ANNEX I

FORMAT AND TRUST MANAGEMENT

1. INTRODUCTION

These technical specifications contain a generic data structure and encoding mechanisms for the EU Digital COVID Certificate (hereinafter called 'DCC'). They also specify a transport encoding mechanism in a machine-readable optical format (hereinafter called 'QR'), which can be displayed on the screen of a mobile device or printed on a piece of paper. The electronic health certificate container formats of these specifications are generic, but in this context used to carry the DCC.

2. TERMINOLOGY

Organisations adopting these specifications for issuing health certificates are called 'Issuers' and organisations accepting health certificates as proof of health status are called 'Verifiers'. Together, these are called 'Participants'. Some aspects set out in this Annex must be coordinated between the Participants, such as the management of a namespace and the distribution of cryptographic keys. It is assumed that a party, hereafter referred to as the 'Secretariat', carries out these tasks.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this annex are to be interpreted as described in BCP 14 (RFC2119, RFC8174) when, and only when, they appear in all capitals, as shown here.

3. ELECTRONIC HEALTH CERTIFICATE CONTAINER FORMAT

The Electronic Health Certificate Container Format (HCERT) is designed to provide a uniform and standardised vehicle for health certificates from different Issuers. The objective of these specifications is to harmonise how these health certificates are represented, encoded and signed with the goal of facilitating interoperability.

The ability to read and interpret a DCC issued by any Issuer requires a common data structure and agreement on the significance of each data field of the payload. To facilitate such interoperability, a common coordinated data structure is defined through the use of a JSON schema that constitutes the framing of the DCC.

3.1. Structure of the payload

The payload is structured and encoded as a CBOR with a COSE digital signature. This is commonly known as a "CBOR Web Token" (CWT), and is defined in RFC 8392. The payload, as defined below, is transported in a `hcert` claim.

The integrity and authenticity of origin of payload data MUST be verifiable by the Verifier. To provide this mechanism, the issuer of MUST sign the CWT using an asymmetric electronic signature scheme as defined in the COSE specification (RFC 8152).

3.2. CWT Claims

3.2.1. CWT Structure Overview

- Protected Header
- Signature Algorithm (`alg`, label 1)

- Key Identifier (`kid`, label 4)
- Payload
- Issuer (`iss`, claim key 1, optional, ISO 3166-1 alpha-2 of issuer)
- Issued At (`iat`, claim key 6)
- Expiration Time (`exp`, claim key 4)
- Health Certificate (`hcert`, claim key -260)
- EU Digital COVID Certificate v1 (`eu_DCC_v1`, claim key 1)
- Signature

3.2.2. Signature Algorithm

The Signature Algorithm (`alg`) parameter indicates what algorithm is used for the creating the signature. It must meet or exceed current SOG-IT guidelines as summarised below.

One primary and one secondary algorithm is defined. The secondary algorithm should only be used if the primary algorithm is not acceptable within the rules and regulations imposed on the implementor.

In order to ensure the security of the system, all implementations have to incorporate the secondary algorithm. For this reason, both the primary and the secondary algorithm MUST be implemented.

The SOG-IT set levels for the primary and secondary algorithms are:

- Primary Algorithm: The primary algorithm is Elliptic Curve Digital Signature Algorithm (ECDSA) as defined in (ISO/IEC 14888–3:2006) section 2.3, using the P–256 parameters as defined in appendix D (D.1.2.3) of (FIPS PUB 186–4) in combination the SHA–256 hash algorithm as defined in (ISO/IEC 10118–3:2004) function 4.

This corresponds to the COSE algorithm parameter ES256.

- Secondary Algorithm: The secondary algorithm is RSASSA-PSS as defined in (RFC 8230) with a modulus of 2048 bits in combination with the SHA–256 hash algorithm as defined in (ISO/IEC 10118–3:2004) function 4.

This corresponds to the COSE algorithm parameter: PS256.

3.2.3. Key Identifier

The Key Identifier (`kid`) claim is used by Verifiers for selecting the correct public key from a list of keys pertaining to the Issuer (`iss`) Claim. Several keys may be used in parallel by an Issuer for administrative reasons and when performing key rollovers. The Key Identifier is not a security-critical field. For this reason, it MAY also be placed in an unprotected header if required. Verifiers MUST accept both options.

Due to the shortening of the identifier (for space-preserving reasons) there is a slim but non-zero chance that the overall list of DSCs accepted by a validator may contain DSCs with duplicate `kid`s. For this reason, a verifier MUST check all DSCs with that `kid`.

3.2.4. Issuer

The Issuer (`iss`) claim is a string value that MAY optionally hold the ISO 3166-1 alpha-2 Country Code of the entity issuing the health certificate. This claim can be used by a Verifier to identify which set of DSCs to use for validation. The Claim Key 1 is used to identify this claim.

3.2.5. *Expiration Time*

The Expiration Time (`exp`) claim SHALL hold a timestamp in the NumericDate format (as specified in RFC 8392 section 2) indicating for how long this particular signature over the Payload SHALL be considered valid, after which a Verifier MUST reject the Payload as expired. The purpose of the expiry parameter is to force a limit of the validity period of the health certificate. The Claim Key 4 is used to identify this claim.

The Expiration Time MUST not exceed the validity period of the DSC.

3.2.6. *Issued At*

The Issued At (`iat`) claim SHALL hold a timestamp in the NumericDate format (as specified in RFC 8392 section 2) indicating the time when the health certificate was created.

The Issued At field MUST not predate the validity period of the DSC.

Verifiers MAY apply additional policies with the purpose of restricting the validity of the health certificate based on the time of issue. The Claim Key 6 is used to identify this claim.

3.2.7. *Health Certificate Claim*

The Health Certificate (`hcert`) claim is a JSON (RFC 7159) object containing the health status information, which has been encoded and serialised using CBOR as defined in (RFC 7049). Several different types of health certificate MAY exist under the same claim, of which the DCC is one.

The JSON is purely for schema purposes. The wire format is CBOR. Application developers may not actually ever de-, or encode to and from the JSON format, but use the in-memory structure.

The Claim Key to be used to identify this claim is -260.

Strings in the JSON object SHOULD be NFC normalised according to the Unicode standard. Decoding applications SHOULD however be permissive and robust in these aspects, and acceptance of any reasonable type conversion is strongly encouraged. If non-normalised data is found during decoding, or in subsequent comparison functions, implementations SHOULD behave as if the input is normalised to NFC.

