eHealth Network

Guidelines on

Technical Specifications
for Digital Green Certificates
Volume 2

European Digital Green Certificate Gateway

Version 1.3

2021-04-21

eHealth Network

This document provides a detailed architecture of the European Digital Green Certificate Gateway (DGCG)—the central broker of crypto material needed to realize EU-wide validation of Digital Green Certificates (DGC), comprising vaccination information, test results, and COVID-19 recovery details of EU citizens to facilitate safe travel, robust event management, and free movement across the EU. For more details on the DGC itself, please refer to

https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/covid-19-digital-green-certificates_en.

Adopted by the eHealth Network on 21 April 2021.

TABLE OF CONTENTS

eHealth Network

TABLE OF TABLES

TABLE OF TABLES

TABLE OF FIGURES

1   Introduction

1.1   Context

The context for this architecture is the vision of the EU Trust framework to establish a secure data exchange for public key material, validation information and value sets between member states.

1.2   Scope of Document

The scope of the document is to specify the Digital Green Certificate Gateway (DGCG) to support the EU trust framework concerning interoperability[1]. This consists of:

- Verification information (crypto material, references to trusted parties, etc.),
- Validation information (rules/patterns to differentiate valid and invalid certificates)

---

[1] https://ec.europa.eu/health/sites/health/files/ehealth/docs/vaccination-proof_interoperability-guidelines_en.pdf

2   Architecture Overview

As said before, DGCG is used to share validation and verification information across all national backend servers.

The following figure gives an overview of DGCG as specified in this document:
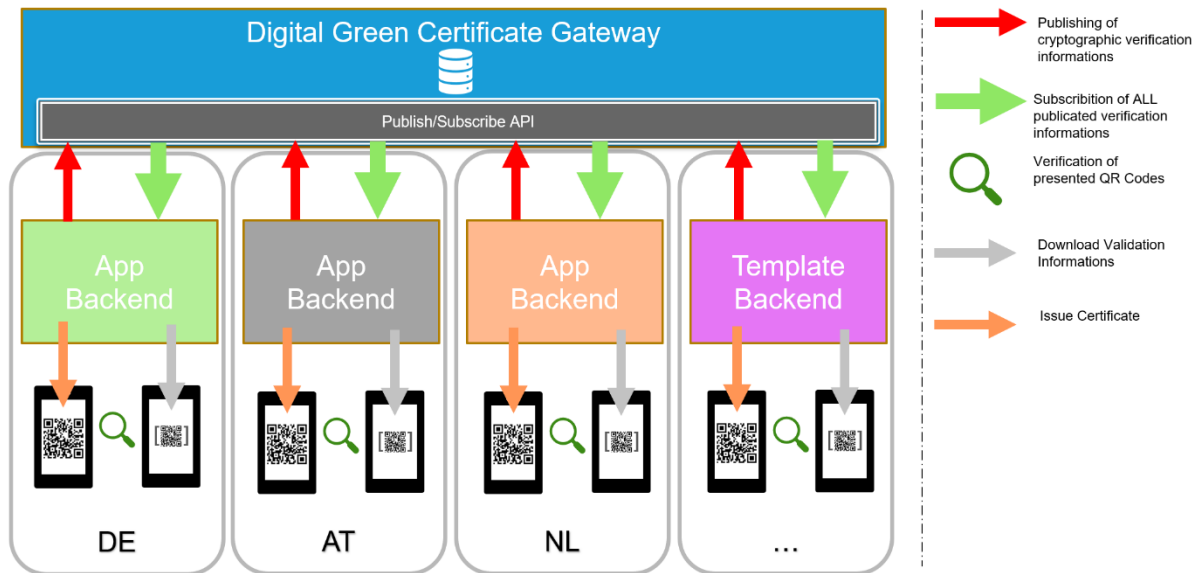

Figure 1: DGCG Overview

By using DGCG, backend-to-backend integration is facilitated, and countries can onboard incrementally, while the national backends retain flexibility and can control data processing of their users.
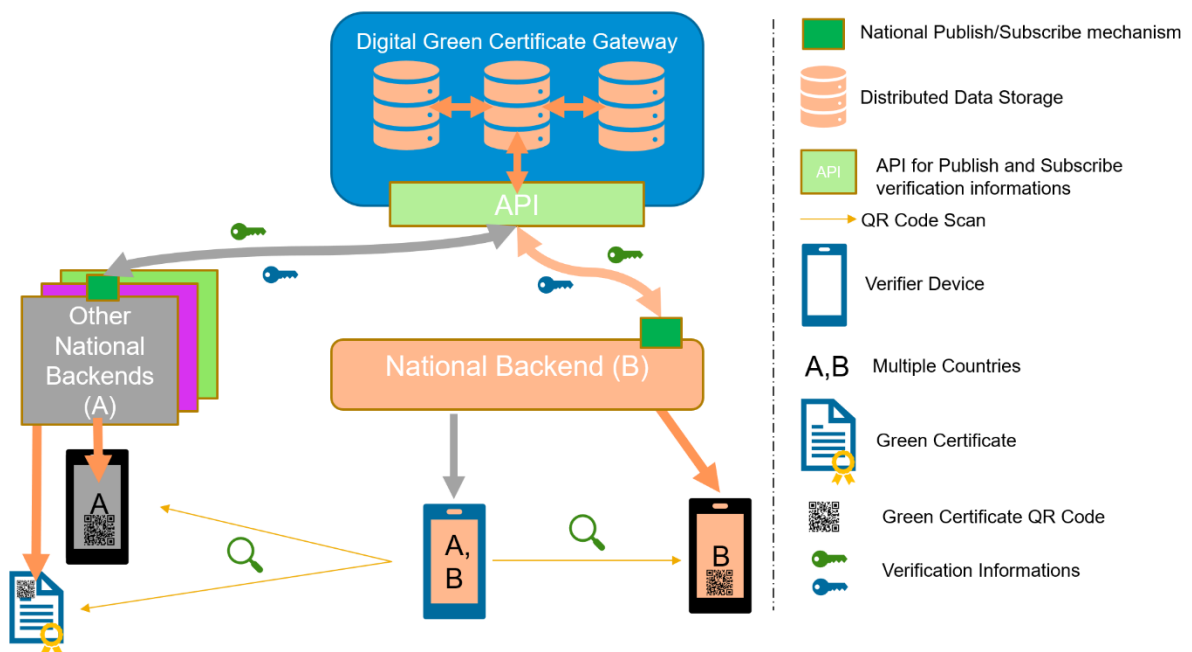

Figure 2: Autonomous National Backends

Each national backend is free to distribute the keys via any preferred technology to support the national verification devices in the best way. If the Digital Green Certificate is in a correctly formatted 2D code, each verifier device can verify each code from other countries, if the verifier is connected to the backend or if it has downloaded and stored the necessary public keys beforehand. (

2.1 Approach

The approach of this architecture is, as described before, to exchange different kinds of information to support the validation of the vaccination status, test result, or recovery status of a citizen (based on a test). For this purpose, the EU Trust Framework introduces a standardized signed CBOR data structure which is represented in a 2D Code. To validate this data structure in each country represented by different EU citizens, cryptographic public keys must be shared across the EU. The DGCG is designed to distribute such information easily to all member states and act as a Trust Anchor. This guarantees a reliable governance structure and trust into the public keys.

