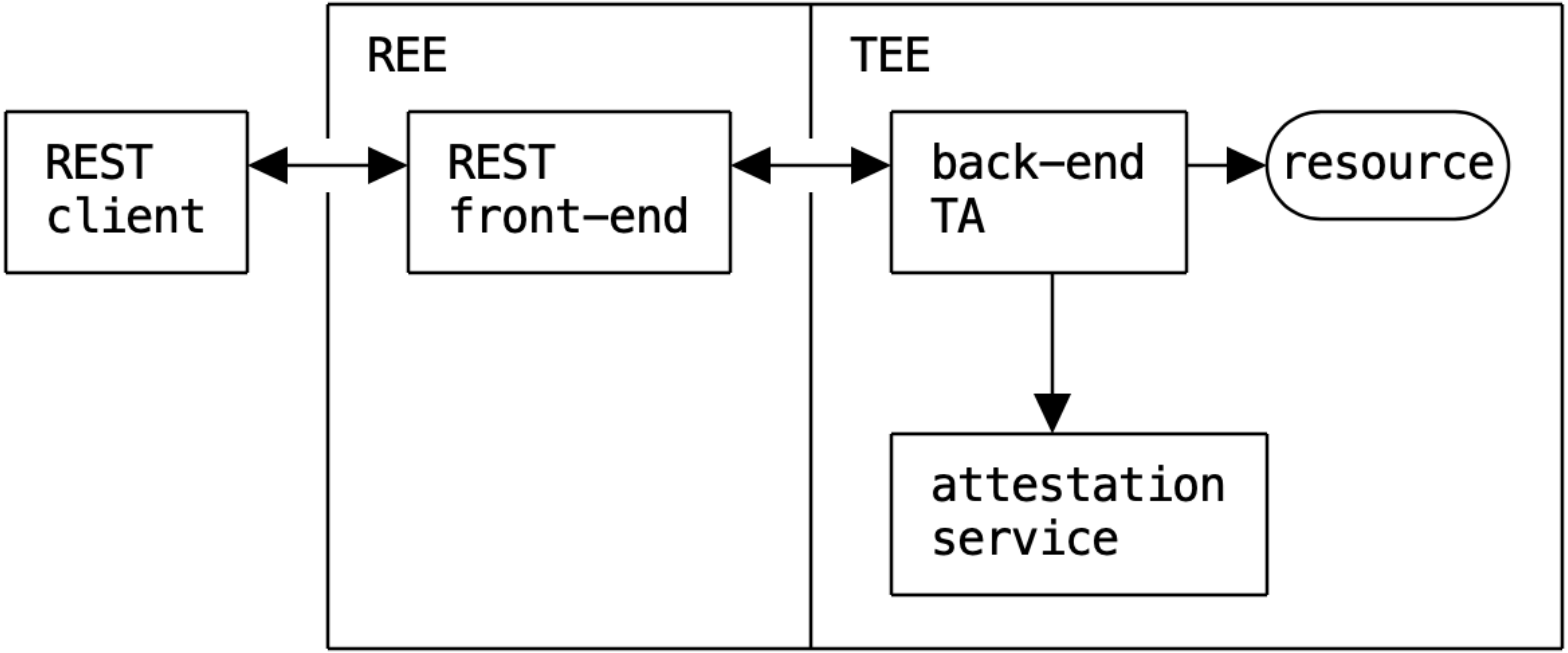
Attested Resource (cont.)

- For this we need to define:
  - Interaction model
  - A compound data format – to pull together resource state, attestation data and freshness
  - A combining function that mixes the inputs, which cannot be subverted by an active adversary

(RESTful) Attested resources

- *Basic* RESTful interface to access attested resources:
  - Request methods, response status codes, MIME types
    - CoAP & HTTP
  - Optional discovery interface based on the CoRE RD
- RESTful interface to the Verifier

