Revocation B2B Centralized

1. Table of Contents

2. Introduction

For a number of reasons a state may wish to invalidate a DCC:

- Holder has been tested positive after vaccination or recovery[1]
- DCC has been shared on social media
- DCC has been fraudulently issued
- DCC has been issued based on a wrong lab result
- To correct an operational/technical mistake (DCC has wrong content)

Currently a number of states implement revocation within their apps using a number of methods, within this document just the single DCC revocation method and their distribution over the DCC Gateway will be considered.

[1] There must be a policy decision whether this sort of suspension should be allowed. This concept still covers this requirement from a technical perspective.

3. Assumptions

This technical concept is based on the following assumptions:

- The revocation list will be distributed in batches from the national backends via the DCCG to all other national backends.
- Following links to single DCCs are considered acceptable for distribution:
    - SHA256(signature)
    - SHA256(UCI)
    - SHA256(issuer_country_code + UCI)
- National backends will provide revocation entries to the gateway
- National backends will download the provided data from the gateway and transform it in any appropriate format for the B2A distribution[2]
- Issuers can revoke only their own DCC's (sovereignty).
- The solution must support up to 80M valid revocations of single DCCs (considering that the revocations expire after 2 years or less).
  This includes the B2B distribution (described in this document) and the proposal for B2A distribution (which will be described in a separate document). Dedicated performance tests must be defined to measure the amount of data on the mobile devices and the speed of verification of a single DCC (for the verifier apps).

[2] An additional document handles recommendations for the efficient B2A distribution of revocation information.

4. Scope

4.1 In scope

- System for sharing revocation lists of individual certificates between DCC participating states - from now on referred to as “attendees”
- The revocation lists are shared in an via the DCCG (DCC gateway): National Backend -> DCCG -> National backends of all attendees
- The API will not be exposed to the public, just to national backends

4.2 Out of scope

- Revocation of certificates which have been generated by a DSC for which the private key has leaked, or for which the issuing system has shown to have been compromised are out of scope. Both situations will be dealt with by revocation of the DSC, and/or revocation of the CSCA.
- Mass revocation of DCC based on invalidation rules. This concept is already covered by the EU DCC Validation Rules[3] document.
- The *Backend to App* is explicitly out of scope for this document since it is within the responsibility of the MS. Recommendations for efficient Backend to App distribution of revocation information will be handled in a separated document
- The defined data formats will not be optimized for mobile applications (this is covered by the B2A distribution)

The solution must enable reasonable quick[4] and efficient[5] sharing of revocation lists and must be as privacy preserving as possible whilst still allowing enough flexibility to the implementers of the *Backend to App* flow.

[3] See: https://ec.europa.eu/health/ehealth/covid-19_en#:~:text=EU%20DCC%20Validation%20Rules
Also consider the whole list of DCC specifications: https://ec.europa.eu/health/ehealth/covid-19_en
[4] I.e.commensurate with the speed by which DCC’s move through society, apps are updated, calls to config systems can be made without privacy issues; so hours and minutes; not seconds.
[5] Especially efficient from a verifier (or other in the field app/phone) perspective; as a few extra MBytes at country level can easily balloon a 10.000 fold once copied onto every verifier.

5. Solution

The minimal viable product involves defining a simple but highly secure API which will allow member states to exchange their revocation lists over the gateway. That API is documented under *DRL[6] Provisioning API*.

5.1 Design goals

The Revocation API needs to be finalized and implementation needs to start in the near future. To achieve that we have followed the following design goals.

- Simplicity: the APIs will be implemented by 50+ states so must be simple.
- The standard trust model must be reused.
- Data must be securely shared over the gateway. The DRL are distributed just between the participating countries.