The entire architecture is inspired by the European Federation Gateway Service (EFGS), which is used to share GAEN diagnosis keys around the member states.

2.2 Assumptions

These are the main **assumptions** underlying this architecture:

1) For effective interoperability, each certificate must contain a string which uniquely identifies the issuing authority.
2) Each authority must pass an onboarding process, which identifies the authority uniquely by adding authentication, signing, issuer domain, CSCA and corresponding client certificates, as well contact persons and 24x7 support information's. Security of the onboarding process is vital for the security of the whole Digital Green Certificate framework. The CSCA certificates should be at least 2 years valid.
3) Each authority (there can be several per member state) provide endpoints for revocation and optionally callbacks, which are secured by mutual authentication. The required certificates are provided during onboarding and are whitelisted by the authorities by downloading the DGCG Trust Lists. The DGCG Trust List contains the public keys of all trusted parties, i.e., authorities who successfully completed the DGCG onboarding process.
4) Each 2D code relies on Key identifier, which allows each member state to check a scanned Green Certificate against the crypto material list received from the DGCG (issuer maps to Green Certificates) The crypto material should be uniquely identifiable. (described in the 2D Code Specification, with the first 8 bytes of a SHA256 cert hash)
5) All Digital Green Certificates and related crypto material are managed by the issuing authority.

2.3 Verification through the backend server

The national Backend servers of the Member State will provide access to the Public Keys according to national specifications, in particular to the entities that will perform verification of DGC using the relevant Public Keys (Trust lists). To this end the national backend servers will be kept up-to-date.

This will enable signature verification services and/or access to the public keys needed to perform verifications.

This feature will be implemented on the national backends level for instance through a secure API interface and SDKs. Such feature would be of particular relevance for airlines.

3   Communication

3.1   Triangle of Trust

The triangle of trust describes the relationship between a holder, issuer, and verifier of credentials in a distributed system. This setup increases data privacy, because the holder is the only actor who has a direct connection to the verifier and the issuer, i.e., the issuer doesn't know where the holder presents their credentials and the issuer doesn't know who trusts him.
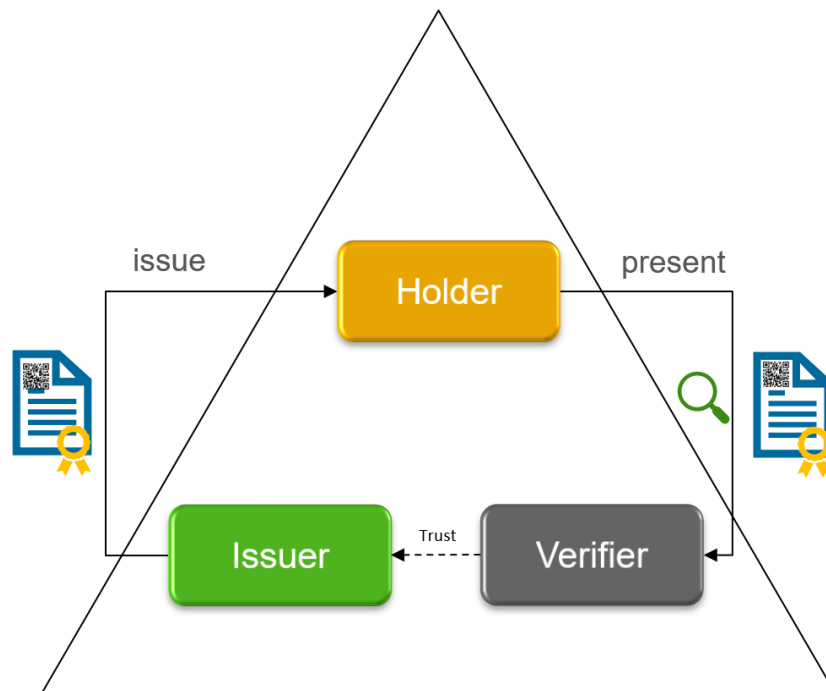
Figure 3: Triangle of Trust

The triangle of trust is the blueprint for Green Certificate interoperability:

- **Holder**: A Green Certificate (DGC) owner (i.e., a citizen with a vaccination, test result, or recovery status (based on a positive test result) - note that the Digital Green Certificate can be held digitally within a wallet app *or* on paper (or both)
- **Issuer**: A national authority
- **Verifier**: A verifier (e.g., customs officers, police, or hotel staff)

But there is an important question: How does the verifier know which issuer is trustworthy? In a personal relationship, one would decide by experience. In this architecture, the DGCG tells the verifier which issuers are trustworthy by providing cryptographically anchored information.
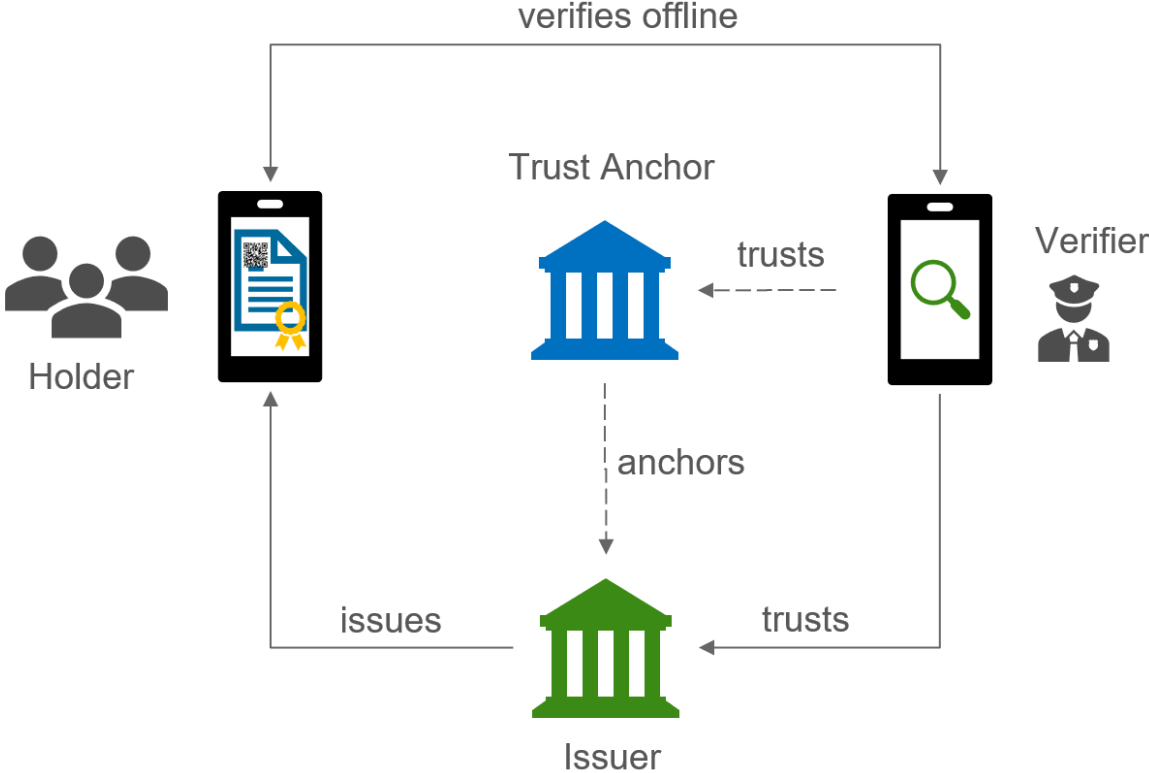
Figure 4: Trust Anchoring

