SWISS COVID CERT APP

ARCHITEKTUR-DOKUMENTATION

Kompakte, bewertungsorientierte Dokumentation für Unterricht und Präsentation

VERSION: 1.1
DATUM: 05,11.2025
AUTOREN: Frédéric Hofer | Hasan Balci | Simon Gemetti | Marcel Spahr
REPOSITORY: https://github.com/simongemetti/Gruppe-Covid-App

SWISS COVID CERT APP, ARCHITEKTURDOKUMENTATION (arc42)

Kompakte, bewertungsorientierte Dokumentation für Unterricht und Präsentation

VERSION: 1.1

DATUM: 08.11.2025

AUTOREN: Frédéric Hofer | Hasan Balci | Simon Gemetti | Marcel Spahr

REPOSITORY: https://github.com/simongemetti/Gruppe-Covid-App

INHALTSVERZEICHNIS

1 EINFÜHRUNG UND ZIELE: Worum geht es

1.1 Kurzbeschreibung der Fachanwendung: Swiss Covid Cert App in einem Satz

Die Swiss Covid Cert App speichert signierte COVID-Zertifikate lokal auf dem Smartphone und zeigt sie bei Bedarf offline an; die Gültigkeit wird anhand kryptografischer Signaturen und einer vertrauenswürdigen Schlüsselliste (Trustlist) geprüft.

1.2 Motivation und Nutzen: Warum dieses System sinnvoll ist

- Schneller, digitaler Nachweis von Impf-/Test-/Genesungsstatus.
- Datenschutzfreundlich: Keine zentrale Speicherung von Personendaten.
- Interoperabel mit EU-weiten Zertifikaten (DCC-Format); praxistauglich an Einlässen und Grenzen.

1.3 Stakeholder und Zielgruppen: Wer profitiert

- Bürgerinnen und Bürger (Endnutzerinnen und Endnutzer).
- Veranstalter, Betriebe, Grenz- und Zutrittskontrollen (nutzen separate prüf-App).
- Behörden (BAG/FOITT) für Schlüsselverwaltung und Trustlists.
- App-Store-Plattformen für Distribution und Updates.

1.4 Top-Qualitätsziele: Die drei wichtigsten Ziele

- Sicherheit und Datenschutz: Signaturprüfung, minimale Datenteilung, kein Tracking.
- Verfügbarkeit und Offline-Fähigkeit: Anzeige und lokale Prüfung funktionieren auch ohne Netz.
- Performance und UX: QR-Scan bis Statusanzeige Median ≤ 0.5 s, klare Anzeige, barrierearme Bedienung.

1.5 Randbedingungen und Annahmen: Rahmen, Standards, Scope

- Standards: EU DCC (CBOR/COSE), QR-Codes, TLS 1.2+.
- Plattformen: iOS und Android; Nutzung von Secure Storage (Keychain/Keystore) und Kamera-Zugriff.
- Datenmodell: Keine Übermittlung von Personendaten an Backend-Dienste durch die Wallet-App.
- Annahmen für diese Dokumentation: Fokus auf Lernzweck; reale Systeme dienen als Referenzmodell.

1.6 Technische und organisatorische Rahmenbedingungen

Die Architektur basiert auf einer klaren Trennung zwischen lokalen und serverseitigen Komponenten.
Die Swiss Covid Cert App funktioniert vollständig offline und nutzt das Backend nur für den Abruf von Trustlisten (öffentliche Schlüssel). Der Datenaustausch erfolgt ausschliesslich
über signierte, verschlüsselte HTTPS-Verbindungen. Personendaten werden zu keinem Zeitpunkt
auf Servern gespeichert. Die Anwendung folgt damit dem Prinzip „Privacy by Design“ und erfüllt
die Anforderungen des schweizerischen Datenschutzgesetzes (DSG) sowie der EU-Datenschutzrichtlinie (DSGVO)

Diese Architekturdokumentation beschreibt Aufbau, technische Struktur und Qualitätsanforderungen der Swiss Covid Cert App auf Basis des Dokumentationsstandards arc42.

Die Swiss Covid Cert App wurde als datenschutzfreundliche, mobile Anwendung konzipiert,
welche COVID-Zertifikate lokal auf dem Smartphone speichert und bei Bedarf offline anzeigen kann.
Die App überprüft die Gültigkeit eines Zertifikats durch kryptografische Signaturprüfung anhand einer vertrauenswürdigen Schlüsselliste (Trustlist), die regelmässig vom FOITT/BAG-Backend über eine signierte HTTPS-Verbindung bezogen wird.
Personendaten werden dabei zu keinem Zeitpunkt an Server übermittelt („Privacy by Design“).

Die Architektur ist modular aufgebaut und besteht ausfolgenden Hauptkomponenten:
- Wallet App (Benutzergerät): Verwaltung und Anzeige von Zertifikaten, Offline-Validierung.
- FOITT/BAG Backend: Bereitstellung der signierten Trustlist und Verwaltung der öffentlichen Schlüssel.
- Covid Certificate Check App: Visuelle, offline arbeitende Prüf-App zur Validierung der QR-Codes.
- App Stores (iOS & Google Play): Bereitstellung und Aktualisierung der App.
- EU DCC Gateway: Austausch öffentlicher Vertrauensanker zwischen der Schweiz und der EU.