Implementation model

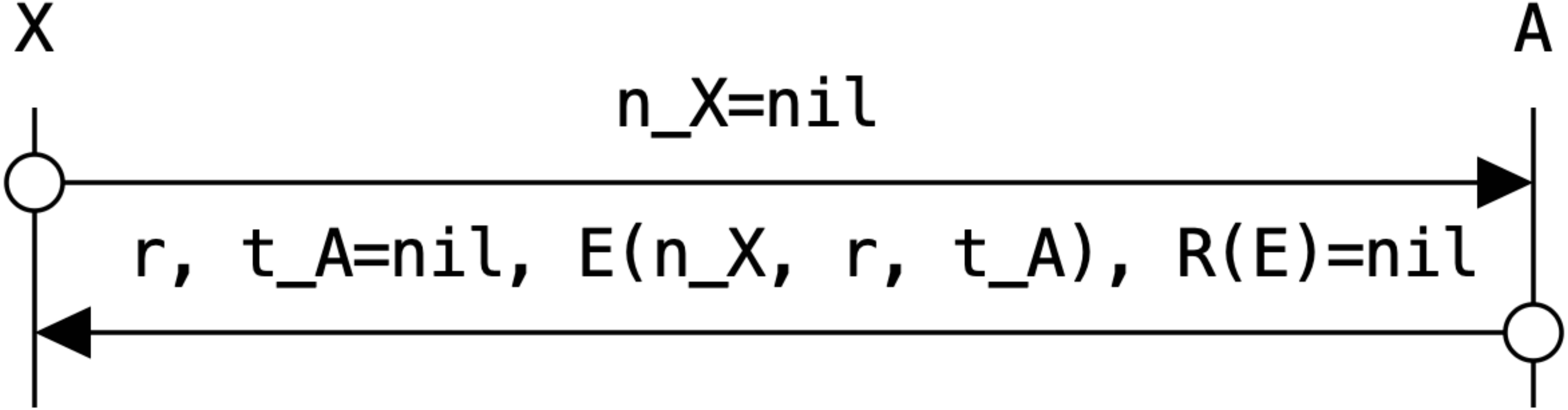


Mixing function

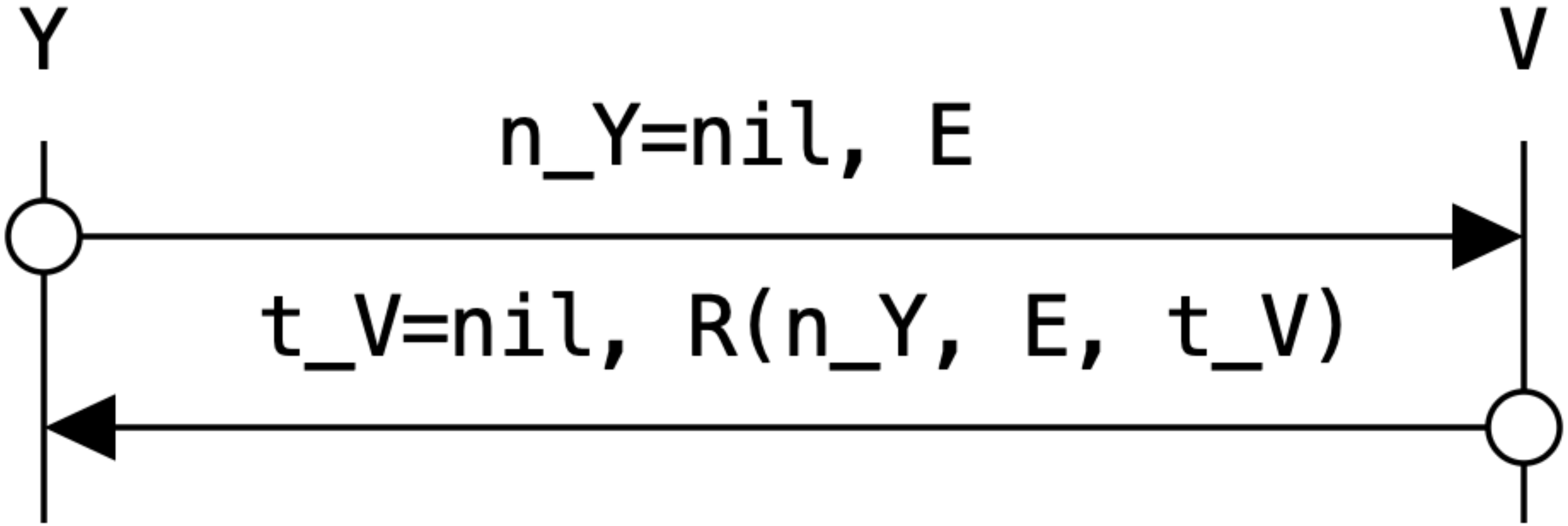
The security guarantee provided by the mixing functions is that the resource and platform security state cannot be separated without breaking the verification process

- `n`, optional nonce provided by the initiator
    - If `n==nil` => `$n = “”`
- `t`, optional timestamp provided by responder
    - If t`==nil` => `$t = “”`
- `r`, to-be-attested resource
- `Mix(n, r, t) = H($n || $r || $t) –> E.nonce`