3.2 Distribution of Verification Information

Exactly how each national app communicates with the corresponding national backend - whether via CDN, active push, or otherwise - is left to each country. Important here is the cryptographically secured E2E protection between the member states.
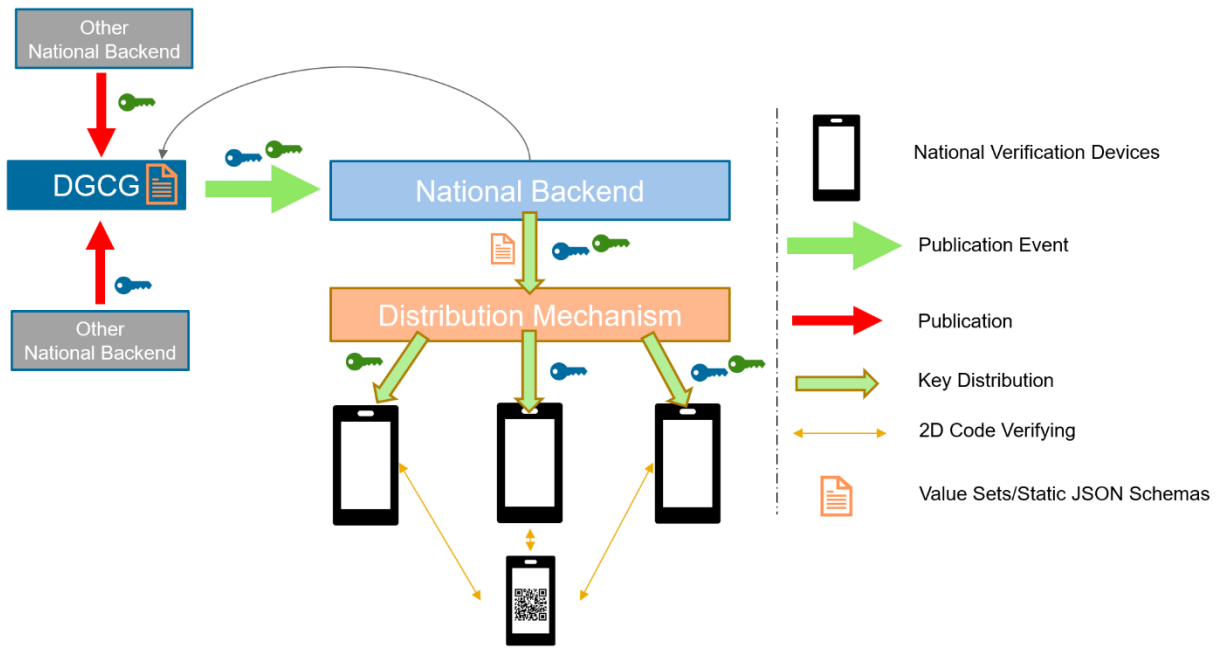
Figure 5: Distribution of Signing and Validation Information

3.3 Device-to-Device Communication

Device-to-device communication is built on a standardized 2D code and verifier format defined by the EU Trust Framework.
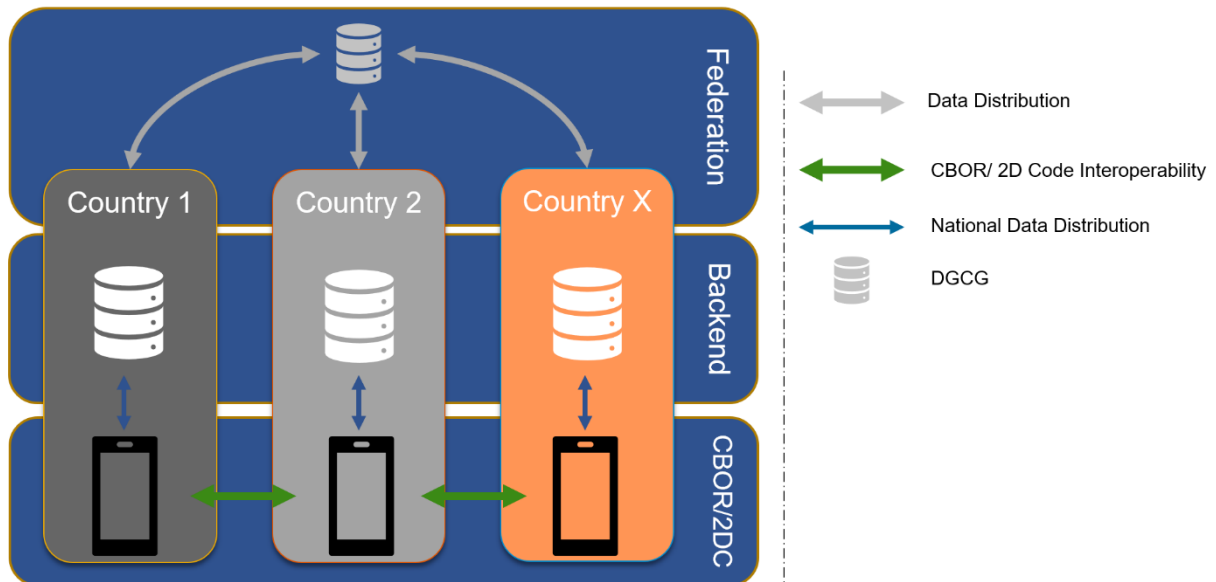
Figure 6: Distribution of Verification Information