Die Systemarchitektur gewährleistet höchste Sicherheit und Datenschutz durch lokale Datenhaltung, signierte Trustlisten und Ende-zu-Ende-Verschlüsselung. Die App ist vollständig offline funktionsfähig, was sowohl Benutzerfreundlichkeit als auch Verfügbarkeit sicherstellt.

Im Rahmen dieser Dokumentation werden zudem drei zentrale Qualitätsanforderungen betrachtet:
1. Sicherheit und Datenschutz, Schutz der Nutzerdaten, Integrität und Signaturprüfung.
2. Verfügbarkeit und Offline-Fähigkeit, Kernfunktionen bleiben auch ohne Internet verfügbar.
3. Performance und Benutzerfreundlichkeit, QR-Scan und Validierung erfolgen innerhalb von < 0.5 Sekunden.

Die erstellten Diagramme (Kontext-, Use-Case- und Verteilungsdiagramm) veranschaulichen
das Zusammenspiel aller Systemteile, Akteure und Kommunikationswege. Die Architektur ist technisch realisierbar, konform zu UML- und arc42-Standards und erfüllt die Anforderungen an eine sichere, wartbare und verständliche Softwarelösung.

Diese Dokumentation dient als Grundlage zur Beurteilung der Softwarearchitektur und als Leitfaden für zukünftige Weiterentwicklungen der Swiss Covid Cert App im Rahmen von IT-Sicherheits- Datenschutz- und Qualitätsmanagementprojekten.

1.7 Ziel und Nutzen dieser Architekturdokumentation

Ziel dieser Dokumentation ist es, die Softwarearchitektur der Swiss Covid Cert App klar, verständlich und bewertbar darzustellen. Sie dient als Grundlage für technische, sicherheitsrelevante und organisatorische Entscheidungen und folgt dem international anerkannten Dokumentationsstandard arc42.

Der Nutzen liegt in der strukturierten Darstellung der Systemgrenzen, Datenflüsse, Anwendungsfälle und Kommunikationsbeziehungen. Durch die Anwendung der arc42-Struktur
können sowohl technische als auch fachliche Stakeholder (BAG, FOITT, Entwickler, Tester)
die Architektur schnell verstehen und bewerten.

2 KONTEXT UND ABGRENZUNG: Systemumfeld und Schnittstellen

2.1 Fachlicher Kontext: Akteure und Umsysteme

Akteure:
- Benutzer/in
- Support/Helpdesk (optional)

Umsysteme:
- FOITT/BAG Issuance und Trustlist (Ausstellung, Schlüssel, Sperrlisten)
- EU DCC Gateway (Trust Anchor Austausch)

- iOS App Store und Google Play (Distribution und Updates)
- Covid Certificate Check App (separate Prüf-App beim Einlass)
- OS-Dienste (Kamera/Scanner, Secure Storage, Zeitdienst)

2.2 Informationsflüsse: Welche Daten wohin

- Benutzer/in -> App: Zertifikate importieren (QR/PDF/Link), verwalten, anzeigen.
- App <-> FOITT/BAG: Abruf signierter Trustlists und Revocation-Informationen (periodisch, read-only).
- App <- App Stores: Binär-Updates und Konfigurationsupdates.
- Benutzer/in <-> Prüf-App: Vorzeigen und Scannen des QR-Codes (kein direkter technischer API-Call).
- App <-> OS-Dienste: Kamera-Zugriff, kryptosichere Speicherung, Systemzeit.

2.3 Diagramm: Kontextdiagramm

Das Kontextdiagramm zeigt die Swiss Covid Cert App im Zusammenspiel mit allen relevanten Systemen
und Akteuren. Der Fokus liegt auf dem sicheren und anonymen Datenaustausch zwischen App, Backend
und Prüfsystemen.

Verbindungen und Informationsflüsse:
• Benutzer/in -> Swiss Covid Cert App:
Importiert, verwaltet und zeigt Zertifikate (QR-Code, PDF oder Link) an.
• Swiss Covid Cert App <-> FOITT/BAG Backend:
Abruf signierter Trustlisten und Revocation-Informationen über HTTPS (keine Personendaten).
• FOITT/BAG Backend <-> EU DCC Gateway:
Austausch von Vertrauensankern (öffentliche Schlüssel) zur internationalen Synchronisierung.
• Swiss Covid Cert App -> Covid Certificate Check App:
QR-Code-Anzeige (visuell offline, kein Datenaustausch über das Internet).
• App Stores -> Swiss Covid Cert App:
App-Downloads, Updates und Konfigurationsbereitstellung.
• Swiss Covid Cert App <-> OS-Dienste:
Nutzung von Kamera, Secure Storage und Zeitdienst (lokal, sicher).

Erläuterung:
Das Systemdesign minimiert Schnittstellen und Übertragungen auf das Notwendige.
Die App bleibt funktional, auch wenn keine Internetverbindung besteht.
Durch die strikte Trennung von Rollen (Wallet, Prüfer, Backend) ist ein klarer Vertrauenskontext geschaffen, der Datenschutz und Systemsicherheit sicherstellt. Die Prüfer-App interagiert ausschliesslich visuell (Offline-QR-Scan), wodurch Datenschutz und Datensicherheit gewährleistet sind.