Attester interface

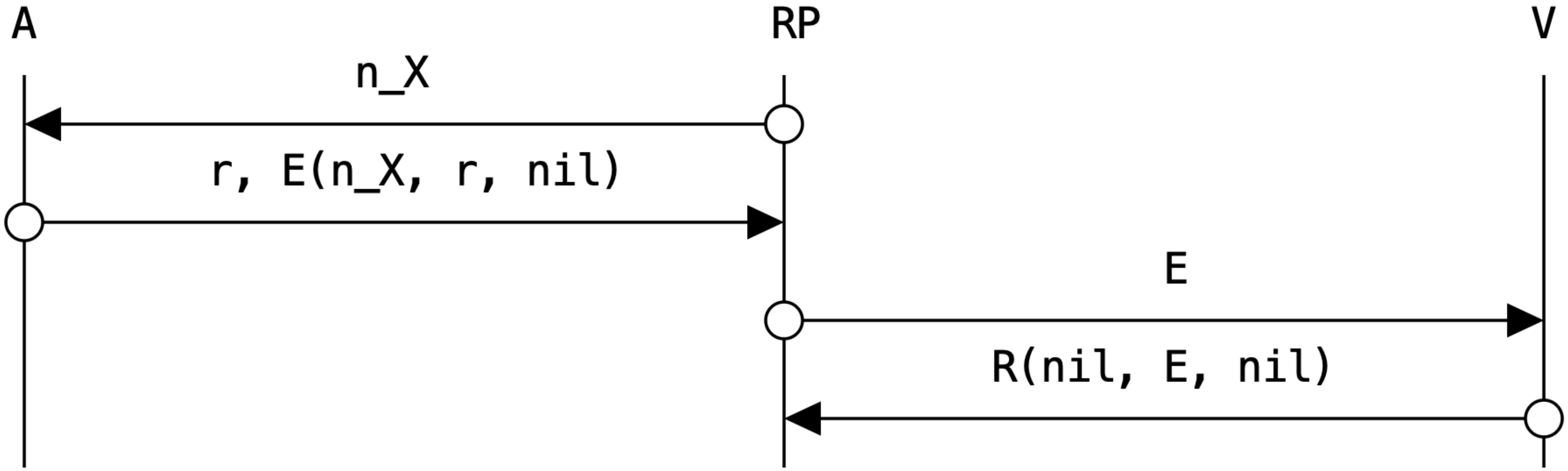


Verifier interface



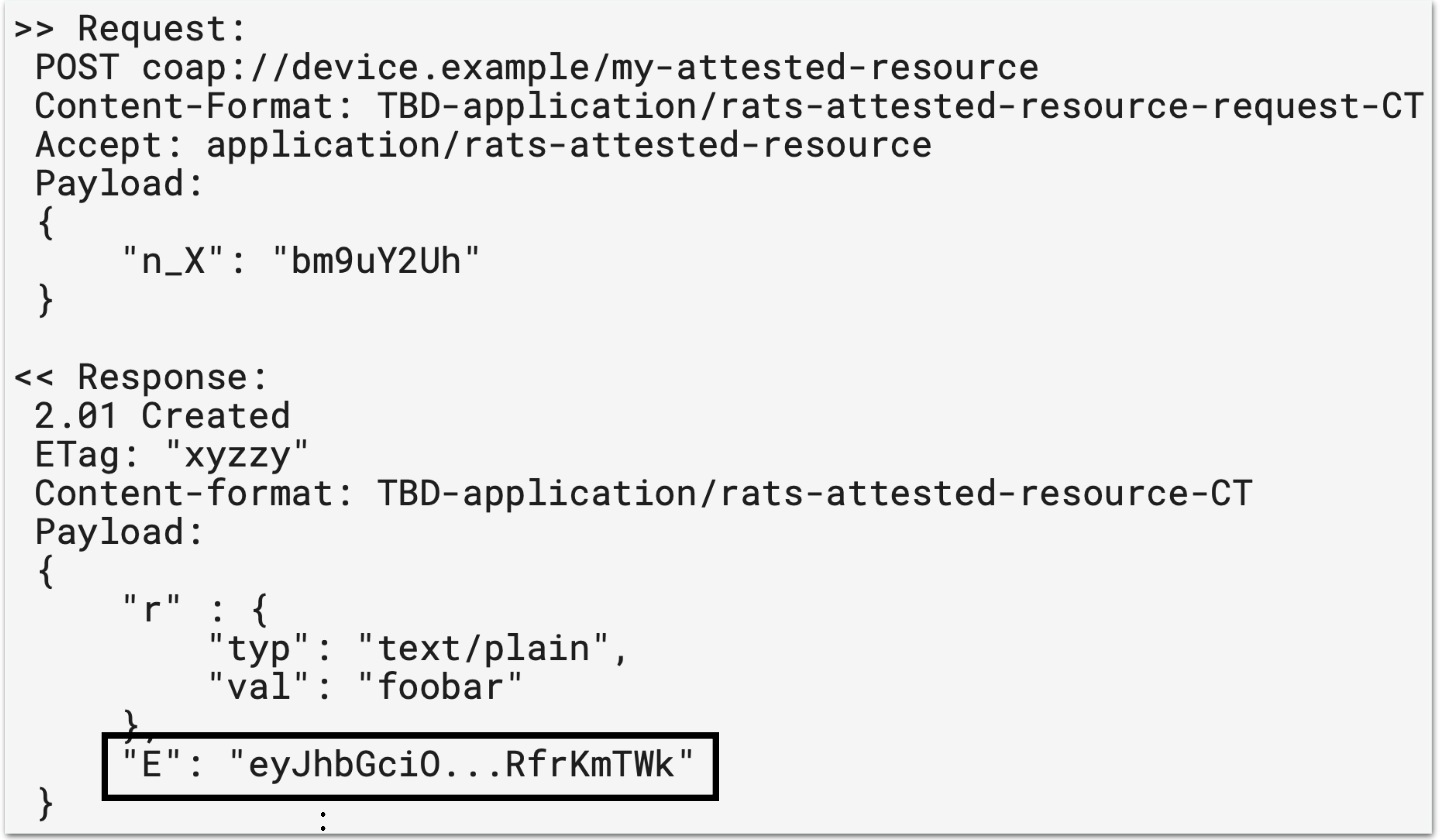
Compositions

Background Check with Nonce-based Freshness (abstract)



Background Check with Nonce-based Freshness (practical)

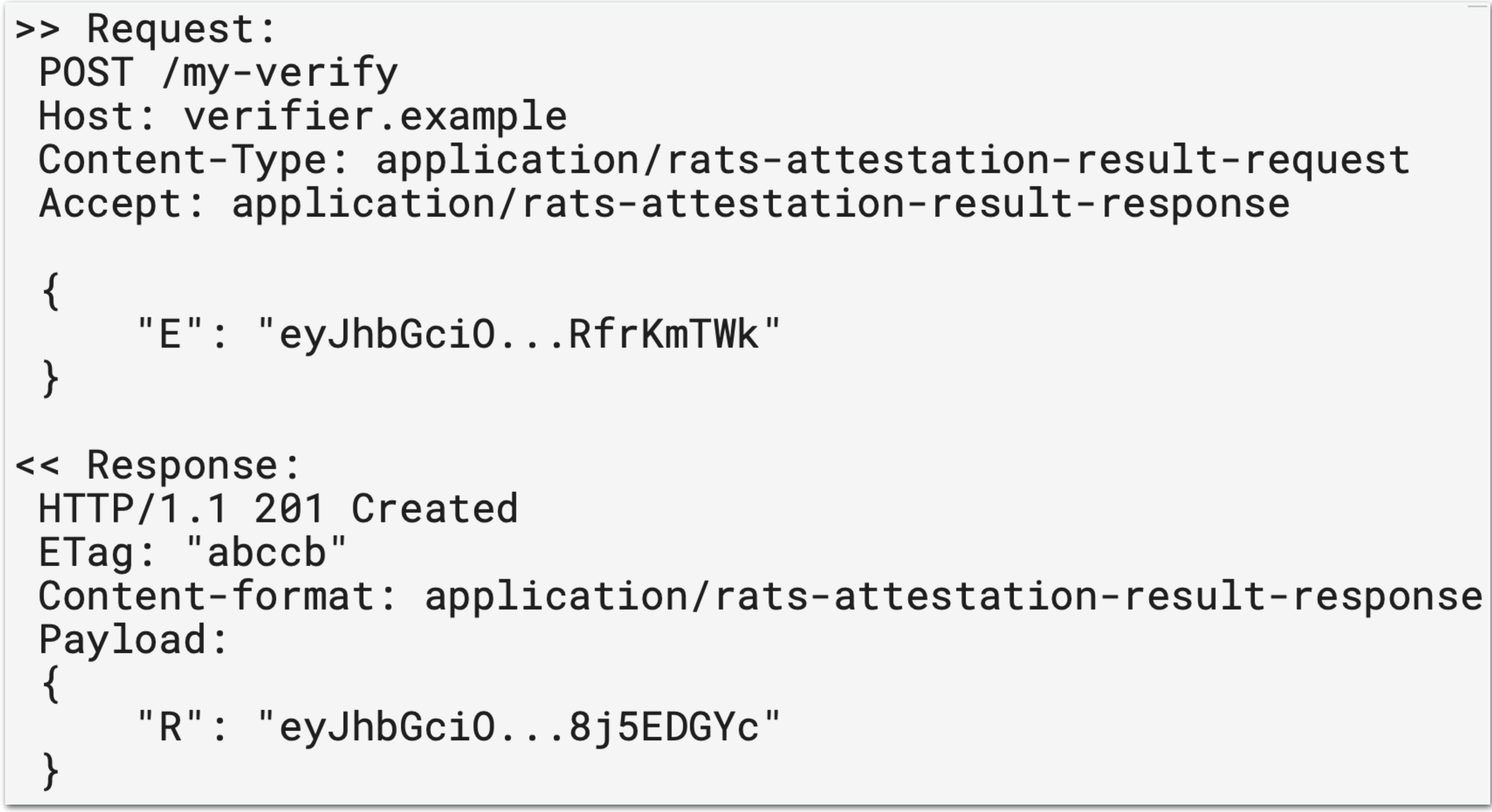
Relying Party <—> Attester (using CoAP)



```
>> Request:
 POST coap://device.example/my-attested-resource
 Content-Format: TBD-application/rats-attested-resource-request-CT
 Accept: application/rats-attested-resource
 Payload:
 {
     "n_X": "bm9uY2Uh"
 }

<< Response:
 2.01 Created
 ETag: "xyzzy"
 Content-format: TBD-application/rats-attested-resource-CT
 Payload:
 {
     "r" : {
         "typ": "text/plain",
         "val": "foobar"
     },
     "E": "eyJhbGciO...RfrKmTWk"
 }
```