4. DATA STRUCTURES AND FORMATS FOR THE CREATION OF A 2D CODE

4.1. CBOR/COSE

To optimize the footprint of the 2D Code, the objects are encoded as CBOR[1] object. To ensure the data integrity, the CBOR Object Signature and Encryption[2] is used. For using CBOR the recommended serialization rules[3] are considered.

During the converting the hints MUST be ensured in the CBOR RFC[4].

It must be ensured that all keys in a JSON Object are UTF-8 encoded.

In addition to the UTF-8 encoded last name(s) and first name(s) of the holder, the 2D barcode must include a second set of the same name fields encoded in ASCII following ICAO Document 9303 part 3[5]. The ASCII-encoded personal name should match the name as included in the travel document issued to the holder.

1 https://tools.ietf.org/html/rfc8949
2 https://tools.ietf.org/html/rfc8152
3 https://tools.ietf.org/html/rfc8949#section-4
4 https://tools.ietf.org/html/rfc8949#section-6
5 https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf

The CWT format is used to establish a standard container format for the DCC. It must be ensured that this kind of defined standard claims are not broken by custom claims or retyped by different datatypes.

4.2. Compression algorithm

To compress the COSE data objects, the zlib compression algorithm shall be used.

4.3. 2D Code Versioning

In productive use cases a lot of different 2D Codes with different assumptions can be scanned by the verifier. To ensure that the context of the scanned code is always clear for processing (e.g. used schema, value sets at this moment, rules etc.) a context prefix is established to represent different versions.

The version field is expressed as string value, as in the data context examples below:

Version Value	Version
HC1	Health Certificate Version 1
HC2	Health Certificate Version 2
…	…

Each context maps to a discovery document, which is provided over the DCCG or national backends. The context may be defined specific by each Member State or by the eHealth Network.

To provide this context to optical readers it is attached to the encoded barcode before QR Code Generation:

[**version**]Base45 Encoding

The encoding of base45 is described in a IETF draft[6].

4.4. Use of public key identification

During the scan of a CBOR object by a code scanner, the verification algorithm must be efficiently matching the used crypto material. For this purpose and the view on the future decentralized scenarios, the used crypto key MUST be uniquely identifiable by a verifier. This is realized by inserting the field "kid" into the COSE header. The key identifier is defined as the first truncated 8 Bytes of a SHA256 Hash. The "kid" claim can also be used in the JWK concept.

4.5. Data fields names

In order to save as much bytes as possible in the 2D Code, each field name must be reduced to acronyms. E.g. Subject to "sub" or Issuer to "iss". The selected field names shall be unique within the scope of the selected context

4.6. COSE/CBOR Content

4.6.1. COSE Structure

A COSE structure contains a protected, unprotected and payload object within one CBOR array defined in the Basic Structure of the RFC8152[7].

Name	CBOR Major Type	Type

[6] https://datatracker.ietf.org/doc/draft-faltstrom-base45/

[7] https://tools.ietf.org/html/rfc8152#section-3.1

protected	2	bstr
payload	2	bstr
signature	2	bstr
unprotected	2	empty

The payload "nil" is not allowed for this 2D code and should be rejected. The choice to place the kid in the protected or unprotected header is left to the issuer, all verifiers must accept both.

4.6.2. Signing header

The header of COSE contains the algorithm used and the key identifier:

Name	**CBOR Major Type**	**Placement In Header**	**Type**	**Value**	**Description**
alg	1	protected	nint	-7/-37 (ES256)	Algorithm Field
kid	4	protected	array	First 8 bytes of the hash value	Key Identifier

4.6.3. Common Payload Values

The common dataset for all types of CBOR objects are defined as the following table:

Claim Key	**Name**	**CBOR Major Type**	**Type**	**Description**
1	iss	2	bstr	Issuer of the DCC
6	iat	2	bstr	Issuing Date of the DCC
4	exp	2	bstr	Expiring Date of the DCC
-260	hcert	5	map	Payload of the DCC (Vac,Tst,Rec)

The syntax is according to the CWT RFC8392[8] and the hcert Definition[9].

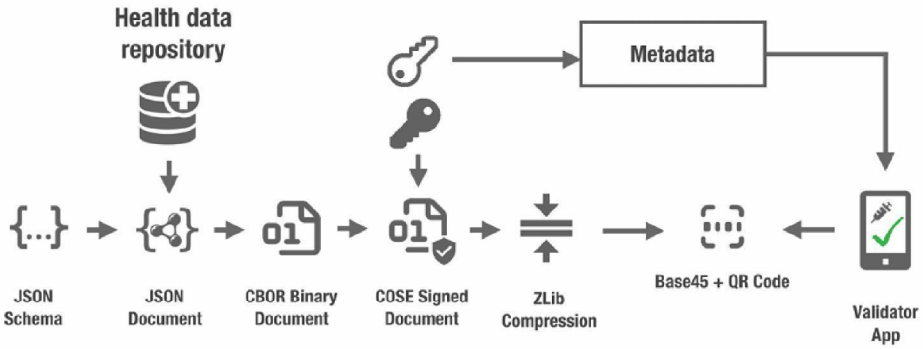
4.7. Optional Data Content

Optional national data content is not allowed. The data content is limited to the defined data elements in the minimum data set specified in the Annex to Regulation 2021/XXXX of the European Parliament and of the Council.

5. SERIALISATION

As serialization pattern, the following scheme is used:

[8] https://tools.ietf.org/html/rfc8392
[9] https://github.com/ehn-digital-green-development/hcert-spec/blob/main/hcert_spec.md



The process should always start with a JSON file, e.g. from a Health Data Repository (external data sources are optionally), which is matching against the defined DCC Schemas. After this checkup, a transformation of human readability may be processed before the serialization to CBOR starts. During this process, it may be decided whether a human readability is useful or not. The acronyms of the claims shall be mapped in every case to the display names before serialization and after deserialization.

It should not be considered to replace the field content with metadata information (e.g. 11 for a value) to safe bytes during the compression. There must be always clear defined values.

New JSON Schemas shall be reported to the eHealth Network to provide this information to all other Member States before using it in productive scenarios.

6. TRANSPORT ENCODINGS

6.1. Raw

For arbitrary data interfaces, the HCERT container and its payloads may be transferred as-is, utilising any underlying, 8 bit safe, reliable data transport. These interfaces MAY include NFC, Bluetooth or transfer over an application layer protocol, for example transfer of an HCERT from the Issuer to a holder's mobile device.