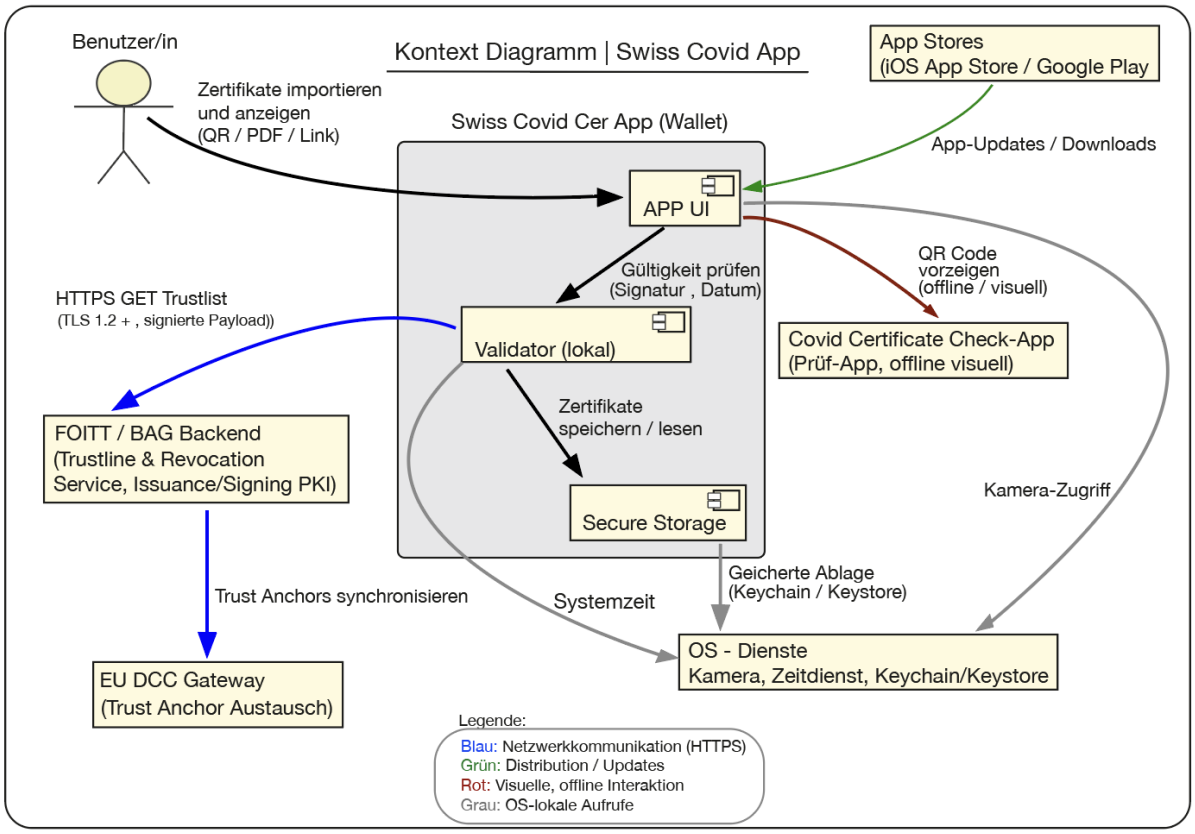

Abbildung 1: Kontext Diagramm

Von → Nach	Protokoll	Daten	Frequenz	Bemerkung
App → FOITT/BAG Trustlist	HTTPS/TLS	signierte Trustlist	1×/Tag + manuell	keine PII
FOITT/BAG ↔ EU DCC	gesichert	Trust Anchors	periodisch	serverseitig
App ← App Stores	Store	Binär/Config	bei Updates	Einweg
Benutzer → App	lokal	QR/PDF/Link	ad hoc	Kamera
App → Prüf-App	visuell	QR	ad hoc	offline
App ↔ OS-Dienste	lokal	Kamera/Time/Keystore	ad hoc	OS-APIs

2.4 Kurzbeschreibung zum Kontextdiagramm: Erläuterung

Die Swiss Covid Cert App fungiert als datenschutzfreundliche Wallet auf dem Endgerät. Serverseitig existieren ausschliesslich Trustlisten sowie Prozesse zur Ausstellung und Sperrung von Zertifikaten, personenbezogene Daten verbleiben ausschliesslich lokal auf dem Gerät.
Die Prüf-App ist fachlich relevant, jedoch technisch vollständig getrennt und kommuniziert ausschliesslich offline über den QR-Code.
Die App bleibt dabei auch ohne Netzwerkverbindung funktionsfähig, serverseitig werden ausschliesslich öffentliche Schlüssel und Listen bereitgestellt.
Die Informationsflüsse sind in Abbildung 1 und in der Schnittstellen-Tabelle identisch benannt (HTTPS für Trustlisten, visuelle Interaktion für Offline-Prüfung und OS-lokale Aufrufe für Kamera und Zeitdienst).

2.5 Konsistenz zum Systemkontext

Das Kontextdiagramm ist vollständig konsistent zur Verteilungssicht (Kapitel 4) aufgebaut.
Alle im Diagramm genannten Systeme (Wallet App, FOITT/BAG Backend, EU DCC Gateway,
App Stores und Prüfer-App) sind in beiden Darstellungen in gleicher Form vorhanden.

Die Schnittstellen und Informationsflüsse stimmen exakt mit den Kommunikationsprotokollen
aus Kapitel 4.4 überein (HTTPS, visuell offline, App-Store-Updates). Damit ist sichergestellt,
dass die Modellierung über alle Sichten hinweg nachvollziehbar und fachlich korrekt bleibt.