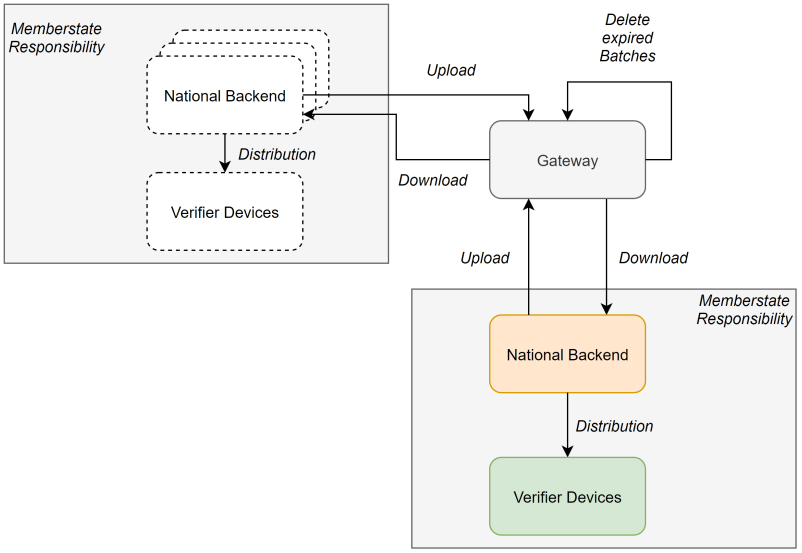
5.2 B2B Communication

Revocation lists of individual certificates are shared through the Digital COVID Certificate Gateway (“the Gateway” or “DCCG”). Attendees upload revocation lists in the form of *batches* of *revocation entries.* A revocation entry signifies the revocation of an individual certificate. A batch is an ID’d, and immutable collection of revocation entries. Its “raison d’être” is to avoid having to sign revocation entries individually before transmitting it to the Gateway, and to be able to delete many revocation entries at once. That could happen automatically after a batch’s expiry date has elapsed. All attendees download a revocation batch *index* of each country within a defined time interval to update their own lists (e.g. within 6 hours).

[6] DRL = DCC Revocation List

5.3 DCC Revocation List (DRL) Provision

The Gateway will provide endpoints and functionality to hold and manage the revocation lists:



5.4 Trust Model

All connections are established by the standard DCCG trust model by the NB_{TLS} and NB_{UP} certificates (see certificate governance). All information is packed and uploaded by CMS messages to ensure the integrity.

5.5 Batch Construction

5.5.1 Batch

Each revocation list contains multiple entries and must be packed in batches which contains a set of hashes and their metadata. A batch is immutable and defines an expiration date which indicates when the batch can be deleted. The expiration date of all items in the batch must be exactly the same - meaning that batches must be grouped by expiry date and by signing DSC. Each batch must contain a maximum of 1000 entries. If the revocation list consists of more than 1000 DCC, multiple batches are created. Any entry may occur in at most one batch. The batch must be packaged into a CMS structure and signed by your NB_{up} certificate.

5.5.2 Batch Index

When a batch is created it will be assigned a unique ID by the gateway and will be automatically added to the index. The index of batches is ordered by the modified date, in ascending chronological order.

5.5.3 Gateway Behavior

The gateway processes revocation batches without any changes: it can neither update nor remove, nor add any information to the batches. The batches are forwarded to all authorized countries (see chapter 5.8).

The gateway actively observes the expiration dates of batches and removes it. After the batch is deleted, the gateway returns for the deleted batch url a gone result and the batch appears in the batch index as “deleted”.

5.6 Hash Types

The revocation list contains hashes which can represent different revocation types/attributes. These types or attributes must be indicated in the provisioning of the revocation lists. The current types are:

Type	**Attribute**	**Hash Calculation**
SIGNATURE	DCC Signature	SHA256 of DCC Signature
UCI	UCI	SHA256 of UCI
COUNTRYCODEUCI	Issuing Country Code + UCI	SHA256 of Issuing CountryCode + UCI

Just the first 128 bits of the hashes are put into the batches and used to identify the revoced DCC[7].

5.6.1 Hash type: SHA256(DCC Signature)

In this case the hash is calculated over the bytes of the COSE_SIGN1 signature from the CWT. For RSA signatures, the entire signature will be used as input. The formula for the EC-DSA signed certificates is using the the `r` value as an input: `SHA256(r)`

[required for all new implementations]

5.6.2 Hash type: SHA256(UCI)

In this case the hash is calculated over the UCI string encoded in UTF-8 and converted to a byte array.

[deprecated[8], but supported for backwards compatibility]

[7] Please also consider 5.7.1.2 for the detailed API descriptions
[8] Deprecated means that this feature shall not be considered for new implementations, but shall be supported for existing implementations for a well defined period of time.

5.6.3 Hash type: SHA256(Issuing CountryCode+UCI)

In this case CountryCode encoded as a UTF-8 string concatenated with the UCI encoded with a UTF-8 string. This is then converted to a byte array and used as input to the hash function.