3.4 Backend-to-Backend Communication

A direct backend-to-backend communication is not necessary, because the main purpose of the DGCG solution is to provide verification information. All participating national backends will provide that information to DGCG, which in turn stores the information and provides them for download. Nevertheless, bilateral communication between national backends is not categorically excluded—in a Self-Sovereign Identity scenario it is explicitly desired in the future.
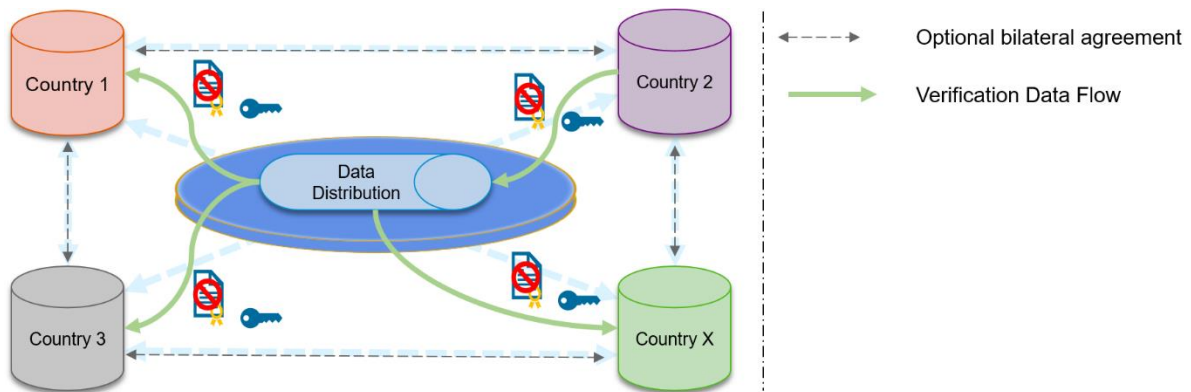
Figure 7: Indirect Backend-to-Backend Communication

3.5 Trust Establishment

To ensure that just data from trusted parties are accepted. The system contains a trust list which is signed entry by entry air gapped by an official signer. This signer, signs with his private key each request of onboarding and provides this signed information to the DGCG operator which can set this entry on the trust list. This guarantees that no external attacker or another party than the trusted signer can create valid records for the trust list. The public key of the trusted signer is shared out of band to the other parties, to establish an effective trust anchoring.

The trust list contains entries for authentication, signing of data packages, CSCAs etc.
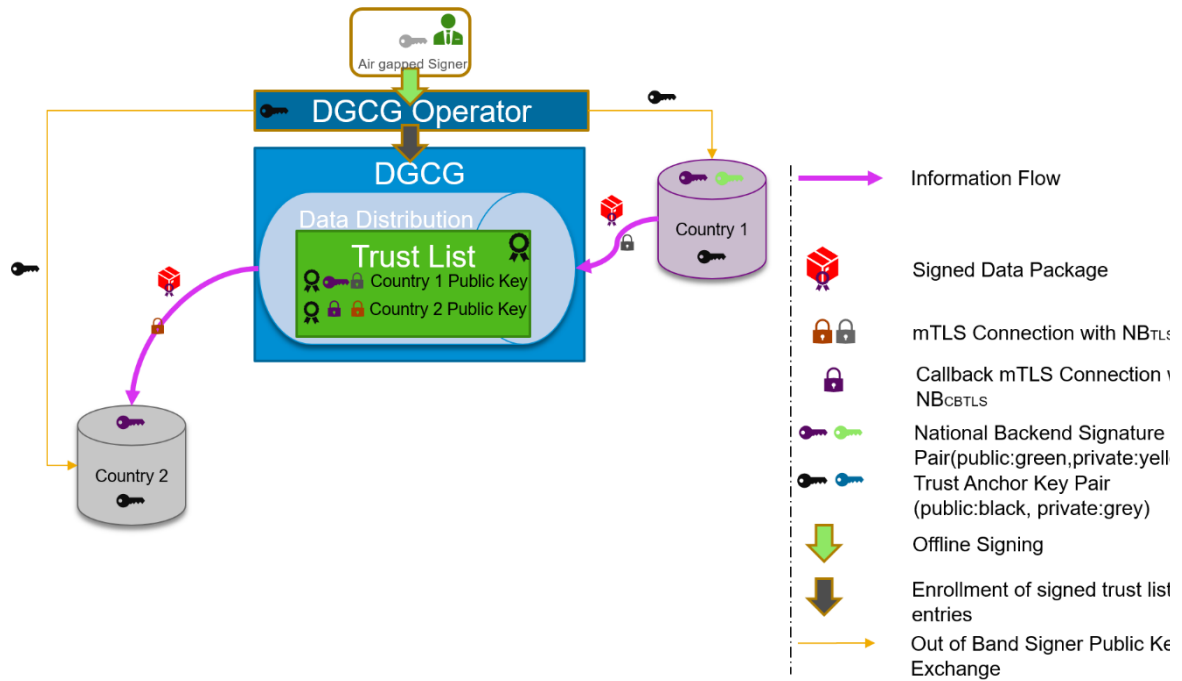
Figure 8: Trusted Data Exchange (Country 1 to Country 2)

To create the Trust List, an onboarding process is established, which onboards attendees by checking different criteria. If the process is successful, all certificates for connection and uploading DSCs (Document Signer Certificates) are anchored in the Trust List.
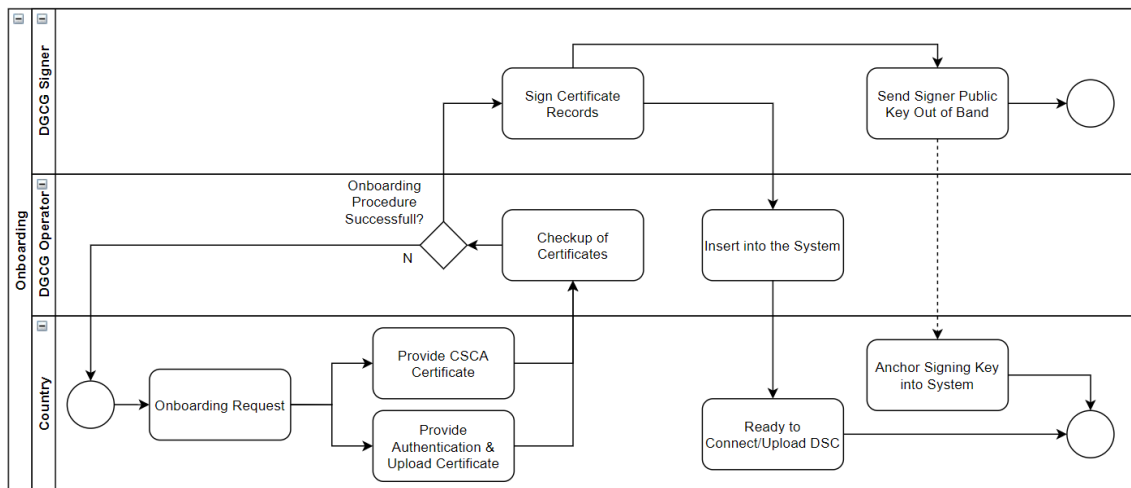
Figure 9: Onboarding Process

eHealth Network

4 Data Structures

4.1 Data Types

4.1.1 JSON Schemas

All JSON objects are based on a JSON schema which are hosted in the DIGIT operator's environment. All schemas are based on the JSON-Schema.org syntax (https://json-schema.org/specification.html)

4.1.2 Static Data Content

The exchanged QR codes are based on a generated CBOR, which contains the minimal datasets of a certificate described by the Regulation on the Digital Green Certificate and further defined by the eHealth Network[2]. The schemas will be hosted under /schema/[schemaName]/[version].json in the DGCG operating environment.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema", //example
  "$id": " https://ec.europa.eu/health/dgc/schema/dgc/100.json",
  "title": "Digital Green Certificate",
  "description": "Interoperability Exchange format",
  "type": "object",
  "properties": {
    "person": {
      "description": "The legal name of a person.",
      "type": "string"
properties:{
                        …
            }
    },
     …
    {
iss:{
        …
    }
    }
  },
  "required": [ "person","certificateMetadata","kid",…]
}
```

Figure 10: Example JSON Schema for the Minimal Dataset

Value sets are hosted as well under the DGCG environment with the pattern /valueset/[setname]/[version].json

---

[2] See https://ec.europa.eu/health/sites/health/files/ehealth/docs/vaccination-proof_interoperability-guidelines_en.pdf and https://ec.europa.eu/health/sites/health/files/ehealth/docs/citizen_recovery-interoperable-certificates_en.pdf

```
{
  fieldname:"aut",
  values:["AstraZeneca AB", "Biontech Manufacturing GmbH",…]
}
```

Figure 11: Value Set Schema

Column mappings are hosted under /mapping/[name]/[version].json.