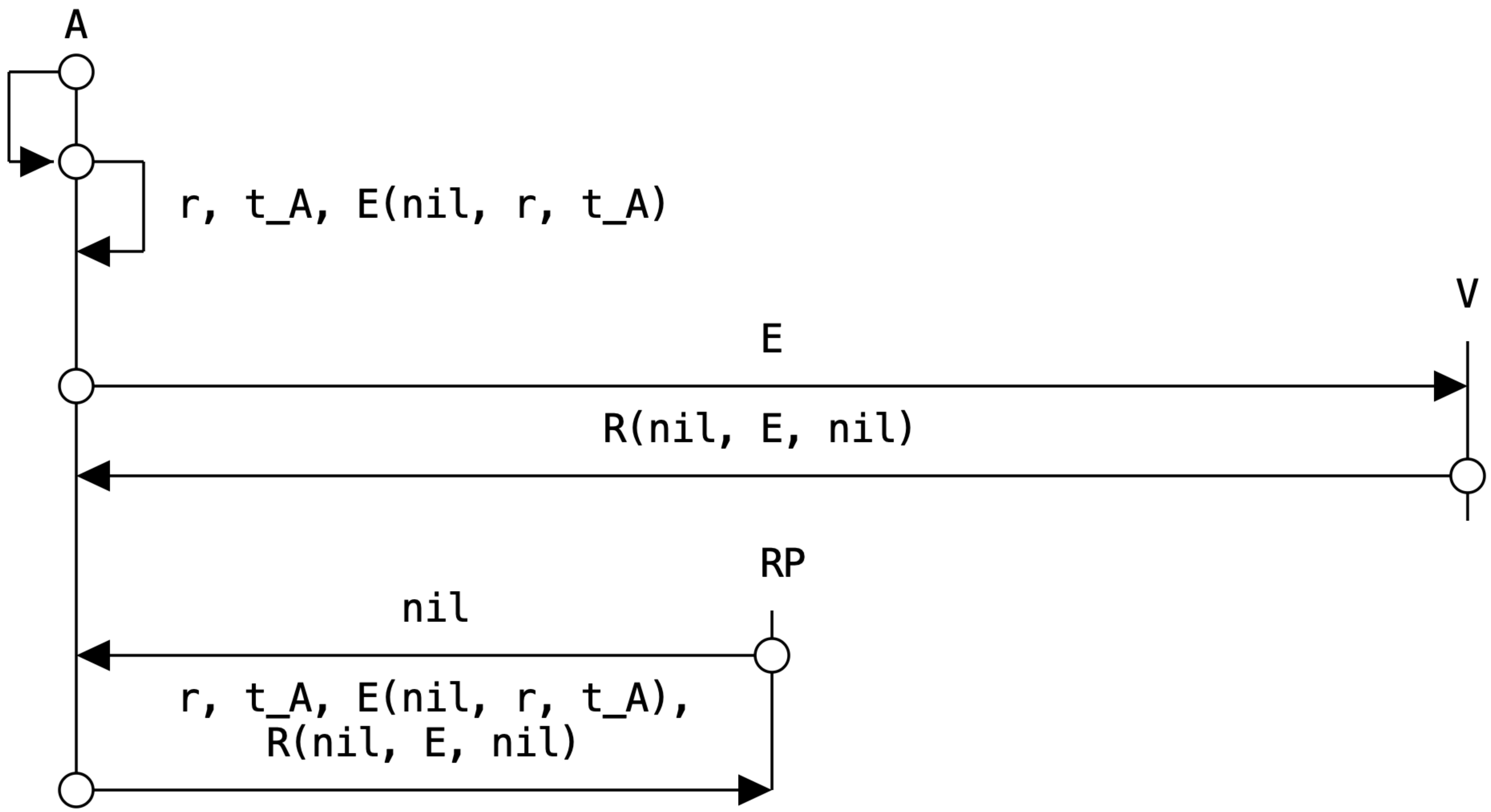
```
E.nonce = H(“bm9uY2Uh” || “{… “foobar” }” || “”))
```

