Token Status List

A simple and scalable credential revocation/status mechanism
[Formerly known as JWT CWT Status List]

- Refresher
- Updates since IETF 119
- Discussion points
- Q&A

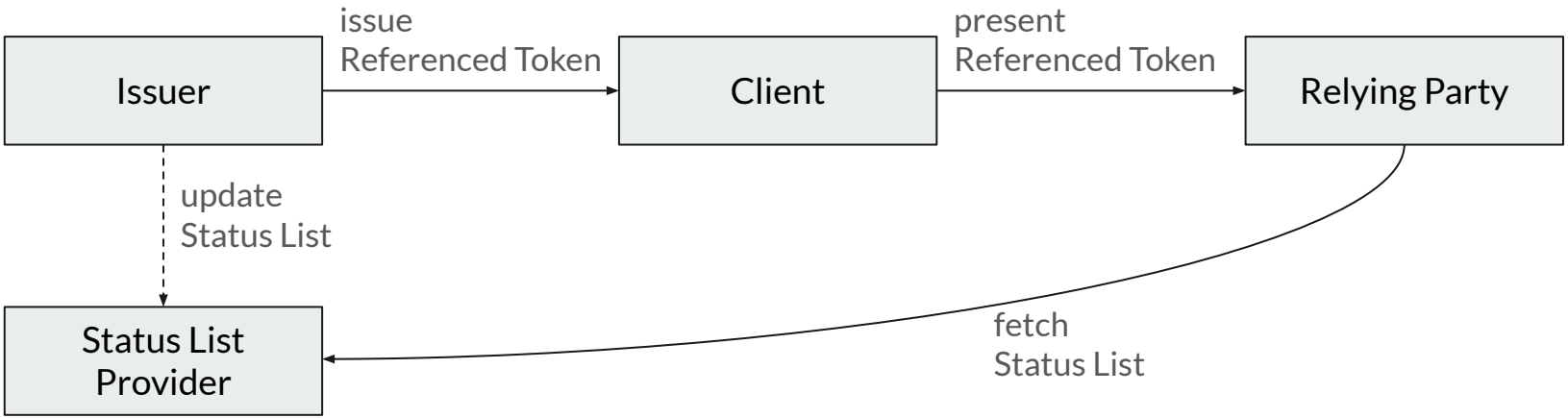
Tobias Looker, Paul Bastian, Christian Bormann

A Refresher - The Problem

How to enable the issuer of a token (e.g CWT or JWT) to communicate dynamic status information about a token after it is issued and before it expires.

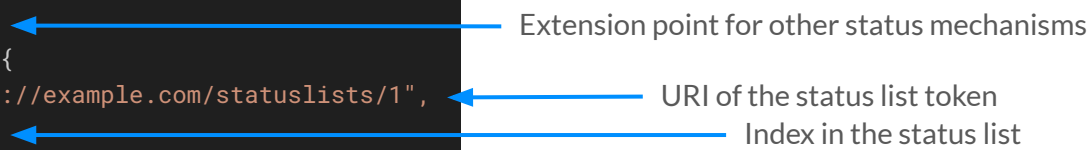
Example - An SD-JWT Verifiable Credential where the Issuer would like to communicate whether the credential is revoked or not.

Big Picture


Issuer
issue
Referenced Token
Client
present
Referenced Token
Relying Party
update
Status List
Status List
Provider
fetch
Status List

Example: Referenced Token

```
{
  "alg": "ES256",
  "kid": "11"
}
.
{
  "iss": "https://example.com",
  … //other claims
  "status": {
    "status_list": {
      "uri": "https://example.com/statuslists/1",
      "idx": 5
    }
  }
}
```

