CoSERV Veraison Prototype

Prototype API

Simple request-response over HTTP using `GET`

CoSERV quert is in the outer-most path segment (allow caching)

Parameterised media type (`profile` is a CoRIM profile)

Request

```
GET /endorsement-distribution/v1/coserv/ogB4I3R... HTTP/1.1
Host: localhost:11443
Accept: application/coserv+cbor; \
        profile="tag:arm.com,2023:cca_platform#1.0.0"
```

Response

```
HTTP/1.1 200
Content-Type: application/coserv+cbor
Content-Length: 702
Date: Tue, 15 Apr 2025 10:37:28 GMT

{
  / profile / 0: "tag:arm.com,2023:cca_platform#1.0.0",
  / query / 1: {
    / artifact / 0: 2, / reference-values /
    / environment-selector / 1: {
      / class / 0: [
        { / class-id / 0: 600(h'7f45...') / impl-id / }
      ]
    }
  },
  / results / 2: { ...  }
}
```

New coserv-service

```
const (
  edApiPath = "/endorsement-distribution/v1"
)

func NewRouter(handler Handler) *gin.Engine {
  // ...
  router.GET("/.well-known/veraison/endorsement-distribution", handler.GetEdApiWellKnownInfo)

  coservEndpoint := path.Join(edApiPath, "coserv/:query")
  router.GET(coservEndpoint, handler.CoservRequest)

  // ...
}
```

New VTS Service

```
service VTS {
  // endorsement distribution API
  rpc GetEndorsements(EndorsementQueryIn) returns (EndorsementQueryOut);
  // ...
}

message EndorsementQueryIn {
  string media_type = 1; // media type (including profile)
  string query = 2;      // base64url-encoded CoSERV
}

message EndorsementQueryOut {
  Status status = 1;
  bytes result_set = 2;  // CBOR-encoded result-set CoSERV
}
```