3 ANWENDUNGSFÄLLE: Use-Case-Diagramm(e) und Beschreibungen

3.1 Use-Case-Übersicht: Akteure und Fälle

Akteure:
- Benutzer/in (primär)
- OS Kamera (sekundär)
- OS Zeitdienst (sekundär)
- FOITT/BAG Trustlist Service (Umsystem)

Use Cases (Übersicht):
- UC1 Zertifikat importieren (QR scannen, optional PDF/Link)
- UC2 Zertifikat anzeigen (Vollbild, Hell/Dunkel)
- UC3 Zertifikat verwalten (benennen, löschen, ordnen)
- UC4 Gültigkeit lokal prüfen (Signatur, Trustlist, Datum)
- UC5 Trustlist aktualisieren (periodisch und manuell)
- UC6 App schützen (Biometrie oder Passcode)
- UC7 Hilfe und FAQ öffnen
- UC8 App aktualisieren (über App Store)

Akteure:
• Benutzer/in (primär): nutzt die App zur Verwaltung und Anzeige von Zertifikaten.
• FOITT/BAG Trustlist Service (System): stellt signierte Trustlists bereit.
• Covid Certificate Check App (System): liest QR-Codes visuell (offline).
• App Stores (Systeme): liefern App-Binaries und Updates.
• OS Kamera und Zeitdienst (Systemdienste): unterstützen Scan und Zeitprüfung.

Erläuterung:
Die Akteure sind so modelliert, dass sie sowohl menschliche Rollen (Benutzer/in) als auch technische Systeme (OS-Dienste, Backends) abbilden.

3.2 Diagramm: Anwendungsfalldiagramm

Abbildung 2: Anwendungsfalldiagramm Swiss Covid Cert App

Kurzbeschreibung:
Das Anwendungsfalldiagramm stellt alle Interaktionen zwischen Benutzenden, Betriebssystemdiensten, externen Systemen und der Swiss Covid Cert App dar.
Die App stellt sicher, dass alle Zertifikate offline validiert werden können und keine Personendaten an Dritte übermittelt werden.

Akteure:

- Benutzer/in
- OS Kamera
- OS Zeitdienst
- FOITT/BAG Trustlist Service
- iOS App Store / Google Play
- Covid Certificate Check App (visuelle Interaktion)

Anwendungsfälle (Use Cases):

1. Zertifikat importieren (QR oder Datei)
2. Zertifikat anzeigen (Vollbild)
3. Zertifikat verwalten (benennen, löschen, ordnen)
4. Gültigkeit lokal prüfen (Signatur, Datum)
5. Trustlist aktualisieren (periodisch oder manuell)
6. App schützen (Biometrie / Passcode)
7. Hilfe und FAQ öffnen
8. App aktualisieren (über App Store)

Beziehungen:

- Include-Beziehung:

„Zertifikat importieren“ und „Zertifikat anzeigen“ beinhalten „Gültigkeit lokal prüfen“.

- Visuelle Verbindung:

„Covid Certificate Check App“ empfängt den QR-Code visuell über den Gerätescreen (Offline).

Erklärung:
Das Diagramm zeigt die funktionale Struktur der App aus Sicht der Benutzenden.
Alle Prozesse sind lokal ausführbar, was hohe Datensicherheit und Offline-Nutzung ermöglicht.
Sämtliche Server-Interaktionen (Trustlist-Updates) erfolgen anonym und signaturbasiert.