```
[
 { name:"aut",
  value:"Authorization Holder/Manufacturer"},
 {
   …
 }
}
```

Figure 12: Mapping Content

⚠ DGCG hosting URL to be clarified.

4.1.3   Discovery Document

To build for the 2D code verification a context about the used schemas and value sets inside of the code, a discovery document is placed on the DGCG to provide this information. The structure is defined as the following example:

```
[
  {
    "version":"HC1", //placed in the 2D Code
    "type": "Vaccination", //Recovery, Test,…
    "description": "This is the first context for 2D Generation (string)
    "schema": „http://dgcg.tech.eu.europe.com/schema/dgc/100.json",
    "valuesets":
["http://dgcg.tech.eu.europe.com/valueset/vacmanufacturer/100.json,
 …]
    "mapping":" http://dgcg.tech.eu.europe.com/mapping/vacdisplay/100.json"
  },
  {
    "version":"HC2",
    "type":"Recovery",
     ..
  }
]
```

Figure 13: Discovery Document

⚠ For 2D code generation, the latest version must be used.

4.1.4 CMS Signed Data

To ensure the end-to-end secure data transfer of certificates, the cryptographic message standard is used. This Cryptographic Message Standard is defined in RFC5652[3]. Each information uploaded by the member states must be digital signed and embedded into this container format. This guarantees from the issuer to the receiving backend a secure data exchange. These packages are sent in Base64 format described in RFC4648[4]. over an HTTP Interface which uses the media type **application/cms** described in RFC7193[5].

The package content contains the signer certificates encoded as raw byte array.

4.2 Data Storage

Uploaded verification information is permanently stored because of the long lifetime of existing Green Certificates. The verification information must be explicitly deleted by the controlling member states if it's no longer needed.

4.2.1 Database Requirements

As for database technology, we gathered the following requirements:

Requirement	Explanation
Object Storage	The database must support storage of different objects without needing schema changes
Strong Consistency	The database must support strong consistency, i.e., new data is fully replicated after each transaction
High Availability	We need redundant, replicated storage to avoid data loss

Table 1: Database Requirements

According to the CAP theorem for distributed data storage, only two of the three requirements *consistency*, *availability*, and *partition tolerance* can be fully met at the same time. Partition tolerance refers to resilience against message loss across the network. Since consistency and availability provide the greatest value for the national backends—and since both the number of partitions and message loss rate will be small—we focus on the first two requirements.

4.2.2 Database

According to the requirements and the given environment within the DIGIT, a relational MySQL Database has been chosen.

---

[3] https://tools.ietf.org/html/rfc5652
[4] https://tools.ietf.org/html/rfc4648
[5] https://tools.ietf.org/html/rfc7193

4.2.3 Data Format

4.2.3.1 Trusted Party Table

Field	Description	Data Type
Id	Primary key	Long
Timestamp	Timestamp of the Record	Timestamp
Country	Country Code	varchar(2)
Sha256Fingerprint	SHA256-fingerprint of the certificate	varchar(*)
Certificate Type	Type of the certificate (Authentication, Signing, Issuer, Client, CSCA)	varchar(*)
RawData	Raw Data of the certificate	binary
Signature	Signature of the Trust Anchor	varchar(*)

Table 2: Trusted Party Table

Certificate Types:

Type	Description
Authentication	Certificate which the member state is using to authenticate at DGCG (NBTLS)
Upload	Certificate which the member state is using to sign the uploaded information's (NBUS)
CSCA	Country Signing Certificate Authority certificate (NBCSCA)

Table 3: Certificate Types

⚠ The DGCG will add client certificates as well (country: EU).

⚠ The DGCG trust anchor certificate is sent out-of-band to the member states to anchor the public key trust network.

4.2.3.2 Signer Information Table

Field	Description	Data Type
Id	Primary key	Long
Timestamp	Timestamp of the Record	Timestamp
Country	Country Code	varchar(2)
Sha256Fingerprint	SHA256-fingerprint of the certificate	varchar(*)