If the transfer of the HCERT from the Issuer to the holder is based on a presentation-only interface (e.g., SMS, e-mail), the Raw transport encoding is obviously not applicable.

6.2. Barcode

6.2.1. Payload (CWT) Compression

To lower size and to improve speed and reliability in the reading process of the HCERT, the CWT SHALL be compressed using ZLIB (RFC 1950) and the Deflate compression mechanism in the format defined in (RFC 1951).

6.2.2. QR 2D Barcode

In order to better handle legacy equipment designed to operate on ASCII payloads, the compressed CWT is encoded as ASCII using Base45 before being encoded into a 2D barcode.

The QR format as defined in (ISO/IEC 18004:2015) SHALL be used for 2D barcode generation. An error correction rate of 'Q' (around 25%) is RECOMMENDED. The Alphanumeric (Mode 2/QR Code symbols 0010) MUST be used in conjunction with Base45.

In order for Verifiers to be able to detect the type of data encoded and to select the proper decoding and processing scheme, the base45 encoded data (as per this specification) SHALL be prefixed by the Context Identifier string "HC1:". Future versions of this specification that impact backwards-compatibility SHALL define a new Context Identifier, whereas the character following "HC" SHALL be taken from the character set [1-9A-Z]. The order of increments is defined to be in that order, i.e., first [1-9] and then [A-Z].

The optical code is RECOMMENDED to be rendered on the presentation media with a diagonal size between 35 mm and 60 mm to accommodate readers with fixed optics where the presentation media is required to be placed on the surface of the reader.

If the optical code is printed on paper using low-resolution (< 300 dpi) printers, care must be taken to represent each symbol (dot) of the QR code exactly square. Non-proportional scaling will result in some rows or columns in the QR having rectangular symbols, which will hamper readability in many cases.

7. Trusted List Format (DSC list)

Each Member State is REQUIRED to provide a list of one or more Certificate Signing Certificate Authorities (CSCAs) and a list of all valid Document Signing Certificates (DSCs), and keep these lists current.

For the list of CSCA certificates, each certificate:

- MUST contain a valid Country attribute in the subject DN that matches the country of issuance.
- MUST contain DN that is unique within the specified country.
- MUST contain a unique Subject Key Identifier (SKI) according to (RFC5280)

In addition, for the list of DSC certificates, each certificate:

- MUST be signed with the private key corresponding to a CSCA certificate published on the aforementioned list.
- MUST contain an Authority Key Identifier (AKI) matching the Subject Key Identifier (SKI) of the issuing CSCA certificate.
- MUST have a validity period that is in line with or longer than the validity period of all certificates signed using that key.
- SHOULD contain a unique Subject Key Identifier derived from the subject public key.

7.1. Simplified CSCA/DSC

As of this version of the specifications, countries should NOT assume that any Certificate Revocation List (CRL) information is used; or that the Private Key Usage Period is verified by implementors.

Instead, the primary validity mechanism is the presence of the certificate on the most recent version of that certificate list.

7.2. ICAO eMRTD PKI and Trust Centers

Member States can use a separate CSCA (as per the WHO advice)(#ref) - but may also use submit their existing eMRT CSCA and/or DSC certificates; and may even chose to procure these from (commercial) trustcenters - and submit these. However, any DSC certificate must always be signed by the CSCA submitted by that country.

8. Security Considerations

When designing a scheme using this specification, Member States shall identify, analyse and monitor certain security aspects.

The following aspects SHOULD be taken into account as a minimum:

8.1. HCERT signature validity time

The Issuer of HCERTs is required to limit the validity period of the signature by specifying a signature expiry time. This requires the holder of a health certificate to renew it at periodic intervals.

The acceptable validity period may be determined by practical constraints. For example, a traveller may not have the possibility to renew the health certificate during a trip overseas. However, it may also be the case that an Issuer is considering the possibility of a security compromise of some sort, which requires the Issuer to withdraw an DSC (invalidating all health certificates issued using that key which is still within their validity period). The consequences of such an event may be limited by regularly rolling Issuer keys and requiring renewal of all health certificates, on some reasonable interval.

8.2. Key management

This specification relies heavily on strong cryptographic mechanisms to secure data integrity and data origin authentication. Maintaining the confidentiality of the private keys is therefore of utmost importance.

The confidentiality of cryptographic keys can be compromised in a number of different ways, for instance:

- The key generation process may be flawed, resulting in weak keys.
- The keys may be exposed by human error.
- The keys may be stolen by external or internal perpetrators.
- The keys may be calculated using cryptanalysis.

To mitigate against the risks that the signing algorithm is found to be weak, allowing the private keys to be compromised through cryptanalysis, this specification recommends all Participants to implement a secondary fallback signature algorithm based on different parameters or a different mathematical problem than the primary.

As regards the risks mentioned related to the Issuers' operating environments, mitigations measures to ensure effective control shall be implemented such as to generate, store and use the private keys in Hardware Security Modules (HSMs). Use of HSMs for signing health certificates is highly encouraged.

Regardless of whether an Issuer decides to use HSMs or not, a key roll-over schedule SHOULD be established where the frequency of the key roll-overs is proportionate to the exposure of keys to external networks, other systems and personnel. A well-chosen roll-over schedule also limits the risks associated with erroneously issued health certificates, enabling an Issuer to revoke such health certificates in batches, by withdrawing a key, if required.

8.3. Input data validation

These specifications may be used in a way that implies receiving data from untrusted sources into systems that may be of mission-critical nature. To minimise the risks associated with this attack vector, all input fields MUST be properly validated by data types, lengths and contents. The Issuer Signature SHALL also be verified before any processing of the contents of the HCERT takes place. However, the validation of the Issuer Signature implies parsing the Protected Issuer Header first, in which a potential attacker may attempt to inject carefully crafted information designed to compromise the security of the system.

9. Trust Management

The signature of the HCERT requires a public key to verify. Member States shall make these public keys available. Ultimately, every Verifier needs to have a list of all public keys it is willing to trust (as the public key is not part of the HCERT).

A simplified variation on the ICAO "Master list" shall be used, tailored to this health certificate application, whereby each country is ultimately responsible for compiling their own master list and making that available to the other Participants. The aid of a coordinating Secretariat for operational and practical purposes shall be available.