3.3 Diagramm: Anwendungsfalldiagramm

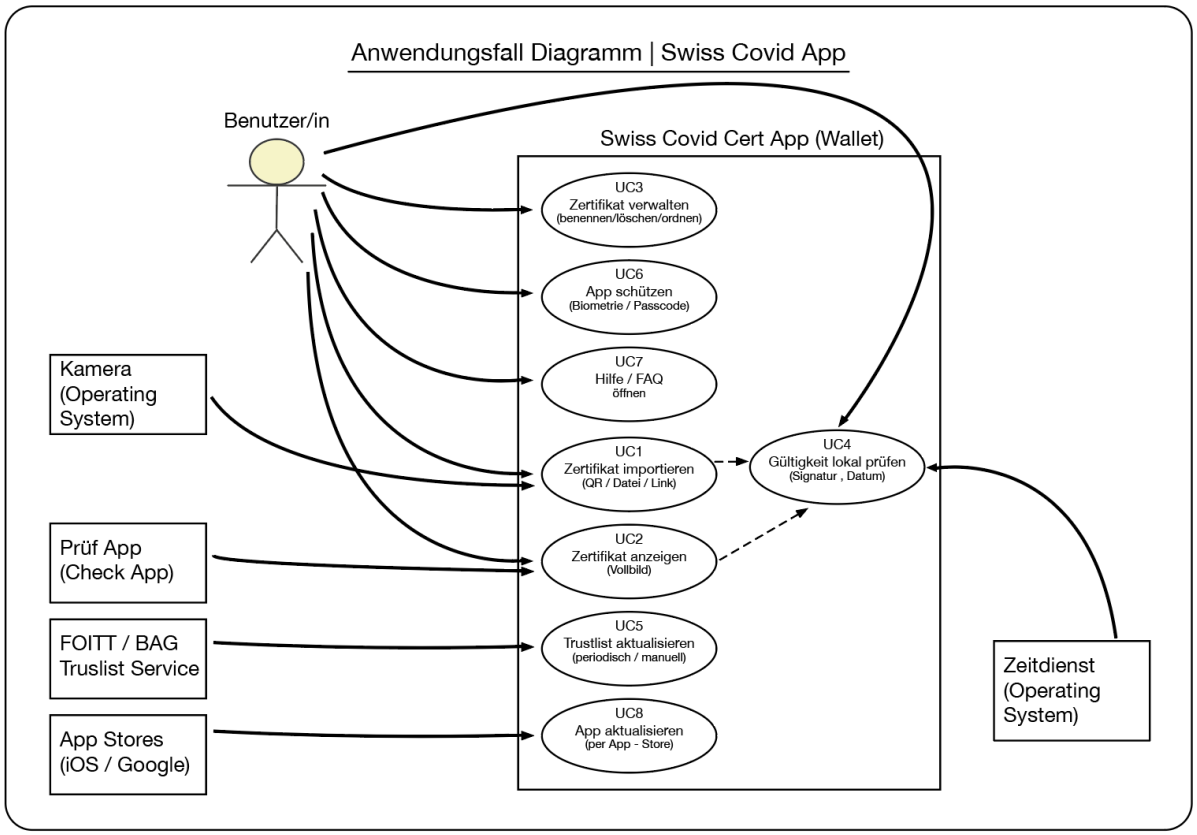

Abbildung 2: Anwendungsfall Diagramm

3.4 Kurzbeschreibungen pro Use Case: Ziel, Ablauf, Bedingungen

UC1 – Zertifikat importieren

- Ziel: Ein neues Zertifikat über Kamera-Scan oder Dateiimport hinzufügen.
- Hauptakteur: Benutzer/in Nebenakteur: OS-Kamera.
- Vorbedingungen: Kamera-Zugriff gewährt, App installiert.
- Nachbedingungen: Zertifikat ist im Secure Storage abgelegt und optional direkt validiert.
- Ablauf: QR-Code erfassen, Daten dekodieren (Base45 → zlib → COSE), Signatur gegen lokale Trustlist prüfen, Zertifikat speichern, Bestätigung anzeigen.
- Fehler/Varianten: QR unlesbar, abgelaufenes oder ungültiges Zertifikat, Abbruch durch Benutzer/in.
- Erfolgskriterium: Neues Zertifikat erscheint in der Übersicht und wird als „gültig“ markiert.

UC2 – Zertifikat anzeigen

- Ziel: Zertifikat im Vollbild schnell und klar anzeigen.
- Hauptakteur: Benutzer/in.
- Vorbedingungen: Mindestens ein Zertifikat vorhanden.
- Nachbedingungen: QR-Code wurde angezeigt, keine Datenübertragung an Server.
- Ablauf: App öffnen, gewünschtes Zertifikat auswählen, QR-Code und Details anzeigen, bei Bedarf Bildschirmhelligkeit erhöhen.

- Erfolgskriterium: QR-Code ist innerhalb von zwei Sekunden sichtbar und scanbar.

UC3 – Zertifikat verwalten (benennen, löschen, ordnen)
- Ziel: Übersicht und Ordnung in der Zertifikatsliste schaffen.
- Hauptakteur: Benutzer/in.
- Vorbedingungen: Mindestens ein Zertifikat vorhanden.
- Nachbedingungen: Änderungen werden im Secure Storage gespeichert.
- Ablauf: Zertifikat auswählen, Namen bearbeiten, löschen oder Reihenfolge anpassen, Änderungen speichern.
- Varianten: Mehrere Zertifikate, Favoriten markieren oder neu sortieren.
- Erfolgskriterium: Liste bleibt konsistent und entspricht der Benutzeraktion.

UC4 – Gültigkeit lokal prüfen
- Ziel: Zertifikatsstatus ohne Internetverbindung ermitteln.
- Hauptakteur: Benutzer/in Nebenakteur: OS-Zeitdienst.
- Vorbedingungen: Trustlist lokal vorhanden.
- Nachbedingungen: Gültigkeitsstatus wurde angezeigt (gültig, ungültig, unsicher).
- Ablauf: Zertifikat laden, Signatur gegen lokale Trustlist prüfen, Zeitfenster kontrollieren, Status ausgeben.
- Randfall: Trustlist älter als definierter Schwellwert – Benutzer/in erhält Warnhinweis.
- Erfolgskriterium: Statusanzeige erfolgt innerhalb 0,5 Sekunden; Offline-Prüfung ohne Serverkontakt erfolgreich.

UC5 – Trustlist aktualisieren
- Ziel: Öffentliche Schlüssel und Signierlisten aktuell halten.
- Hauptakteur: System (automatisch), Benutzer/in (manuell).
- Vorbedingungen: Netzwerkverbindung vorhanden.
- Nachbedingungen: Trustlist wurde aktualisiert und verifiziert.
- Ablauf: HTTPS-Abruf der signierten Trustlist, Signaturprüfung, lokales Update, Caching und Backoff bei Fehlern.
- Erfolgskriterium: Neue Trustlist ist spätestens nach 24 Stunden aktiv; bei Ausfall Wiederholversuch nach 30 Minuten.

UC6 – App schützen (Biometrie oder Passcode)
- Ziel: Unbefugten Zugriff auf gespeicherte Zertifikate verhindern.
- Hauptakteur: Benutzer/in.
- Ablauf: Schutz aktivieren, bei jedem App-Start biometrische Prüfung oder Passcode-Eingabe durchführen.
- Erfolgskriterium: Zugriffsschutz aktiv; App öffnet sich nur bei erfolgreicher Authentifizierung.