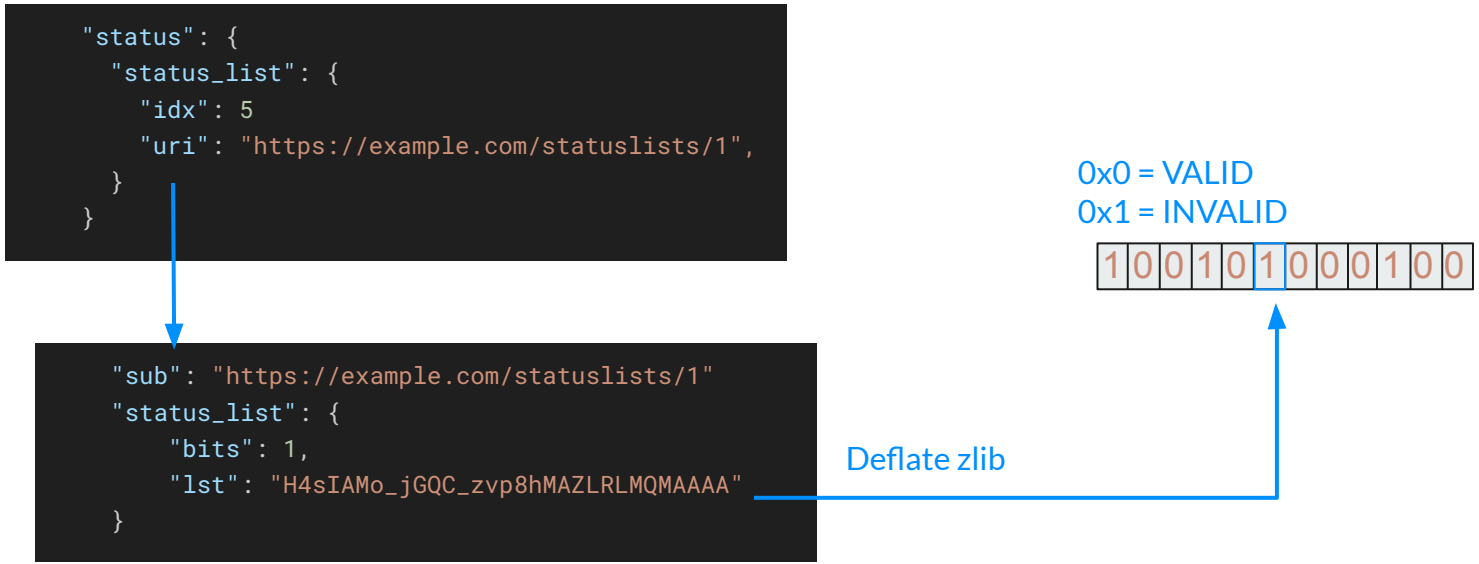


Example: Status List in JWT

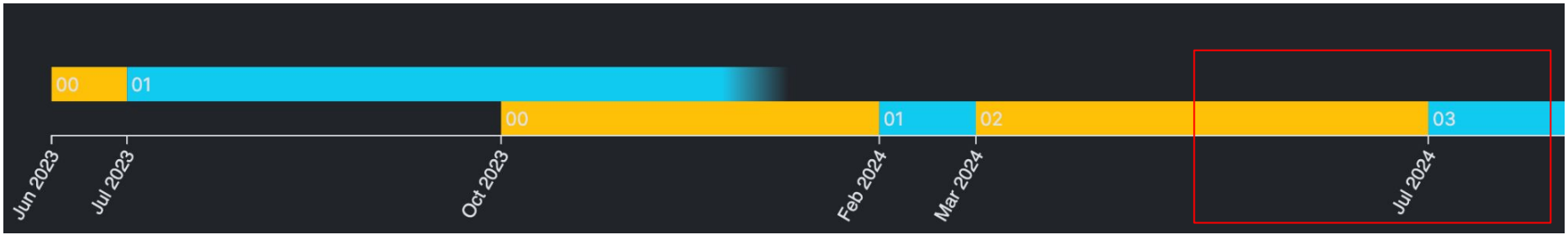
```
eyJhbGciOiJFUzI1NiIsImtpZCI6IjEyIiwidHlwIjoic3RhdHVzbGlzdCtqd3QifQ.eyJleHAiOjE2ODc1MTc3NzAsImlhdCI6MTY4NjkxMjk3MCwiaXNzIjoiaHR0cHM6Ly9leGFtcGxlLmNvbSIsInN0YXR1c19saXN0Ijp7ImJpdHMiOjIsImxzdCI6Ikg0c0lBTW9fakdRQ196dnA4aE1BWkxSTE1RTUFBQUEifSwic3ViIjoiaHR0cHM6Ly9leGFtcGxlLmNvbS9zdGF0dXNsaXN0cy8xIn0.8uaUXshaJdGWGjvwPwaa2Gtt0M7-M7dGO9rXaz3x99LCdG5tKb-ARL1ezqguLTs63VeudYWqpdg4HpN-D2hOkg
```

```
{
    "alg": "ES256",
    "kid": "12",
    "typ": "statuslist+jwt"
}
.
{
    "exp": 1687517770,
    "iat": 1686912970,
    "iss": "https://example.com",
    … //other claims
    "status_list": {
        "bits": 1,
        "lst": "H4sIAMo_jGQC_zvp8hMAZLRLMQMAAAA"
    },
    "sub": "https://example.com/statuslists/1"
}
```