Relying Party <—> Verifier (using HTTP)



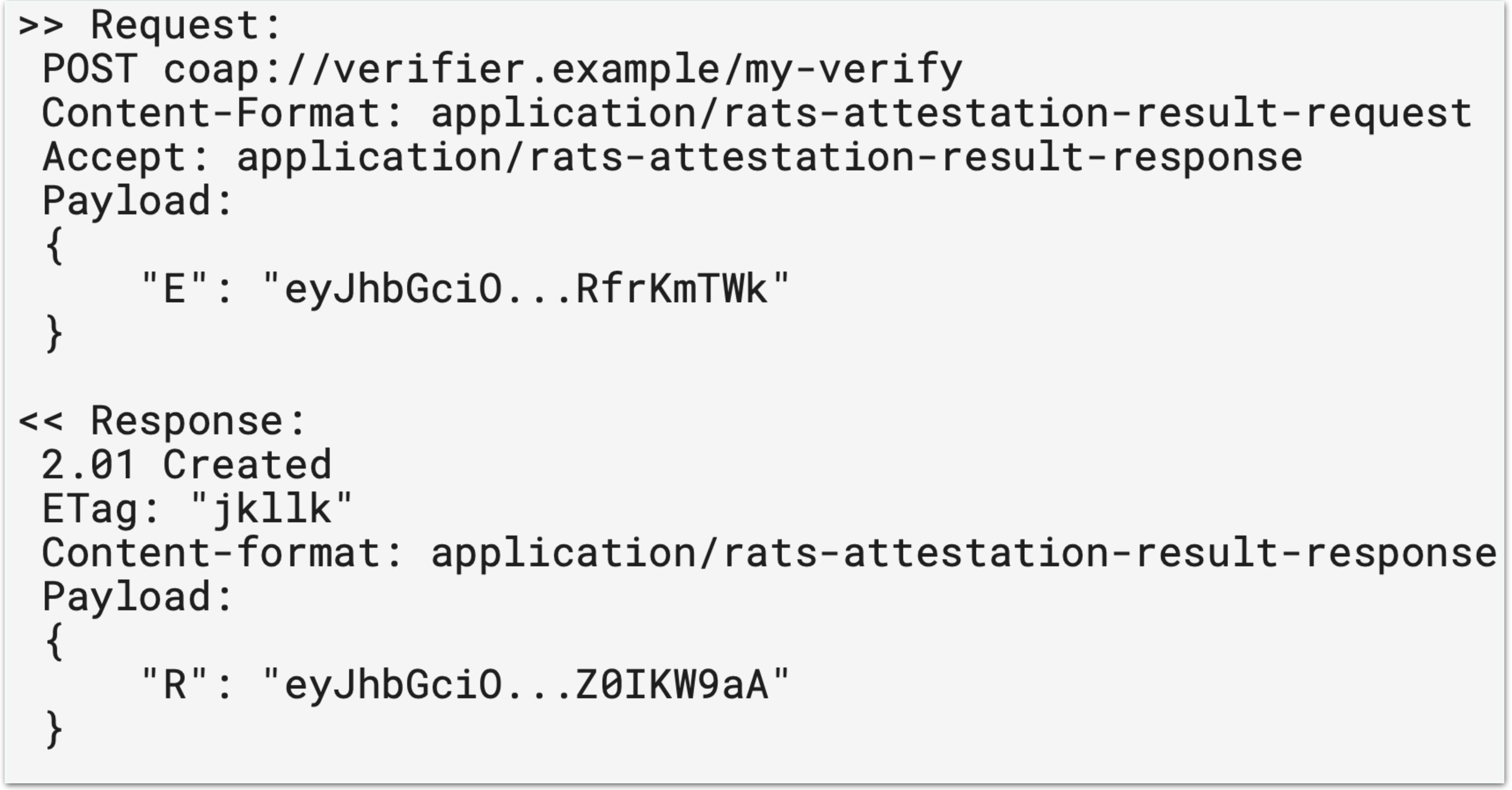
```
>> Request:
 POST /my-verify
 Host: verifier.example
 Content-Type: application/rats-attestation-result-request
 Accept: application/rats-attestation-result-response

 {
     "E": "eyJhbGciO...RfrKmTWk"
 }

<< Response:
 HTTP/1.1 201 Created
 ETag: "abccb"
 Content-format: application/rats-attestation-result-response
 Payload:
 {
     "R": "eyJhbGciO...8j5EDGYc"
 }
```

Passport with Timestamp-based Freshness (abstract)