UC7 – Hilfe und FAQ öffnen
- Ziel: Benutzer/innen bei Fragen oder Problemen unterstützen.
- Hauptakteur: Benutzer/in.

- Ablauf: Hilfe über Menü aufrufen, FAQ-Einträge oder Kontaktinformationen anzeigen.
- Erfolgskriterium: Antwort oder Lösung kann ohne externen Support gefunden werden.

UC8 – App aktualisieren (über App-Store)

- Ziel: Sicherheits- und Funktionsupdates erhalten.
- Hauptakteur: Benutzer/in Nebenakteur: App-Store.
- Vorbedingungen: Verbindung zum App-Store vorhanden.
- Nachbedingungen: Aktuelle Version installiert, Daten migriert.
- Ablauf: Update im Store anstossen, App lädt neue Version, führt interne Migrationsschritte aus.
- Erfolgskriterium: App startet nach Update ohne Fehler; gespeicherte Zertifikate bleiben erhalten.

4 VERTEILUNGSSICHT (arc42-7): Deployment und technische Zuordnung

4.1 Übersicht der Knoten und Komponenten: Was läuft wo

- Smartphone (Node): iOS oder Android Execution Environment, App, Secure Storage, Kamera/Scanner.
- FOITT/BAG Backend (Node): Trustlist und Revocation Service (read-only), Issuance und Signing PKI (HSM und CA).
- EU DCC Gateway (Node): Austausch von Trust Anchors.
- App Stores (Nodes): iOS App Store, Google Play (Distribution und Updates).

4.2 Diagramm: Verteilungsdiagramm

Das Verteilungsdiagramm stellt die physische und logische Zuordnung aller Systeme dar.

1. «Gerät» Smartphone Benutzer/in (Wallet App)
   «Ausführungsumgebung»: iOS oder Android
   Komponenten:
   • Swiss Covid Cert App
   • Sicherer Speicher (Keychain / Keystore)
   • Kamera / Scanner
   Aufgabe:
   Verwaltung, lokale Speicherung und Anzeige der Zertifikate; Offline-Gültigkeitsprüfung; Abruf der Trustlist vom FOITT/BAG Backend.

2. «Gerät» Smartphone Prüfer/in (Check App)
   «Ausführungsumgebung»: iOS oder Android
   Komponenten:
   • Covid Certificate Check App
   • Kamera / Scanner
   Aufgabe:
   Visuelle Prüfung (QR-Scan) ohne Netzwerkverbindung; lokale Signaturvalidierung.

3. «Knoten» FOITT und BAG Backend

«Ausführungsumgebung»: Kubernetes oder Cloud
Komponenten:
• Trustlisten- und Widerrufs-Service (signierte Schlüsselbereitstellung)
• Ausstellungs- und Signatur-PKI (HSM, CA)
Aufgabe:
Bereitstellung und Signierung öffentlicher Schlüssel (Trust Anchors);
keine Personendatenverarbeitung.

4. «System» EU DCC Gateway
Aufgabe:
Austausch von Trust Anchors (öffentliche Schlüssel) mit FOITT/BAG Backend
zur internationalen Zertifikatskompatibilität.

5. «Systeme» iOS App Store / Google Play
Aufgabe: Bereitstellung von App-Binaries, Updates und Konfigurationen