Certificate Type	Type of the certificate (Authentication, Signing, Issuer, Client, CSCA)	varchar(*)
RawData	Raw Data of the certificate	binary
Signature	Signature of the Uploader	varchar(*)

Table 4: Signer Information Table

Certificate Types:

Type	Description
DSC	Certificate which the member state is using to sign documents (NBDSC)

4.2.3.3 Audit Event Table

Field	Description	Data Type
Id	Primary key	Long
Timestamp	Timestamp of the Record	Timestamp
Country	Country Code	varchar(2)
Uploader Sha256Fingerprint	SHA256-fingerprint of the certificate	varchar(*)
Authentication Sha256Fingerprint	SHA256-fingerprint of the certificate	
Event	Event which occurs	varchar(*)
Description	Description of the Event	varchar(*)

Table 5: Audit Table

4.2.3.4 User Rights

Role	Table	Grants	Actor
Admin	Signer Information	Full Access	DGCG Operator
Admin	Trusted Party Table	Full Access	DGCG Operator
Admin	Audit Table	Full Access	DGCG Operator
User	Signer Information	Read/Write/Delete	DGCG Application
User	Audit Table	Write	DGCG Application
User	Trusted Party Table	Read	DGCG Application

Table 6: Signer Information Table

5 Building Block Schema

In a rough overview, DGCG is built on top of Tomcat, which is directly linked to a F5 load balancer, combined with a reverse proxy. The main modules are for upload, download, and revoke. A trust list API provides additionally all anchored party information's.
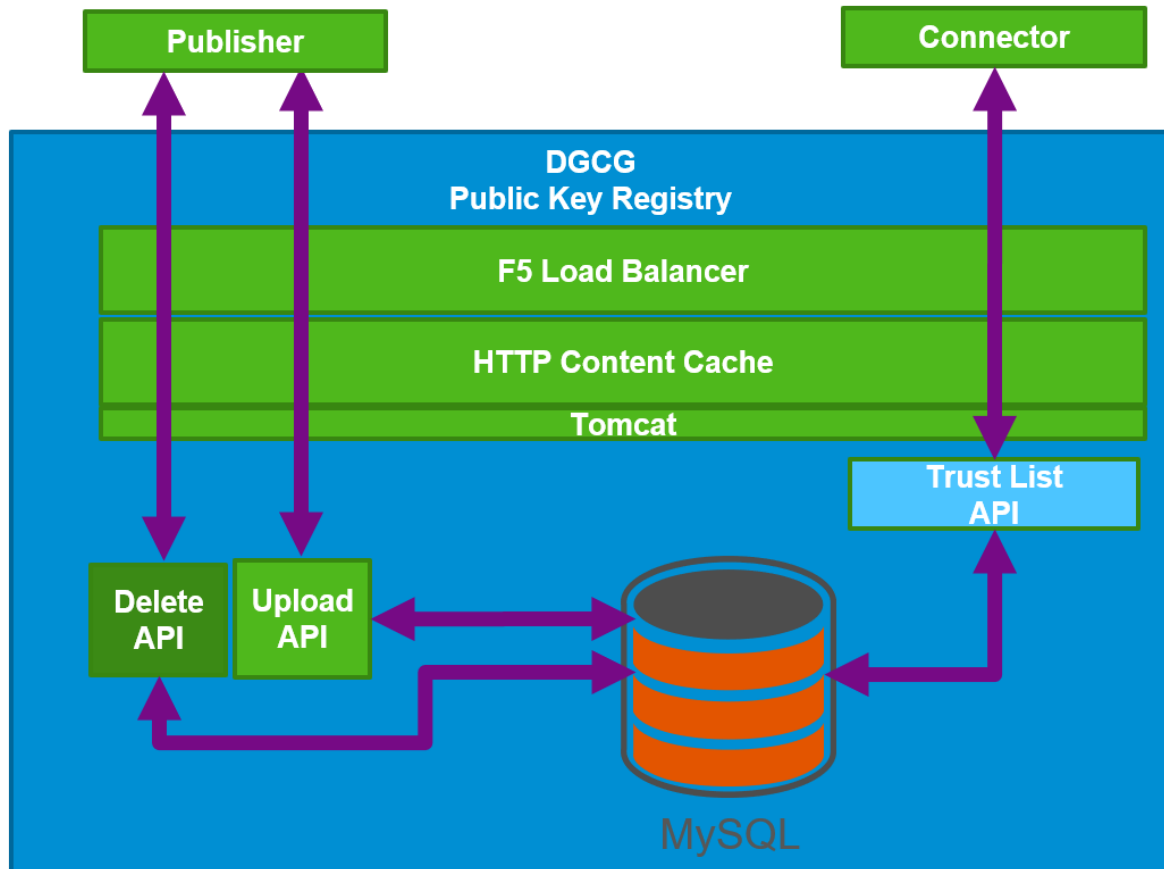

Figure 14: Building Block Schema

⚠️ DGCG performs mutual authentication with the national backends—its API validates the provided server certificate of the national backend and proves its identity by way of a client certificate to them. Each national backend has to explicitly whitelist DGCG's client certificate and has to provide its server certificate's public key to the DGCG for whitelisting.

6   Interfaces

6.1   Overview

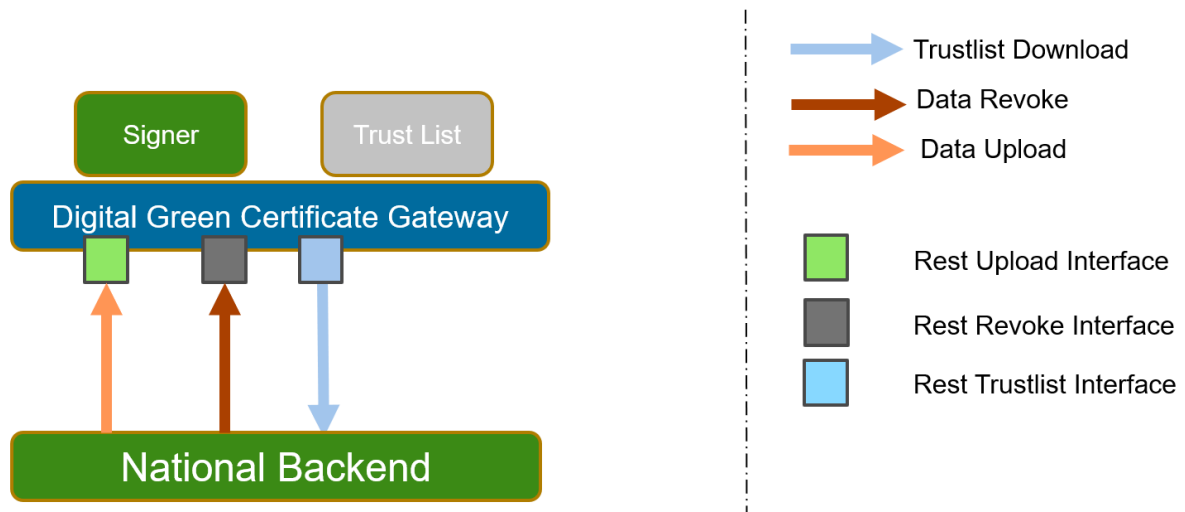DGCG provides a simple REST API with common upload and download functionality for trusted information.

Figure 15: API Overview

The functions of the interfaces are:

- Upload/Revoke of signer information's, and
- Download Trust List information's

All these actions are represented by GET, POST, and DELETE actions.

For detailed description of REST interfaces, we rely on the Open API Specification 3.0[6]. This allows a comprehensive human-readable and machine-readable representation of all aspects of the defined interface.

We define access points according to the following scheme:

---

[6] https://swagger.io/specification/

Figure 16: Interface Definition Overview

⚠ The Digital Green Certificate Gateway Service API performs no signing of data packages. Each national backend needs to format and sign the data by itself.

6.2 MIME Type

The packages for Upload and Delete interfaces are sent in Base64 format described in RFC4648[7]. The used MIME Type is

application/cms

described in RFC7193[8].

Usage:

```
Content-Type: application/cms
Content-Transfer-Encoding: base64
```

6.3 Upload Interface

6.3.1 Overview

The upload interface consists of one POST request (in version 1.0), to upload information about the verification. The registration of the certificate takes max 10 seconds. After this, the successful registration can be verified by fetching the list of valid certificates and ensuring that the uploaded certificate is on the list.

---

[7] https://tools.ietf.org/html/rfc4648
[8] https://tools.ietf.org/html/rfc7193

Figure 17: Upload Interface

6.3.2 Parameters

Figure 18: Upload Parameters

6.3.3 Responses

Code	Description
201	Signer Information was created sucessfully.
401	Unauthorized. No Access to the system. (Client Certificate not present or whitelisted)

Media type
application/json

Example Value | Schema

```json
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

| 403 | Forbidden. Signer Information package is not accepted. (hash Value or signature wrong, client certificate matches not to the signer of the package) |

Media type
application/json

Example Value | Schema

```json
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

| 406 | Content is not acceptable. (Wrong Format, no CMS, not the correct signing etc.) |

Media type
application/json

Example Value | Schema