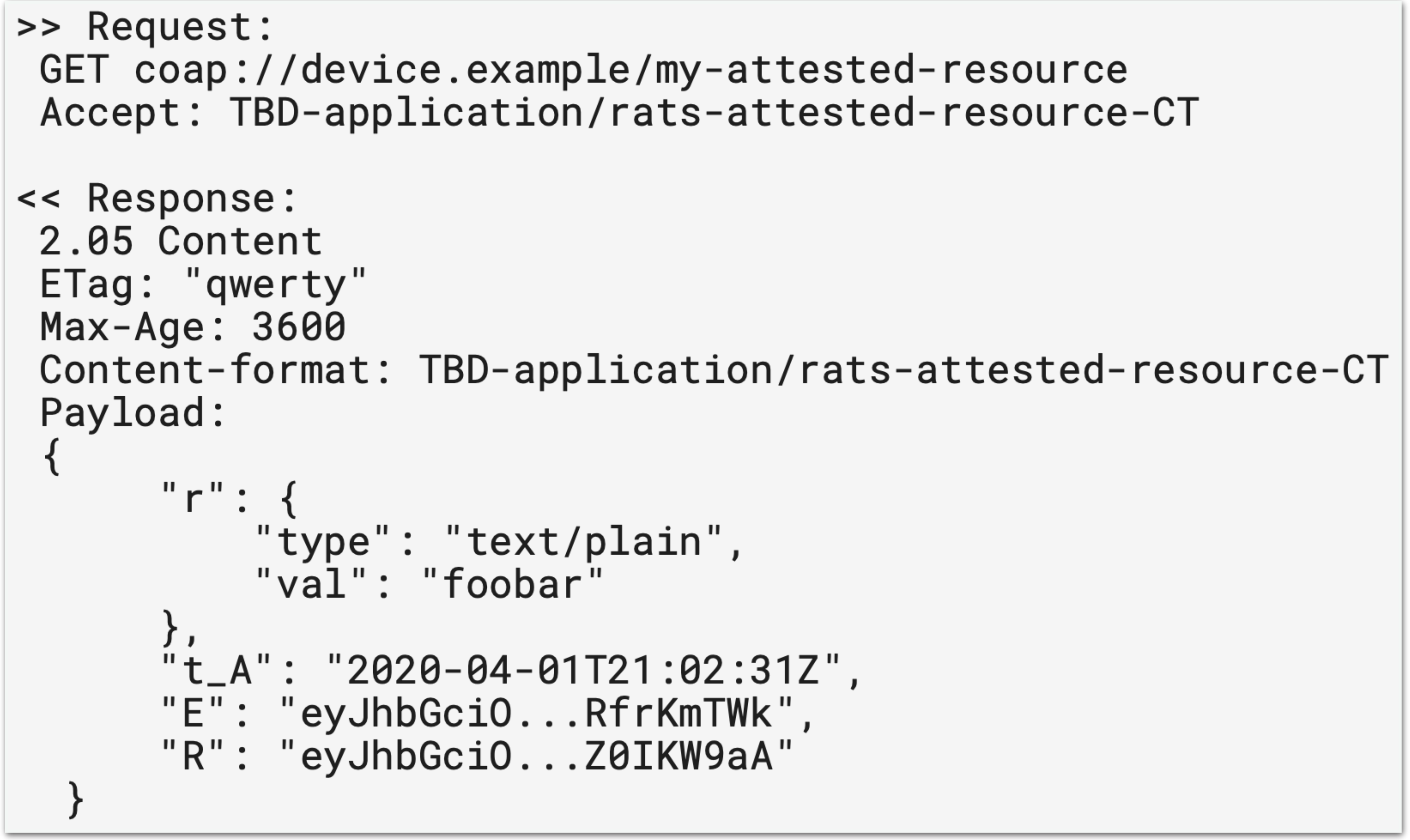
Passport with Timestamp-based Freshness (practical)

Attester <—> Verifier (using CoAP)