[deprecated[7], but supported for backwards compatibility]

5.6.4 Ensuring sovereignty of revocations

A member state must only revoke certificates that they have issued themselves. Due to the limitations regarding personal data, this cannot be enforced by the gateway. However the gateway provides enough information such that the Verifier apps can partition the revocation list by the revoking country.

5.7 API Structure

5.7.1 Revocation Entry Provisioning API

5.7.1.1 Purpose

The API delivers the revocation list entries in batches including an batch index.

5.7.1.2 Endpoints

5.7.1.2.1 Batch List Download Endpoint

The endpoints follow a simple design, returning a list of batches with a small wrapper providing metadata. The batches are sorted by *date* in *ascending (chronological)* order:

```
/revocation-list

Verb: GET
Content-Type: application/json
Response: JSON Array

{
      ‘more’:true|false,
      ‘batches’:
                  [{
                        ‘batchId’: ‘{uuid}’,
                        ‘country’: ‘NL’,
                        ‘date’: ‘2021-11-01T00:00:00Z’
                        ‘deleted’: true | false
                  }, ..
                  ]
}
```

Note: The result is limited by default to 1000. If the flag ‘more’ is set to true, the response indicates that more batches are available for download. To download more items the client must set the `If-Modified-Since` header to a date no earlier than the last entry received.

The response contains a CMS including a signature which must match to the NB_{UP} certificate of the country. All items in the JSON array contain the following structure:

Field	**Definition**
more	Boolean Flag which indicates that there are more batches.
batches	Array with the existing batches.
batchId	https://en.wikipedia.org/wiki/Universally_unique_identifier
country	Country Code ISO 3166
date	ISO 8601 Date UTC. Date when the batch was added or deleted.
deleted	boolean. True if deleted. When the deleted flag is set, the entry can be finally removed from the query results after 7 days.

Response Codes

Code	**Description**
200	All ok.
204	No content, if “If-Modified-Since” Header content has no match.

Request Header

Header	**Mandatory**	**Description**
If-Modified-Since	Yes	This header contains the last downloaded date to get just the newest results. On the initial call the header should be the set to the `‘2021-06-01T00:00:00Z’`

5.7.1.2.2 Batch Download Endpoint

The batches contain a list of certificate identifiers:

`/revocation-list/{batchId}`

```
Verb: GET
Accepts: application/cms
Response:CMS with Content

{
      ‘country’: ‘NL’,
      ‘expires’: ‘2022-11-01T00:00:00Z’,
      ‘kid’:’23S+33f=’,
      ‘hashType’:’SIGNATURE’,
      ‘entries’:[{
                  ‘hash’:’e2e2e2e2e2e2e2e2’
                }, ..]
}
```

The response contains a CMS including a signature which must match to the NB_{UP} certificate of the country. All items in the JSON array contain the following structure:

Field	Mandatory	Type	Definition
expires	Yes	String	Date when the item can be removed. ISO8601 Date/Time UTC[9]
country	Yes	String	Country Code ISO 3166
hashType	Yes	String	Hash Type of the provided entries (see Hash Types)
entries	Yes	JSON Object Array	See Table Entries
kid	Yes	String	base64 encoded KID of the DSC used to sign the DSC. If the KID is not known then the string `UNKNOWN_KID` (excluding the `) can be used.

Notes:

- Batches MUST be grouped by expiry date and DSC - all items MUST expire at the same time and have been signed by the same key.

[9] For revocation: the expiry date of the DCC (or the DSC if the DCC isn't known).
For suspension: the end date of the suspension.

- Expiry time is a date/time UTC because that is required because EU-DCC is a global system and we must use an unambiguous time.
- National Backend (NB) MUST remove items from their blacklist when the **expires** date is reached. For suspension this date is the end date of the suspension, for revocation it is the expiry date of the DCC (if known) or the expiry date of the corresponding DSC used to sign the DCC.
- NB MAY remove items from their blacklist in the case that the **kid** used to sign the DCC is revoked.

5.7.1.2.2.1 Entries

Field	Mandatory	Type	Definition
hash	Yes	String	First 128 bits of the SHA256 hash encoded as a base64 string

Note: The entries object contains currently just a hash, but to be compatible with changes in the future an object was chosen, instead of an json array.

Response Codes

Code	Description
200	All ok.
410	Batch gone. Batch can be deleted in the national backend.