The system consists of (only) two layers; for each Member State one or more country level certificates that each signs one or more document signing certificates that are used in day to day operations.

The Member State certificates are called Certificate Signer Certificate Authorities (CSCAs) and are (typically) self-signed certificates. Member States may have more than one (e.g., in case of regional devolution). These CSCA certificates regularly sign the Document Signing Certificates (DSCs) used for signing HCERTs. Member States will each maintain a public register of the DSC certificates that is kept current, communicated to the Secretariat and also published at a stable URL for bilateral exchange. Member States MUST remove any revoked or stale certificates from this list.

The "Secretariat" is a functional role. It shall regularly aggregate and publish the Member States DSCs, after having verified these against the list of CSCA certificates (which have been conveyed and verified by other means).

The resulting list of DSC certificates shall then provide the aggregated set of acceptable public keys (and the corresponding kids) that Verifiers can use to validate the signatures over the HCERTs. Verifiers MUST fetch and update this list regularly.

Member States may also bilaterally exchange CSCA certificates with a number of other Member States, verify these bilaterally and thus compile their own lists of CSCA and DSC certificates which is specific to that Member State. Verifiers may choose to rely on such a national list.

Such Member State-specific lists may be adapted in the format for their own national setting. As such, the file format of this trusted list may vary, e.g., it can be a signed JWKS (JWK set

format per RFC 7517 section 5) or any other format specific to the technology used in that Member State.

In order to ensure simplicity, Member States may both submit their existing CSCA certificates from their ICAO eMRTD systems or, as recommended by the WHO, create one specifically for this health domain.

9.1. The Key Identifier (kids)

The key identifier (kid) is calculated when constructing the list of trusted public keys from DSC certificates and consists of a truncated (first 8 bytes) SHA-256 fingerprint of the DSC encoded in DER (raw) format.

Verifiers do not need to calculate the kid based on the DSC certificate and can directly match the key identifier in issued health certificate with the kid on the trusted list.

9.2. Differences to the ICAO eMRTD PKI trust model

While patterned on best practices of the ICAO eMRTD PKI trust model, a number of simplifications shall be made in the interest of speed:

- A Member State may submit multiple CSCA certificates.
- The DSC (key usage) validity period may be set to any length not exceeding the CSCA and MAY be absent.
- The DSC certificate MAY contain policy identifiers (Extended Key Usage) that are EHN specific.
- Member States may choose to never do any verification of published revocations; but instead purely rely on the DSC lists they get daily from the Secretariat or compile themselves.

9.3. Extended key Usage Identifiers

The document signing certificate MAY contain Extended Key Usage extension fields; these being:

- OID 1.3.6.1.4.1.0.1847.2021.1.1 -- valid for test
- OID 1.3.6.1.4.1.0.1847.2021.1.2 -- valid for vaccinations
- OID 1.3.6.1.4.1.0.1847.2021.1.3 -- valid for recovery

The DSC may contain an extended key usage extension with zero or more key usage policy identifiers that constrain the types of HCERTs this certificate is allowed to verify. If present the verifiers SHALL verify the key usage against the stored HCERT.

In absence of any key usage extension, this certificate can be used to validate any type of HCERT. Other documents MAY define relevant additional extended key usage policy identifiers used with validation of HCERTs.

ANNEX III

COMMON STRUCTURE OF THE UNIQUE CERTIFICATE IDENTIFIER

1. INTRODUCTION

Each EU digital COVID certificate (DCC) shall include a unique certificate identifier (UCI). The UCI may be used, at later stages, to verify the certificate. The UCI helps support the interoperability of the DCCs, while it should be implemented under the responsibility of the Member States. The UCI is a means to verify the veracity of the certificate and, where applicable, to link to a registration system (for example, an IIS). These identifiers will also enable (paper and digital) assertions by the Member States that individuals have been vaccinated or tested.

2. COMPOSITION OF THE UNIQUE IDENTIFIER

UCI shall follow a common structure and format easing human- and/or machine-interpretability of information and could relate to elements such as country of vaccination, the vaccine itself and a Member State specific identifier. It should ensure flexibility to Member States to format it, in full respect of data protection EU and national legislation. The order of the separate elements follows a defined hierarchy that can enable future modifications of the blocks while maintaining its structural integrity.

The possible solutions for the composition of the UCI form a spectrum wherein the modularity and human-interpretability are the two main diversifying parameters and one fundamental characteristic:

- Modularity: the degree to which the code is composed of distinct building blocks that contain semantically different information
- Human-interpretability: the degree to which the code is meaningful or can be interpreted by the human reader
- Globally unique; the Country or Authority identifier is well-managed; and each country (authority) is expected to manage its segment of the namespace well by never recycling or re-issuing identifiers. The combination of this ensures that each identifier is globally unique.

3. GENERAL REQUIREMENTS

The following are required in relation to the UCI:

(1) Charset: Only uppercase US-ASCII alpha numerical characters ('A' to 'Z', '0' to '9') are allowed; with additional special characters for separation from RFC3986[10], namely {'/','#',':'};

(2) Maximum length: designers should try to aim for a length of 27-30 characters[11];

(3) Version prefix: This refers to the version of the UCI schema. The version prefix is '01' for this version of the document; the version prefix is composed of two digits;

---

10 Fields such as Sex, Batch/lot number, Administering centre, Health Professional identification, Next vaccination date may not be needed for purposes other than medical use.

11 For implementation with QR codes, Member States could consider an extra set of characters up to a total length of 72 characters (including the 27-30 of the identifier itself) may be used to convey other information. The specification of this information is up to the Member States to define.

(4) Country prefix: The country code is specified by ISO 3166-1. Longer codes (e.g. 3 characters and up (e.g 'UNHCR') are reserved for future use;

(5) Code suffix / Checksum:

5.1 Member States should use a checksum when it is likely that transmission, (human) transcription or other corruptions may occur (i.e. when used in print).

5.2 The checksum must not be relied upon for validating the certificate and is not technically part of the identifier but is used to verify the integrity of the code. This checksum should be the ISO-7812-1 (LUHN-10)[12] summary of the entire UCI in digital/wire transport format. The checksum is separated from the rest of the UCI by a '#' character.

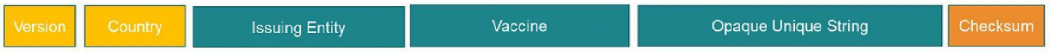
Backwards-compatibility should be ensured: over time Member States that change the structure of their identifiers (within the main version, currently set at v1) must ensure that any two identifiers that are identical represent the same vaccination certificate/assertion. Or in other words; Member States cannot recycle identifiers.