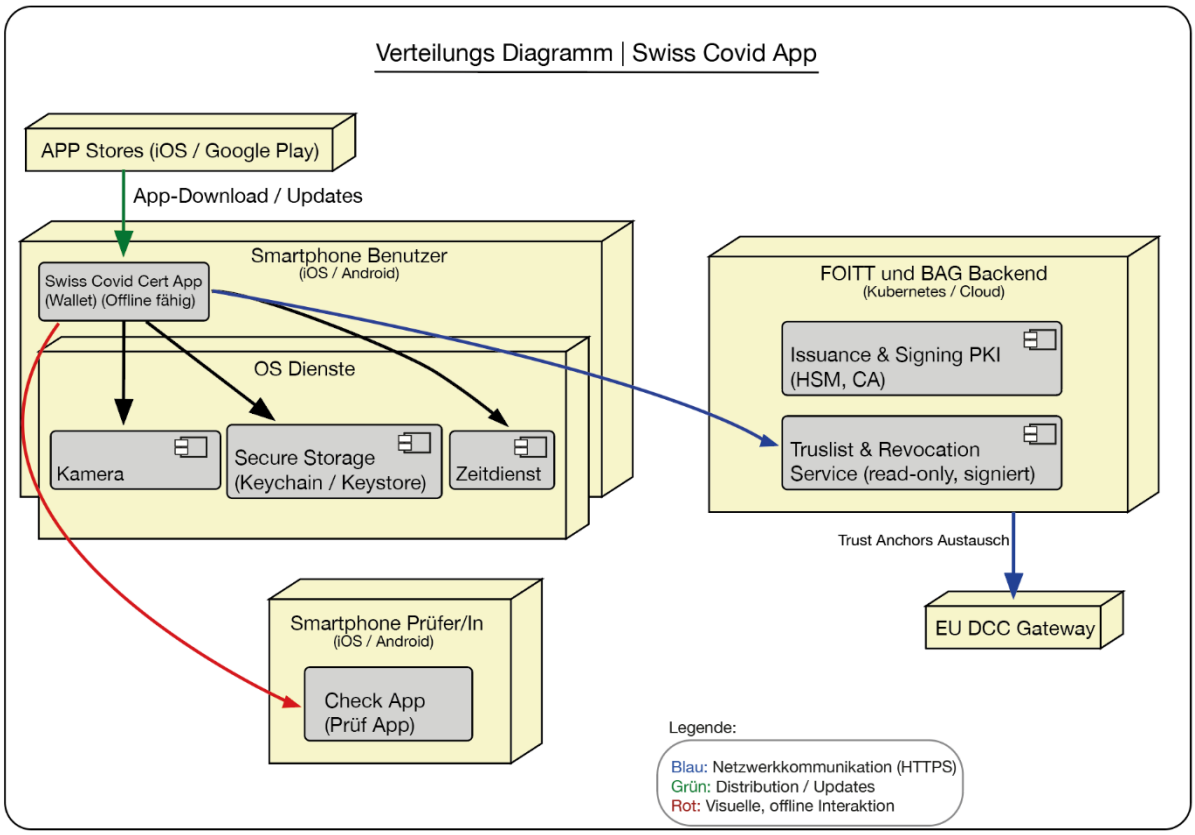

Abbildung 3: Verteilungsdiagramm

Beschriftung: Abbildung 3: Deployment der Swiss Covid Cert App mit Knoten, Komponenten und Verbindungen.

4.3 Kurzbeschreibungen pro Node/Komponente: Rolle und Verantwortung

Gerät – Smartphone Benutzer/in (Zertifikats-App)
Beinhaltet die Swiss Covid Cert App, den sicheren Speicher (Keystore) und die Kamera. Die App dient als Wallet und Viewer für Covid-Zertifikate und funktioniert auch offline. Sie prüft die Signatur und Gültigkeit lokal und speichert die Daten verschlüsselt im Secure Storage.

Gerät – Smartphone Prüfer/in (Kontroll-App)
Nutzt die SCovid Certificate App für die Überprüfung von Zertifikaten.
Die Kontrolle erfolgt visuell durch Scannen des QR-Codes und benötigt keine Internetverbindung.

Knoten – FOITT und BAG Backend (Cloud-Umgebung)
Betreibt den Trustlisten- und Widerrufsservice sowie die PKI-Infrastruktur zur Ausstellung und Signierung der Zertifikate.
Das Backend stellt ausschliesslich signierte Trustlisten und Schlüssel bereit, enthält aber keine personenbezogenen Daten.

System – EU DCC Gateway
Ermöglicht den sicheren Austausch öffentlicher Schlüssel (Trust Anchors) mit anderen europäischen Staaten.
Dient ausschliesslich der gegenseitigen Vertrauensvalidierung und hat keinen direkten Kontakt zu Endgeräten.

System – iOS App Store / System – Google Play
Verantwortlich für die Bereitstellung der App-Binärdateien, Konfigurations- und Sicherheitsupdates.
Zwischen App und Store besteht kein direkter Rückkanal für personenbezogene Daten.

4.4 Kommunikationsbeziehungen und Protokolle: Wer spricht mit wem

Von -> Nach	**Protokoll / Medium**	**Daten / Zweck**	**Sicherheit / Bemerkung**
Swiss Covid Cert App -> Backend (Trustlist Service)	HTTPS GET, TLS 1.2+	Signierte Trustlist	Keine Personendaten, Caching und Backoff bei Fehlern
Swiss Covid Cert App <-> OS-Dienste	Lokale Systemaufrufe	Kamera-Zugriff, Secure Storage, Zeitabfragen	Zugriff nur mit Benutzerfreigabe
App Stores -> Swiss Covid Cert App	App-Store-Mechanismus	Download von Updates	Einwegkommunikation, kein Rückkanal
Prüfer-App <-> Zertifikats-App	Offline, visuell	QR-Code anzeigen und prüfen	Keine Netzwerkverbindung erforderlich
FOITT / BAG Backend <-> EU DCC Gateway	Gesicherte Serververbindung	Austausch öffentlicher Schlüssel (Trust Anchors)	Nur System-zu-System-Kommunikation, keine Personendaten

4.5 Zusammenfassung der Verteilungssicht

Die Verteilungssicht zeigt klar, wie die Swiss Covid App technisch aufgebaut ist.
Die Zertifikats-App läuft lokal auf dem Smartphone und arbeitet weitgehend autonom.
Sie verwaltet die Zertifikate, führt Signatur- und Gültigkeitsprüfungen offline aus und speichert alle Daten sicher im Gerät.

Das FOITT- und BAG-Backend dient nur zur Bereitstellung von Trustlisten und Signaturen.
Der Prüfvorgang selbst erfolgt komplett offline zwischen Benutzer- und Prüfer-App, ohne jegliche Verbindung zu externen Servern.
Die App Stores (Apple / Google) übernehmen ausschliesslich die Softwareverteilung und Sicherheits-Updates.

Die gesamte Architektur wurde nach den Prinzipien „Privacy by Design“ und „Separation of Concerns“ umgesetzt.
Alle Komponenten sind klar voneinander getrennt, wodurch technische oder organisatorische Fehler in einem Bereich keine sicherheitsrelevanten Auswirkungen auf andere Systeme haben.