Example: How it fits together



Changes since IETF 119 (Brisbane)



Changes: -03

- remove unused reference to RFC9111
- **add validation rules for status list token**
- **introduce the status list aggregation mechanism**
- relax requirements for status_list claims to contain other parameters
- change cwt referenced token example to hex and annotated hex
- require TLS only for fetching Status List, not for Status List Token
- remove the undefined phrase Status List endpoint
- remove http caching in favor of the new ttl claim
- clarify the sub claim of Status List Token
- relax status_list iss requirements for CWT
- Fixes missing parts & iana ttl registration in CWT examples

Add validation rules for status list token

- First iteration on validation rules
- Demonstrates some of the problems we have with the different options we support
  - Makes the validation rules unnecessarily complex
- We will probably need more iterations on these and try to simplify
- **Feedback welcome!**

Status list aggregation mechanism

- Enable caching of status lists for specific credentials/issuers
- An optional mechanism that allows a RP to get all relevant status lists, e.g. for offline presentations of credentials
- Two modes to retrieve:
  - Embedded in a Status List Token (additional, optional claim)
  - Provided by the issuer via additional metadata (out of scope for this draft)

```
{
  "iat": 1686920170,
  "iss": "https://example.com",
  "status_list": {
    "bits": 1,
    "lst": "eNrbuRgAAhcBXQ",
    "aggregation_uri": "https://example.com/statuslists/all"
  },
  "sub": "https://example.com/statuslists/1"
}
```

```
{
  "status_lists" : [
     "https://example.com/statuslists/1",
     "https://example.com/statuslists/2",
     "https://example.com/statuslists/3"
  ]
}
```

Discussion: Unsigned Option

Current situation

- The current way the unsigned option is defined creates complexity (additional definitions, extra handling for verification etc.)
- All implementations we are aware of are using the signed variant
  - But: There were some voices to keep the unsigned option

We would like to simplify

- Remove the unsigned option entirely?
- Other ideas/opinions?

Discussion: Identifier List?

- As we introduce a registry for status mechanisms in JSON/CBOR, currently two candidates exist:
  - Status-List (this spec)
  - Status-Attestation (Giuseppe's proposal - OCSP-stapling like)
- We also got requests for a simple CRL-like mechanism
  - UUID based instead of index the issuer has to choose → less complexity for issuer
  - Better for cases with very low revocation rates
  - Could allow for additional metadata (like timestamp etc)
- Can re-use/reference big parts of this draft
  - Rough first draft: https://github.com/c2bo/draft-bormann-identifier-list
- Thoughts about this?

Outlook

- Most main sections of the draft are in a good state
- Most major discussion points are resolved
  - 38 open issues mainly about details, editorials
- Our intention is to push the draft to WGLC at next IETF in Dublin

Questions?

Links

Datatracker: https://datatracker.ietf.org/doc/draft-ietf-oauth-status-list/

Git Repository: https://github.com/oauth-wg/draft-ietf-oauth-status-list

Current Editors Copy: https://oauth-wg.github.io/draft-ietf-oauth-status-list/#go.draft-ietf-oauth-status-list.html