4. OPTIONS FOR UNIQUE IDENTIFIERS FOR VACCINATION CERTIFICATES

The different options presented below are available to Member States and other parties and may co-exist among different Member States. Member States may deploy different options in different version of the UCI schema. The UCI should clearly allow distinguishing which option is applied in a given Member State.

In both Options 1 and 3, vaccine manufacturers should preferably be internationally identifiable.

Option 1 - identifier with semantics



This is the most modular approach and consists of three blocks. The issuing entity refers to the authority issuing the certificate while the vaccine block provides information about the vaccine shot used. Finally, the opaque unique string pertains to the vaccinated individual. Member States are free to determine how each block is coded. For example, the vaccine block could encode different data elements in different Member State implementations (i.e. vaccine product identifier, vaccine/lot identifier(s)), depending also on the data availability. Each block will be able to be understood by a human reader (assuming they can interpret the coding). This solution gives the greatest latitude to Member States to populate each block in the manner they see fit by exploiting existing event or evidence/status identifiers, for the registries of authorised vaccination providers.

Each block should consist of alphanumeric characters (i.e. special characters are not allowed within a block). Alphanumeric blocks should be separated by the special character '/'. If two UCIs are identical up until the first and/or second slash, this means that they are issued by the same issuing entity and/or that the same vaccine lot/batch has been used. Member States are responsible for defining the specifications of each block as they see fit. For instance, the Member States can determine the length of each block based on their actual needs, as long as the total cumulative length of all the blocks as well as the separators does not exceed the defined total length of the identifier (see requirement 2 of the General Requirements).

---

[12] The Luhn mod N algorithm is an extension to the Luhn algorithm (also known as mod 10 algorithm) which works for numeric codes and is used for example for calculating the checksum of credit cards. The extension allows the algorithm to work with sequences of values in any base (in our case alpha characters).

This will result in greater heterogeneity but will also enhance the possibilities for offline and analogue verification. The option makes it easy to generate and write down the UCI (e.g. by hand in paper-based documents).

To avoid having the UCI include personally identifiable information (PII), Member States are strongly urged to refrain from using, for example, a Social Security number or similar long-term stable identifier.

It is recommended therefore that Member States use a non-guessable, random event identifier rather than an identifier that reveals something about the bearer. And use the country's IIS or other registration system to hold the provenance and identity of the bearer.

Option 2 - opaque identifier - no structure

Apart from the country code and the code version in the beginning and the checksum of at the end, the code is not modular but it consists of a single field. This single field serves as the unique identifier of the vaccination in the national vaccination registry of the corresponding country. It is the Member states' responsibility to determine the mechanism for generating and indexing the aforementioned single unique vaccination identifiers.

The opaque unique string should consist of alphanumeric characters exclusively; no other characters (e.g. "/") are allowed. This option provides the maximum flexibility to the Member States in the management of their UCIs.

Option 3 - some semantics

This option consists of two fields: the issuing entity and the opaque unique string. As opposed to Option 2, the opaque unique string does not need to contain information about the issuing entity. The use of an opaque unique string transfers the responsibility to the Member State for generating the opaque unique string while removing the human interpretability requirement. As in Option 2, Member States will be responsible for determining the mechanism for generating and indexing the opaque unique strings.

The two blocks should consist of alphanumeric characters exclusively; no other characters (e.g., "/") are allowed. The blocks are separated by the slash ("/") character and if two UCIs compare identical up until the slash, this means that they are issued by the same issuing entity.

It is possible that the definitions of fields change in the future; and that fields need to be added. The use of RFC3986 allows for such; in a manner well understood in internet engineering and available libraries[13].

[13] https://tools.ietf.org/html/rfc3986

ANNEX IV

PUBLIC KEY CERTIFICATE GOVERNANCE

1. INTRODUCTION

The secure and trusted exchange of signature keys for EU digital COVID certificates (DCCs) between Member States is realized by the EU Digital COVID Certificate Gateway (DCCG), which acts as a central repository for the public keys. With the DCCG, Member States are empowered to publish the public keys that they use to sign digital COVID certificates. Relying Member States can use the DCCG to fetch up-to-date public key material on a timely basis. Later, the DCCG can be extended to exchange trustworthy supplementary information that the Member States provide, like validation rules for digital green certificates. The trust model of the EU DCC framework is a Public Key Infrastructure (PKI). Each Member State maintains one or more Country Signing Certificate Authority (CSCA), certificates of which are relatively long lived. The CSCA issues public key certificates for the national, short lived, Document Signers (i.e. signers for digital green certificates), which are called Document Signer Certificates (DSCs). The CSCA acts as a trust anchor such that relying Member States can use the CSCA certificate to validate the authenticity and integrity of the regularly changing DSC certificates. Once validated, Member States can provide these certificates (or just the public keys contained therein) to their DCC validation applications. Besides CSCAs and DSCs, the DCCG also relies on PKI to authenticate transactions, sign data, as the basis for authentication and as a means to ensure integrity of the communication channels between Member States and the DCCG.

Digital signatures can be used to achieve data integrity and authenticity. Public Key Infrastructures establish trust by binding public keys to verified identities (or issuers). This is necessary to allow other participants to verify the data origin and the identity of the communication partner and decide about trust. In the DCCG, multiple public key certificates are used for authenticity. This annex defines which public key certificates are used and how they should be designed in order to allow broad interoperability among Member States. It provides more details on the necessary public key certificates and it gives guidance on certificate templates and validity periods for countries that want to operate their own CSCA. Since DCCs shall be verifiable for a defined timeframe (starting from the issuing, expire after a given time), it is necessary to define a verification model for all signatures applied on the public key certificates and the digital green certificates.

2. TERMINOLOGY

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this annex are to be interpreted as described in BCP 14 (RFC2119, RFC8174) when, and only when, they appear in all capitals, as shown here.

The following table contains abbreviations and terminology used throughout this annex.