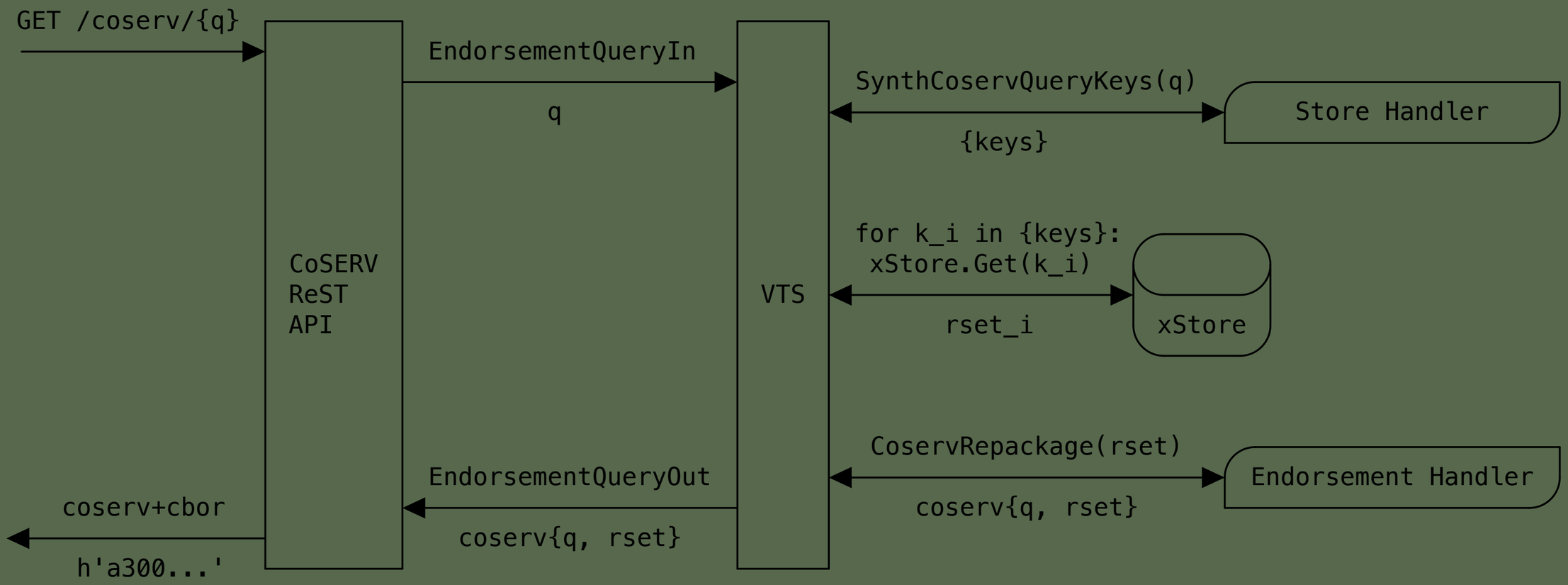
Plugins

Local mode: reuses / expands `Store` and `Endorsement`[1] handlers

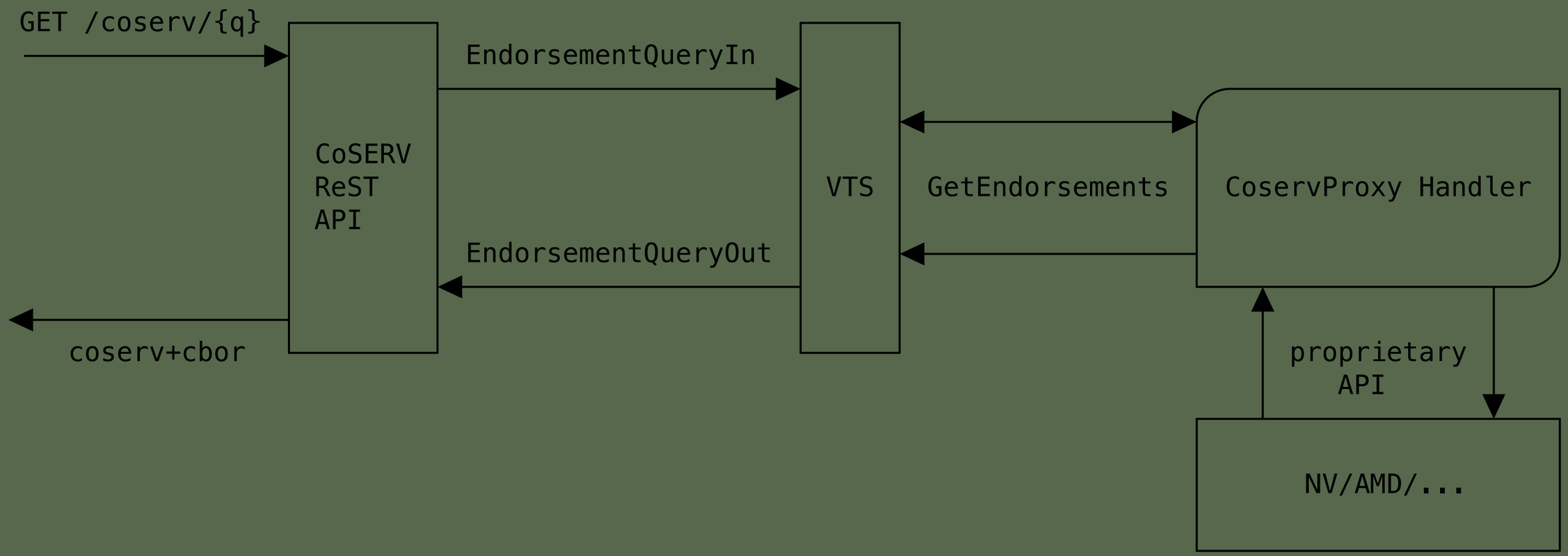
Proxy mode: new `CoservProxy` handler type

[1] Also, small tweak to synthesize a new key for instance queries on ingest.

Local


GET /coserv/{q}
CoSERV
ReST
API
EndorsementQueryIn
q
VTS
SynthCoservQueryKeys(q)
{keys}
Store Handler
for k_i in {keys}:
xStore.Get(k_i)
rset_i
xStore
CoservRepackage(rset)
coserv{q, rset}
Endorsement Handler
EndorsementQueryOut
coserv{q, rset}
coserv+cbor
h'a300...'

Proxy


GET /coserv/{q}
CoSERV
ReST
API
EndorsementQueryIn
EndorsementQueryOut
coserv+cbor
VTS
GetEndorsements
CoservProxy Handler
proprietary
API
NV/AMD/...

TODO

- Expiry timestamp
- HTTP cache control headers
- COSE signature
- Example proxy plugin (NVIDIA and/or AMD)

Pointers

- corim git:(coserv)
- services git:(coserv)