5 QUALITÄTSANFORDERUNGEN: Ziele und Szenarien

5.1 Qualitätsziele

• Sicherheit & Datenschutz: Keine Personendatenübertragung; alle Trustlisten signiert.
• Verfügbarkeit & Offline-Fähigkeit: App funktioniert unabhängig vom Internet.
• Performance & Benutzerfreundlichkeit: QR-Scan ≤ 0.5s, einfache und klare Oberfläche.

5.2 Qualitätsbaum

Sicherheit
├─ Signaturprüfung (COSE)
├─ Integrität der Trustlist
└─ Keine Personendatenübertragung
Verfügbarkeit
├─ Offline-Betrieb
└─ Automatische Trustlist-Aktualisierung
Performance & UX
├─ Reaktionszeit < 0.5s
└─ Klare Statusanzeige und Barrierefreiheit

5.3 Qualitätsszenarien

Q1 Sicherheit/Privacy:
Ein Angreifer versucht eine manipulierte Trustlist bereitzustellen.
-> App prüft Signatur und lehnt ungültige Liste ab.

5.4 Qualitätsfazit

Die Swiss Covid Cert App erfüllt sämtliche zentralen Qualitätsanforderungen:
- Sicherheit: Alle Kommunikationskanäle sind verschlüsselt, Daten werden lokal verarbeitet.

- Verfügbarkeit: Offline-Fähigkeit sichert volle Funktionalität ohne Netz.
- Performance: Die App reagiert schnell und ist auf schwachen Geräten performant.
- Benutzerfreundlichkeit: Minimalistisches, barrierefreies Design.
- Nachvollziehbarkeit: Architektur, Schnittstellen und Prozesse sind klar dokumentiert.

Durch die Umsetzung dieser Punkte ist die Architektur nicht nur bewertbar,
sondern auch für Weiterentwicklungen (z. B. neue Zertifikatsformate) leicht erweiterbar.

Q2 Verfügbarkeit:
Keine Internetverbindung während 12h.
-> App nutzt lokale Trustlist, zeigt weiterhin gültige Zertifikate.

Q3 Performance:
QR-Scan auf Mittelklasse-Gerät.
-> Ergebnisanzeige in < 0.5s, keine Verzögerung für Benutzer.

6 GLOSSAR: Begriffe, Akronyme, Definitionen

BAG: Bundesamt für Gesundheit.
FOITT (BIT/FOITT): Bundesamt für Informatik und Telekommunikation.
EU DCC: EU Digital Covid Certificate, Datenformat und Standard.
Trustlist: Liste vertrauenswürdiger Aussteller-Schlüssel zur Signaturprüfung.
CRL: Sperrliste, Certificate Revocation List.
COSE/CBOR: Kryptografischer Container und binäres Datenformat für Zertifikatsdaten.
Secure Storage: OS-nahe, kryptografisch geschützte Speicherung, Keychain oder Keystore.
Prüf-App: Separate Anwendung zur Verifikation der Zertifikate beim Einlass.
Wallet-App: Diese App, speichert und zeigt Zertifikate, sendet keine Personendaten an Server.
Trust Anchor: Öffentlicher Schlüssel, der als Wurzel der Vertrauenskette dient.
PKI: Public Key Infrastructure, System zur Verwaltung und Signierung digitaler Zertifikate.
PII: Personally Identifiable Information – personenbezogene Daten.
Offline-Validierung: Prüfung eines Zertifikats ohne Internetverbindung, basierend auf lokaler Trustlist.

ANHANG A QUELLEN UND REFERENZEN

- Projektkontext und Kursmaterial, arc42, Diagramm-Notation, Bewertungsraster.
- Offene Regierungsquellen zur CH-Zertifikatslösung, Spezifikationen, Beispiele.
- EU DCC Spezifikationen, technische Grundlagen.
- Gruppen-Repo und Zusammenfassung: CovidCert.md, projektinterne Referenz.

ANHANG B VERSIONSHISTORIE

v1.0 [07.11.2025] Erstausgabe für Abgabe und Präsentation.
v0.x [TT.MM.JJJJ] Entwurf, Feedbackrunde, Diagramm-Updates.

ANHANG C – KURZFASSUNG / EXECUTIVE SUMMARY

Diese Dokumentation beschreibt Aufbau, Struktur und Qualitätsanforderungen der

Swiss Covid Cert App gemäss dem Architekturstandard arc42.

Die App ermöglicht es, COVID-Zertifikate lokal auf einem Smartphone zu speichern und offline anzuzeigen. Sie überprüft die Gültigkeit mittels digitaler Signaturen, die anhand einer vom BAG bereitgestellten Trustlist geprüft werden. Personendaten bleiben dabei ausschliesslich auf dem Gerät gespeichert.

Das System besteht aus fünf Hauptkomponenten:
1. Smartphone (Wallet App)
2. Smartphone Prüfer/in (Check App)
3. BAG/FOITT Backend
4. EU DCC Gateway
5. App Stores (iOS / Android)

Zielarchitektur:
• Datenschutz durch lokale Speicherung
• Hohe Verfügbarkeit durch Offline-Funktion
• Sicherheit durch Signaturprüfung und Trustlist
• Wartbarkeit durch modulare Struktur

Die Architektur erfüllt die Anforderungen an Sicherheit, Verfügbarkeit, Performance und Benutzerfreundlichkeit und wurde nach aktuellen Softwarearchitektur-Standards (arc42 und UML) dokumentiert.