Term	Definition
Certificate	Or public key certificate. An X.509 v3 certificate that contains the public key of an entity

CSCA	Country Signing Certificate Authority
DCC	EU Digital COVID Certificate. A signed digital document that contains vaccination, test or recovery information
DCCG	EU Digital COVID Certificate Gateway. This system is used to exchange DSCs between the Member States
$DCCG_{TA}$	The Trust Anchor certificate of the DCCG. The corresponding private key is used to sign the list of all CSCA certificates offline
$DCCG_{TLS}$	The TLS server certificate of the DCCG
DSC	Document Signer Certificate. The Public Key Certificate of a Member State's Document Signing Authority (e.g. a system that is allowed to sign DCCs). This certificate is issued by the CSCA of the Member State
EC-DSA	Elliptic Curve Digital Signature Algorithm. A cryptographic signature algorithm based on elliptic curves
Member State	Member State of the European Union
mTLS	Mutual TLS. The Transport Layer Security Protocol with mutual authentication
NB	National Backend of a Member State
NB_{CSCA}	The CSCA certificate of a Member State (could be more than one)
NB_{TLS}	The TLS client authentication certificate of a national backend
NB_{UP}	The certificate that a national backend uses to sign data packages that are uploaded to the DCCG
PKI	Public Key Infrastructure. Trust model based on public key certificates and certificate authorities
RSA	Asymmetric cryptographic algorithm based on integer factorization used for digital signatures or asymmetric encryption

3. DCCG COMMUNICATION FLOWS AND SECURITY SERVICES

This section gives an overview of the communication flows and security services in the DCCG system. It also defines which keys and certificates are used to protect the communication, the uploaded information, the digital green certificates, and a signed trust list that contains all on boarded CSCA certificates. The DCCG works as a data hub that allows the exchange of signed data packages for Member States.

Uploaded data packages are provided by the DCCG "as is", meaning that the DCCG does not add or delete DSCs from the packages it receives. The national backend (NB) of the Member States shall be enabled to verify the end-to-end integrity and authenticity of the uploaded data.

In addition to this - National Backends and the DCCG will use mutual TLS authentication to establish a secure connection. This is in addition to the signatures in the data exchanged.

3.1. Authentication and connection establishment

The DCCG uses Transport Layer Security (TLS) with mutual authentication to establish an authenticated encrypted channel between the Member State's national backend (NB) and the Gateway environment. Therefore, the DCCG holds a TLS server certificate, abbreviated DCCGTLS - and the National Backends hold a TLS client certificate – abbreviated NB_{TLS}. Certificate templates are provided in Section 4. Every national backend can provide their own TLS certificate. This certificate will be whitelisted explicitly and thus may be issued by a publicly trusted certificate authority (e.g. a certificate authority that follows the baseline requirements of the CA Browser forum), by a national certificate authority or it can be self-signed. Every Member State is responsible for their national data and the protection of the private key used to establish the connection to the DCCG. The "bring your own certificate" approach requires a well-defined registration and identification process as well as revocation and renewal procedures as described in Section 3.1. The DCCG uses a whitelist where the TLS certificates of NBs are added after their successful registration. Only NBs that authenticate themselves with a private key that corresponds to a certificate from the whitelist can establish a secure connection to the DCCG. The DCCG will also use a TLS certificate that allows the NBs to verify that they are indeed establishing a connection to the "real" DCCG and not some malevolent entity posing as DCCG. The certificate of the DCCG will be provided to the NBs after successful registration. The $DCCG_{TLS}$ certificate will be issued from a publicly trusted CA (included in all major browsers). It is the responsibility of the Member States to verify that their connection to the DCCG is secure (for example, by checking the fingerprint of the $DCCG_{TLS}$ certificate of the server connected to against the one provided post registration).

3.2. Country Signing Certificate Authorities and Validation Model

Member States taking part in the DCCG framework MUST use a CSCA to issue the DSCs. Member States MAY have more than one CSCA, e.g. in case of regional devolution. Each Member State can either use existing certificate authorities or they can setup a dedicated (possibly self-signed) certificate authority for the DCC system.

Member States MUST present their CSCA certificate(s) to the DCCG operator during the official onboarding procedure. After successful registration of the Member State (*see section 4.1 for more details*), the DCCG operator will update a signed trust list that contains all CSCA certificates that are active in the DCC framework. The DCCG operator will use a dedicated asymmetric key pair to sign the trust list and the certificates in an offline environment. The private key will not be stored on the online DCCG system, such that a compromise of the online system does not enable an attacker to compromise the trust list. The corresponding trust anchor certificate $DCCG_{TA}$, will be provided to the National Backends during the onboarding process.

Member States can retrieve the trust list from the DCCG for their verification procedures. The CSCA is defined as the certificate authority that issues DSCs, hence Member States that use a multi-tier CA hierarchy (e.g. Root CA -> CSCA -> DSCs) MUST provide the subordinary certificate authority that issues the DSCs. In this case, if a Member State uses an existing certificate authority, then the DCC system will ignore anything above the CSCA and whitelist only the CSCA as the trust anchor (even though it is a sub-ordinate CA). This is as the ICAO model, only allows for exactly 2 levels - a 'root' CSCA and a single 'leaf' DSC signed by just that CSCA.

In case a Member State operates its own CSCA, the Member State is responsible for the secure operation and key management of this CA. The CSCA acts as the trust anchor for DSCs, and therefore protecting the private key of the CSCA is essential for the integrity of the DCC environment. The verification model in the DCC PKI is the shell model, which states that all certificates in the certificate path validation must be valid at a given time (i.e. the time of signature validation). Therefore, the following restrictions apply:

- The CSCA SHALL NOT issue certificates that are longer valid than the CA certificate itself;
- The document signer SHALL NOT sign documents that are longer valid than the DSC itself;
- Member States that operate their own CSCA MUST define validity periods for their CSCA and all issued certificates and they MUST take care of certificate renewal.

Section 4.2 contains recommendations for validity periods.

3.3. Integrity and authenticity of uploaded data

National backends can use the DCCG to upload and download digitally signed data packages after successful mutual authentication. In the beginning, these data packages contain the DSCs of the Member States. The key pair that is used by the national backend for the digital signature of uploaded data packages in the DCCG system is called National Backend upload signature key pair and the corresponding public key certificate is abbreviated by NBUP certificate. Each Member State brings its own NBUP certificate, which can be self-signed, or issued by an existing certificate authority, such as a public certificate authority (i.e. a certificate authority that issues certificate in accordance with the CAB-Forum baseline requirements). The NBUP certificate shall be different from any other certificates used by the Member State (i.e. CSCA, TLS client or DSCs).