Response Headers

Header	Description
ETag	Batch ID.

5.7.1.2.3 Batch Upload Endpoint

The upload is done over the same endpoint via POST Verb:

```
/revocation-list

Verb: POST
Accepts: application/cms
Request: CMS with Content
ContentType: application/cms
Content:
{
      ‘country’: ‘NL’,
      ‘expires’: ‘2022-11-01T00:00:00Z’,
      ‘kid’:’23S+33f=’,
      ‘hashType’:’SIGNATURE’,
      ‘entries’:[{
                  ‘hash’:’e2e2e2e2e2e2e2e2e2’
                 }, ..]
```

```
}
```

The batch will be signed using the NB_{UP} certificate. The Gateway will verify that the signature was set by the NB_{UP} for the given *country*. If the signature check fails then the upload will fail.

NOTE: Every batch is immutable and can't be changed after uploading. It can be deleted, though. The ID of every deleted batch is stored, and an upload of a new batch with the same ID is rejected.

5.7.1.2.4 Batch Delete Endpoint

You can delete a batch over the same endpoint via DELETE Verb:

```
/revocation-list

Verb: DELETE
Accepts: application/cms
ContentType: application/cms
Request: CMS with Content
Content:
{
      'batchId': '...'
}
```

or, for compatibility reasons, to the following endpoint with the POST verb:

```
/revocation-list/delete

Verb: POST
Accepts: application/cms
ContentType: application/cms
Request: CMS with Content
Content:
{
      'batchId': '...'
}
```

5.8 API Protection / GDPR

This section specifies measures for the implementation to comply with the provisions of the Regulation 2021/953 as regards to the processing of personal data.

5.8.1 Existing Authentication

The Gateway currently uses the NB_{TLS} certificate to authenticate the countries connecting to the Gateway. This authentication can be used to determine the identity of the country connected to Gateway. That identity can then be used to implement access control.

5.8.2 Access Control

In order to be able to lawfully process personal data the Gateway will implement an access control mechanism.

The gateway implements an Access Control List combined with Role Based Security. In that scheme, two tables will be maintained - one table describing which Roles can apply which Operations to which Resources, with the other table describing which Roles are assigned to which Users.

In order to implement the controls required by this document, three Roles are required, and that is:

```
RevocationListReader
RevocationUploader
RevocationDeleter
```

The following endpoints will check to see if the User has the Role `RevocationListReader`; if they do then access will be granted, if they do not then an `HTTP 403 Forbidden`will be returned:

```
GET /revocation-list/
GET /revocation-list/{batchId}
```

The following endpoints will check to see if the User has the Role `RevocationUploader`; if they do then access will be granted, if they do not then an `HTTP 403 Forbidden` will be returned:

```
POST /revocation-list
```

The following endpoints will check to see if the User has the Role `RevocationDeleter`; if they do then access will be granted, if they do not then an `HTTP 403 Forbidden` will be returned:

```
DELETE /revocation-list
POST /revocation-list/delete
```

Access control will not be implemented in any of the existing endpoints on the Gateway.

The Gateway must also provide a reliable method whereby the administrators can manage the Roles that are linked to the Users in such a way as to reduce the chance of human errors whilst also not burdening the functional administrators.

6. Appendix

6.1 Appendix 1: Terms

NB	National backend
UCI	Universal Certificate Identifier
Suspended	The DCC is considered invalid for a limited duration of time, for example due to a positive COVID test.

6.2 Considerations about ECDSA Signature

ECDSA signatures come in pairs (see Wikipedia and this article). For each (r,s) valid signature, (r, -s mod n) is also a valid signature. Without mitigation, revoking a DCC would require including two hash(signature) entries.

Mitigation options:

1. Block any signature component `s` that is not a 'low `s` value' greater than half of the curve order `n (s > n/2)` during verification. However, great care should be taken that the many signer implementations all emit ‘low `s` value’ signatures.
2. Convert the signature to canonical 'low `s`' form at hash creation and verification, so that the `s`-component is smaller than half of the curve order `n` (by calculating `-s mod n` if `s` is greater than half the curve order).
3. At verification time, check both the hash of `(r,s)` and `(r, -s mod n)` against the denylist.
4. At hash generation and verification, only use the value of `r` of the signature, and ignore `s`.

Examples :

Original	Forged