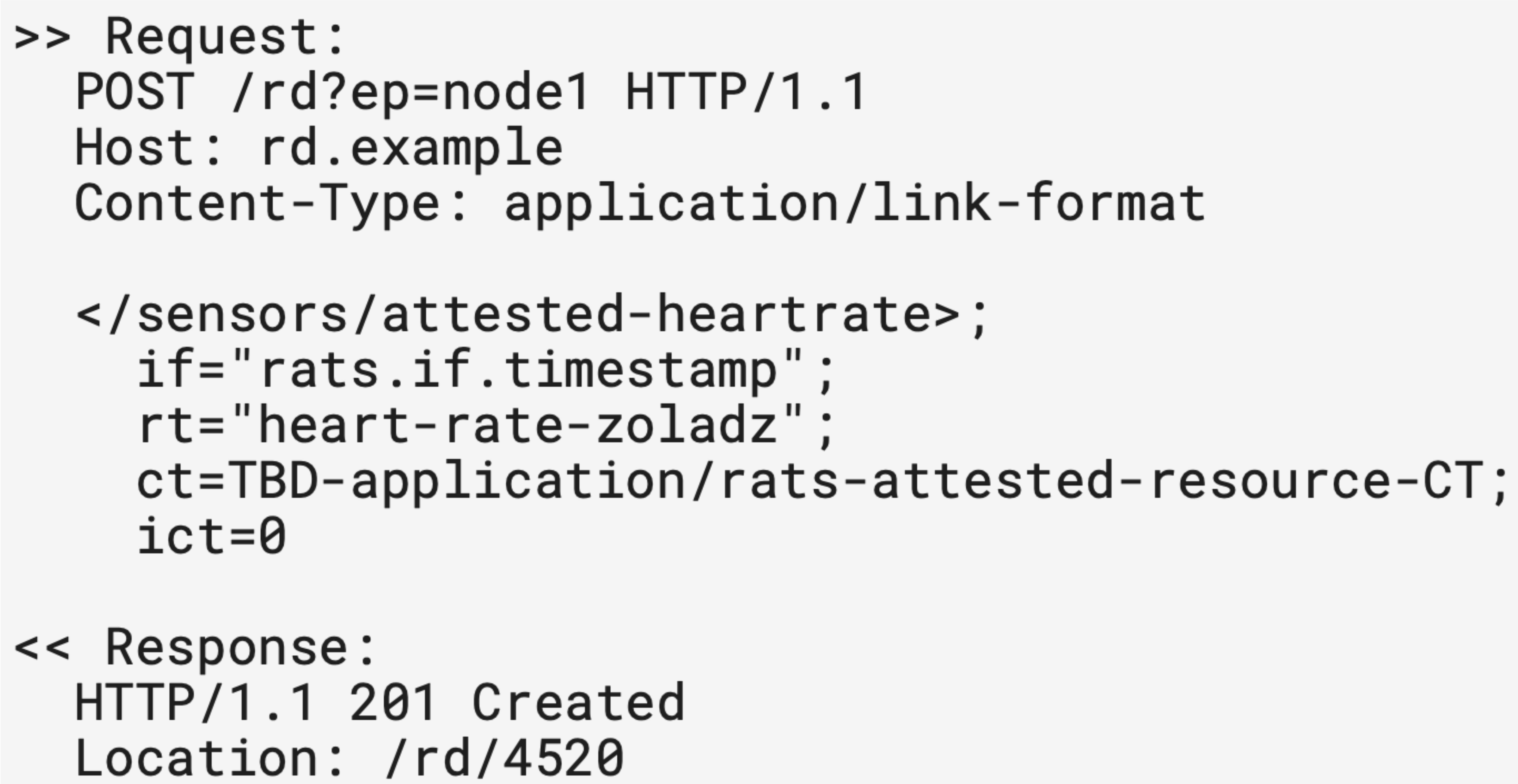
```
>> Request:
 POST coap://verifier.example/my-verify
 Content-Format: application/rats-attestation-result-request
 Accept: application/rats-attestation-result-response
 Payload:
 {
     "E": "eyJhbGciO...RfrKmTWk"
 }

<< Response:
 2.01 Created
 ETag: "jkllk"
 Content-format: application/rats-attestation-result-response
 Payload:
 {
     "R": "eyJhbGciO...Z0IKW9aA"
 }
```

Relying Party <—> Attester (using CoAP)



```
>> Request:
 GET coap://device.example/my-attested-resource
 Accept: TBD-application/rats-attested-resource-CT

<< Response:
 2.05 Content
 ETag: "qwerty"
 Max-Age: 3600
 Content-format: TBD-application/rats-attested-resource-CT
 Payload:
 {
      "r": {
          "type": "text/plain",
          "val": "foobar"
      },
      "t_A": "2020-04-01T21:02:31Z",
      "E": "eyJhbGciO...RfrKmTWk",
      "R": "eyJhbGciO...Z0IKW9aA"
  }
```

Discovery



```
>> Request:
  POST /rd?ep=node1 HTTP/1.1
  Host: rd.example
  Content-Type: application/link-format

  </sensors/attested-heartrate>;
    if="rats.if.timestamp";
    rt="heart-rate-zoladz";
    ct=TBD-application/rats-attested-resource-CT;
    ict=0

<< Response:
  HTTP/1.1 201 Created
  Location: /rd/4520
```

Discuss

- Is there any appetite for a *generic* substrate (either rats-rear or something *similar*)?
- Should we go forward?

ACKs

- Henk Birkholz
- Michael Richardson
- Kathleen Moriarty

Questions?