The Member States MUST provide the upload certificate to the DCCG operator during the initial registration procedure (*see Section 4.1 for more details*). Every Member State is responsible for their national data and it must protect the private key that is used for signing the uploads.

Other Member States can verify the signed data packages using the upload certificates that are provided by the DCCG. The DCCG verifies the authenticity and integrity of the uploaded data with the NB upload certificate before they are provided to other Member States.

3.4. Requirements on the technical DCCG architecture

The requirements on the technical DCCG architecture are as follows:

- The DCCG uses mutual TLS authentication to establish an authenticated encrypted connection with the NBs. Therefore, the DCCG maintains a whitelist of registered NB_{TLS} client certificates;
- The DCCG uses two digital certificates ($DCCG_{TLS}$ and $DCCG_{TA}$) with two distinct key pairs. The private key of the $DCCG_{TA}$ key pair is maintained offline (not on the online components of the DCCG);
- The DCCG maintains a trust list of the NB_{CSCA} certificates that is signed with the $DCCG_{TA}$ private key;
- The ciphers used MUST meet the requirements from *Section 5.1*.

4. CERTIFICATE LIFECYCLE MANAGEMENT

4.1. Registration of National Backends

Member States MUST register with the DCCG operator to take part in the DCCG system. This section describes the technical and operational procedure that MUST be followed to register a national backend.

The DCCG operator and the Member State MUST exchange information on technical contact persons for the onboarding process. It is assumed that the technical contact persons are legitimated by their Member States and identification/authentication is performed over other channels. For example, the authentication can be achieved when the Member States technical contact provides the certificates as password-encrypted files via E-Mail and shares the corresponding password with the DCCG operator via telephone.

The Member State MUST provide three digital certificates during the registration and identification process:

- The Member States TLS certificate NB_{TLS}
- The Member States upload certificate NB_{UP}
- The Member States CSCA certificate(s) NB_{CSCA}

All provided certificate MUST adhere to the requirements defined in *Section 5*. The DCCG operator will verify that the provided certificate adheres to the requirements of *Section 5*. After the identification and registration, the DCCG operator:

- adds the NB_{CSCA} certificate(s) to the trust list signed with the private key that corresponds to the DCCGTA public key;
- adds the NB_{TLS} certificate to the whitelist of the DCCG TLS endpoint;
- adds the NB_{UP} certificate to the DCCG system;
- provides the $DCCG_{TA}$ and $DCCG_{TLS}$ public key certificate to the Member State.

4.2. Certificate authorities, validity periods and renewal

In case that a Member State wants to operate its own CSCA, the CSCA certificates will most probably be self-signed certificates. They act as the trust anchor of the Member State and therefore the Member State must strongly protect the private key corresponding to the CSCA certificate's public key. It is recommended that the Member States use an offline system for their CSCA, i.e. a computer system that is not connected to any network. Multi person control should be used to access the system (e.g. following the four eyes principle). After signing DSCs, operational controls should be applied and the system that holds the private CSCA key should be stored safely with strong access controls. Hardware Security Modules or Smart Cards can be used to further protect the CSCA private key. Digital certificates contain a validity period that enforces certificate renewal. Renewal is necessary to use fresh cryptographic keys and to adapt the key sizes when new improvements in computation or new attacks threaten the security of the cryptographic algorithm that is used. The shell model applies (see *Section 3.2*).

The following validity periods are recommended based given the one-year validity for digital COVID certificates:

- CSCA: 4 years
- DSC: 2 years

- Upload: 1-2 years
- TLS Client authentication: 1-2 years

For a timely renewal, the following usage period for the private keys are recommended:

- CSCA: 1 year
- DSC: 6 months

Member States MUST create new upload certificates and TLS certificates timely, e.g. one month, before expiration in order to allow smooth operation. CSCA and DSC SHOULD be renewed at least one month before the private key usage ends (considering the necessary operational procedures). Member States MUST provide updated CSCA, upload and TLS certificates to the DCCG operator. Expired certificates SHALL be removed from the whitelist and trust list.

Member States and the DCCG operator MUST keep track of the validity of their own certificates. There is no central entity that keeps record of the certificate validity and informs the participants.

4.3. Revocation of certificates

In general, digital certificates can be revoked by their issuing CA using certificate revocation lists or Online Certificate Status Responder (OCSP). CSCAs for the DCC system SHOULD provide certificate revocation lists (CRLs). Even if these CRLs are currently not used by other Member States, they SHOULD be integrated for future applications. In case a CSCA decides not to provide CRLs, the DSC certificates of this CSCA must be renewed when CRLs become mandatory. A CSCA SHOULD NOT use the Online Certificate Status Protocol (OCSP) for their DSCs, due to privacy concerns. Verifiers SHOULD NOT use OCSP for validation of the DSCs and SHOULD use CRLs. It is RECOMMENDED that the national backend performs necessary validation of DSC certificates downloaded from the DCC Gateway and only forwards a set of trusted and validated DSC to national DCC validators. DCC validators SHOULD NOT perform any revocation checking on DSC in their validation process. One reason for this is to protect the privacy of DCC holders by avoiding any chance that the use of any particular DSC can be monitored by its associated OCSP responder.

Member States can remove their DSCs from the DCCG on their own using valid upload and TLS certificates. It must be noted that removing a DSC certificate will mean that all DCCs issued with this DSC will become invalid when Member States fetch the updated DSC lists. Clearly, the protection of private key material corresponding to DSCs is crucial. Member States MUST inform the DCCG operator when they must revoke upload or TLS certificates, for example due to compromise of the national backend. The DCCG operator can then remove the trust for the affected certificate, e.g. by removing it from the TLS whitelist. The DCCG operator can remove the upload certificates from the DCCG database. Packages signed with the private key corresponding to this upload certificate will become invalid when national backends remove the trust of the revoked upload certificate. In case that a CSCA must be revoked, Member States SHALL inform the DCCG operator as well as other Member States that they have trust relationships with. The DCC operator will issue a new trust list where the affected certificate is not contained anymore. All DSCs issued by this CSCA will become invalid when Member States update their national backend trust store. In case that the $DCCG_{TLS}$ certificate or the $DCCG_{TA}$ certificate must be revoked, the DCCG operator and the Member States must work together to establish a new trusted TLS connection and trust list.

5. CERTIFICATE TEMPLATES

The following sections contain cryptographic requirements and guidance as well as requirements on certificate templates. For the DCCG certificates, this section defines the certificate templates.