```json
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

| 409 | Conflict. Choosen UUID is already used. Please choose another one. |

Figure 19: Upload Responses

6.3.4 Transmission Protocol

During the upload, the uploader identity is extracted from the client certificate. If the client certificate is valid, the submitted content is validated and stored in the database.
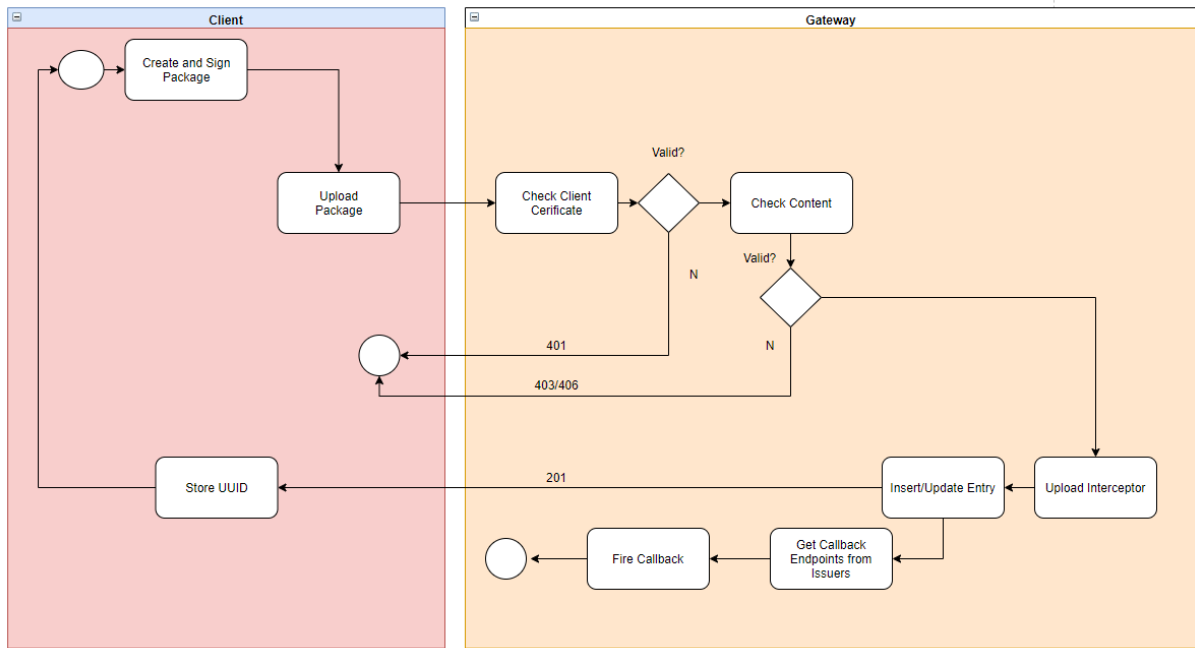
NaNeHealth Network

Figure 20: Upload Transmission Process

Content Checks:

	Check	Description
1	Uploader Certificate	Does the used Uploader Certificate match to the registered TLS Client certificate for the Country?
2	CSCA Checkup	Does the contained certificates match to the registered CSCA(s)?
3	Already Exist Check	Is there still an existing certificate with the same SHA-256 Fingerprint? If yes, write an entry in the audit log.

6.4 Signer Delete Interface

6.4.1 Overview

The upload interface contains a DELETE request to delete signer information. The deletion of the certificate takes max 10 seconds. After this, the deletion can be verified by fetching the list of valid certificates and ensuring that the deleted certificate is not included on the list.

Figure 21: Delete Interface

6.4.2 Parameters

The request requires a single header parameter, X-RECORD-UUID (SHA256-fingerprint of the cert):

Name	Description
X-RECORD-UUID string *(header)*	XJ32-SKFJ3-KJKFJR-KDJEJ4

Figure 22: Delete Parameters

6.4.3   Responses

Responses

Code	Description	Links
200	OK.	*No links*
401	Unauthorized. No Access to the system. (Client Certificate not present or whitelisted) Media type application/json Example Value | Schema	*No links*

```
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

403	Forbidden call in cause of invalid client certificate for this resource. Media type application/json Example Value | Schema	*No links*

```
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

410	Object is permanently gone. Media type application/json Example Value | Schema	*No links*

```
{
  "Code": "0x001",
  "Problem": "[PROBLEM]",
  "Sent Value": "[Sent Value]",
  "Details": "..."
}
```

Figure 23: Delete Response
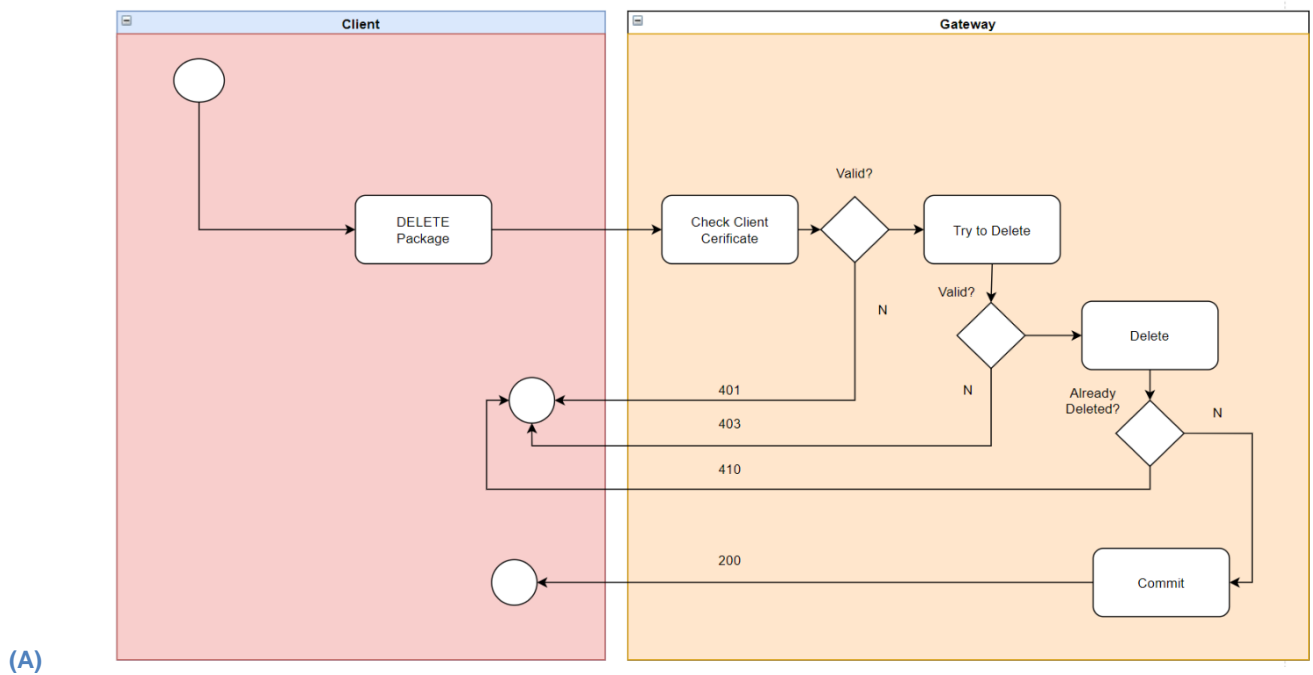
6.4.4   Transmission Protocol

Figure 24: Delete Flow

6.5   Trust List Interface

6.5.1   Overview

The trust interface consists three GET requests to retrieve the DGCG Trust Lists.