5.1. Cryptographic requirements

Cryptographic algorithms and TLS cipher suites shall be chosen based on the current recommendation from the German Federal Office for Information Security (BSI) or SOG-IS. These recommendations and the recommendations of other institutions and standardization organization are similar. The recommendations can be found in the technical guidelines TR 02102-1 and TR 02102-2[14] or SOG-IS Agreed Cryptographic Mechanisms[15].

5.1.1. Requirements on the DSC

The requirements provided for in *Annex I, section 3.2.2.* shall apply. Hence, it is strongly RECOMMENDED that Document Signers use the Elliptic Curve Digital Signature Algorithm (ECDSA) with NIST-p-256 (as defined in appendix D of FIPS PUB 186-4). Other elliptic curves are not supported. Due to the space restrictions of the digital green certificate, Member States SHOULD NOT use RSA-PSS, even if it is allowed as a fall back algorithm. In case that Member States use RSA-PSS, they SHOULD use a modulus size of 2048 or max. 3072 bit. SHA-256 SHALL be used as cryptographic hash function (see ISO/IEC 10118-3:2004).

5.1.2. Requirements on TLS, Upload and CSCA

For digital certificates and cryptographic signatures in the DCCG context, the major requirements on cryptographic algorithms and key length are summarized in the following table (as of 2021):

Signature Algorithm	**Key size**	**Hash function**
EC-DSA	Min. 250 Bit	SHA-2 with an output length ≥ 256 Bit
RSA-PSS (recommended padding) RSA-PKCS#1 v1.5 (legacy padding)	Min. 3000 Bit RSA Modulus (N) with a public exponent e > 2^16	SHA-2 with an output length ≥ 256 Bit
DSA	Min. 3000 Bit prime p, 250 Bit key q	SHA-2 with an output length ≥ 256 Bit

The recommended elliptic curve for EC-DSA is NIST-p-256 due to its widespread implementation.

5.2. CSCA certificate (NB_{CSCA})

The following table gives guidance on the NB_{CSCA} certificate template if a Member State decides to operate its own CSCA for the DCC system.

Bold entries are required (MUST be included in the certificate), *italic* entries are recommended (SHOULD be included). For absent fields, no recommendations are defined.

Field	Value

[14] BSI - Technical Guidelines TR-02102 (bund.de)
[15] SOG-IS - Supporting documents (sogis.eu)

Subject	**cn= <non-empty and unique common name>,** *o=<Provider>,* **c=<Member State operating the CSCA>**
Key usage	**certificate signing,** *CRL signing* (at minimum)
Basic Constraints	**CA = true, path length constraints = 0**

In accordance to *Annex I, section 7*, the subject name MUST be non-empty and unique within the specified country. The country code (c) MUST match the country that will use this CSCA. The certificate MUST contain a unique subject key identifier (SKI) according to RFC 5280.

5.3. Document Signer (DSC)

The following table provides guidance on the DSC certificate template in accordance to *Annex I, section 7*. **Bold** entries are required (MUST be included in the certificate), *italic* entries are recommended (SHOULD be included). For absent fields, no recommendations are defined.

Field	Value
Serial Number	**unique serial number**
Subject	**cn=<non-empty and unique common name>,** *o=<Provider>,* **c=<Member State that uses this DCS>**
Key Usage	**digital signature** (at minimum)

The DSC MUST be signed with the private key corresponding to a CSCA certificate that is used by the Member State.

The following extension are to be used in accordance to [*see also Annex I, Section 7*]:

- The certificate MUST contain a Authority Key Identifier (AKI) matching the Subject Key Identifier (SKI) of the issuing CSCA certificate
- The certificate SHOULD contain a unique Subject Key Identifier (in accordance to RFC 5280)

In addition, the certificate SHOULD contain the CRL distribution point extension pointing to the certificate revocation list (CRL) that is provided by the CSCA that issued the DSC.

The following extensions are to be used as defined in *Annex I, section 9.3*. Countries MAY also include an extendedKeyUsage entry with zero or more (i.e. up to 3) entries from:

Field	**Value**
extendedKeyUsage	1.3.6.1.4.1.0.1847.2021.1.1 for Test Issuers
extendedKeyUsage	1.3.6.1.4.1.0.1847.2021.1.2 for Vaccination Issuers
extendedKeyUsage	1.3.6.1.4.1.0.1847.2021.1.3 for Recovery Issuers

5.4. Upload Certificates (NBUP)

The following table provides guidance for the national backend upload certificate. **Bold** entries are required (MUST be included in the certificate), *italic* entries are recommended (SHOULD be included). For absent fields, no recommendations are defined.

Field	Value
Subject	**cn=<non-empty and unique common name>**, *o=<Provider>*, **c=<Member State that uses this upload certificate>**
Key Usage	**digital signature** (at minimum)

5.5. National Backend TLS Client Authentication (NB_{TLS})

The following table provides guidance for the national backend TLS client authentication certificate. **Bold** entries are required (MUST be included in the certificate), *italic* entries are recommended (SHOULD be included). For absent fields, no recommendations are defined.

Field	Value
Subject	**cn=<non-empty and unique common name>**, *o=<Provider>*, **c=<Member State on the NB>**
Key Usage	**digital signature** (at minimum)
Extended key usage	**client authentication (1.3.6.1.5.5.7.3.2)**

The certificate MAY also contain the extended key usage *server authentication (1.3.6.1.5.5.7.3.1)*, but it is not required.

5.6. Trust list signature certificate ($DCCG_{TA}$)

The following table defines the DCCG Trust Anchor certificate.

Field	Value
Subject	**cn=Digital Green Certificate Gateway, o=<Provider>, c=<country>**
Key Usage	**digital signature** (at minimum)

5.7. DCCG TLS Server certificates ($DCCG_{TLS}$)

The following table defines the DCCG TLS certificate.

Field	Value
Subject	**cn=<FQDN or IP address of the DCCG>, o=<Provider>, c= <country>**
SubjectAltName	**dnsName: <DCCG DNS name> or iPAddress: <DCCG IP address>**

Key Usage	**digital signature** (at minimum)
Extended Key usage	**server authentication (1.3.6.1.5.5.7.3.1)**

The certificate MAY also contain the extended key usage *client authentication (1.3.6.1.5.5.7.3.2)*, but it is not required.

The TLS certificate of the DCCG will be issued by a publicly trusted certificate authority (included in all major browsers and operating systems, following the CAB-Forum baseline requirements).