Figure 25: Trust List Interface

⚠ The trust list provides an aggregated view on a given query, not on the uploaded batch of certificates.

6.5.2 Parameters

Figure 26: Trust List Interface

ℹ️ Depending on the Route, the two parameters can be optional or are not necessary.

6.5.3 Responses

The Trust List calls return an array of trusted entries which contains the SHA256 fingerprint of the certificate and the related information about the country, certificate type and the BASE64 encoded raw data of the certificate itself.

⚠️ The signature is calculated over the rawdata and can be checked against the trust list.

NOTE: The Type Authentication, Upload and CSCA can just be verified over the Public Key Certificate which was received out of band during the onboarding.

The Type DSC signature can be validated against the Upload certificate of the country and against the onboarded CSCAs on the trust list.

Figure 27: Trust List Responses

ℹ The "RawData" Field contains an X509 certificate, uploaded over signerCertifcate upload.

ℹ The Signature is calculated over the byte presentation of the raw data.

7 Technology Choice

Note: We selected the technologies below from a restricted set of alternatives provided by the designated platform operator (DIGIT[9] in Luxemburg).

Component	Technology	Core Features
REST API	Java / Spring Boot	Powerful, versatile web framework
Database	MySQL	Supports JSON documents
Load Balancer	F5	Reverse proxy, load balancing, detailed traffic metrics, SSL offloading, Client Auth
Web Server	Tomcat	

Table 7: Technology Choice

---

[9] https://ec.europa.eu/info/departments/informatics_en

8 Implementation Roadmap

Feature	Expected Version
Onboarding	1.0
mTLS/Signed Data Upload	1.0
Download/Upload of Signer Certificates	1.0
Revoke of uploaded Signer Certificate	1.0
Static Data Content	1.0
Onboarding of CSCA Certificates	1.0
Check of Signer Uploads against CSCA Certificates	1.0
Trust lists	1.0
Callbacks for Changes	TBD (candidate for 1.0)
Upload of Validation Schemas	TBD
Validation and Revocation Rules	TBD

Table 8: Roadmap

9 Glossary

Term	Description
2D code / QR code	Two-dimensional barcode
Certificate	Technical Certificate like X509.
Certificate issuance	The act of creating a Digital Green Certificate
Civil identity	Defined by the eHN Minimal Data Set: Person Name (The legal name of the vaccinated person (surname(s) and forename(s) in that order), Person Date of birth
CMS	Cryptographic Message Syntax, RFC 5652
Crypto Material / Cryptographic material	All material, including documents, devices, or equipment that contains cryptographic information and is essential to the encryption, decryption, or authentication of telecommunications
DGC	Digital Green Certificate. Includes also the machine processable part of the according paper document. Currently, it can be differentiated between three different types: vaccination, test result, recovery certificate (based on a positive test result)
DGC Gateway / EU Digital Green Certificate Gateway	Exchange of public keys, certificates, and other crypto material between national backends
DGCI	Digital Green Certificate Identifier. Universally unique ID assigned to the DGC when issued by the issuer app
Holder / DGC Owner	Person in possession of a Digital Green Certificate
Holder verification / verification	The process of verification to answer the question, if a person who states to be the legal holder of a Digital Green Certificate is the same person who holds the certificate
ICAO	International Civil Aviation Organization is a specialized agency of the United Nations
Issuer	A person or system that works in behalf of an Issuing (health) authority to issue Digital Green Certificates
Issuer App / Issuer application frontend	The application that is used by the issuer to issue Digital Green Certificates. The issuer application frontend provides a user interface that is used by the issuer to enter the necessary data. The application will communicate with the backend / dgca-issuer-service to implement the process of digital green certificates signing. *Component: dgca-issuer-web*
Issuing authority	Institutions named by a member state to act in its will for issuing digital green certificates
JSON	JavaScript Object Notation is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value)
Key pair	Public and private key in the context of asymmetric cryptography

Member states / MS	Member state of the European Union (currently 27) or a third country in scope of the system (e.g. based on an adequacy decision)
National certificate backend / backend	The issuer backend is accessed by the issuer frontend application and the respective wallet apps. The backend itself publishes its public keys to the DGCG where they can be distributed to other MS. Each MS hosts its own issuer backend. DGC signing, TAN generation, TAN validation, public key publication, and digital green certificate validation/revocation inf, etc.
NTP server	The Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.
Offline verification	Process of verification without the need of an active internet connection during the time of verification. Attention: offline verification will make use of an internet connection that downloads necessary crypto material in advance.
Onboarding	The structured process of including and gaining interoperability on a technical and organizational level of a member state to issue and verify digital green certificates
Second factor / 2FA	Authentication method in which a computer user is granted access only after successfully presenting two pieces of evidence (or factors) to an authentication mechanism. E.g., a transaction authentication number.
Service X	Sample service, e.g., in a booking process
SMS	Short Message Service. Text messaging service component
TAN	Transaction authentication number
Trust Anchor	An authoritative entity for which trust is assumed and not derived
Trust Lists / CTL	A predefined list of items signed by a trusted entity
Verifier	Person that verifies DGCs
Verifier App	verifies digital green certificates with the help of the dgca-verifier-service. Consists of a frontend (to be used by the verifier) and a backend (providing trusted key for a member state). *Component: dgca-verifier-app*
Wallet App	Application that holds a digital green certificate and provides a frontend to be used by the holder. *Component: dgca-wallet-app*

APPENDIX

(A) Authentication:
- Connections to DGCG will be over HTTP using TLS with cryptographic settings that meet or exceed the relevant ENISA recommendations on algorithms, key sizes, and parameters.
- There will be mutually authenticated TLS (mTLS) connections between DGCG and each national backend.
- Trust validation happens by means of certificate validation based on TLS client, TLS server certificate, and server name (CN/subjectOtherName, according the CAB forum standard).
- This is combined with pinning by both parties on explicit certificate, a dedicated CA in the chain, or issuing CA by DGCG.
- For this reason, each national backend will inform DGCG of the certificate in the chain below which they consider any client certificate as being authorized by the national backend to connect to DGCG on their behalf.
- In the most extreme case, this may be just the actual leaf certificate or a self-signed certificate. In general, implementers are urged to provide a (dedicated) CA certificate as to minimize operational logistics (from the perspective of the DGCG operator) around key rollover, revocation, and general long-term certificate management.
- The operator of DGCG will communicate the certificate in the chain below which they consider any server certificate as being appropriate for DGCG.

(B) Operational and Runtime Considerations:
- The national backends will communicate a contact point for operational matters if such is not readily evident from the certificate.
- The operator of DGCG will communicate the certificates used by each national backend to all other national backends.