**Corona Services Overview**

[cwa-documentation/backend-infrastructure-architecture.pdf at master · corona-warn-app/cwa-documentation (github.com)](https://github.com/corona-warn-app/cwa-documentation/blob/master/backend-infrastructure-architecture.pdf)

1. CWA Server: Handles the following tasks:
   * accepting submission of new keys
   * aggregation/distribution of diagnosis keys and configuration files
   * uploading and downloading keys to/from the federation gateway to enable interoperability with other EU nations
2. Verification Server: Deals with test result retrieval and verification (including issuing TANs). The components regarding the verification are managed and deployed separately.

**CWA-Server: Submission Service:**

* [cwa-server/SUBMISSION.md at main · corona-warn-app/cwa-server (github.com)](https://github.com/corona-warn-app/cwa-server/blob/main/docs/SUBMISSION.md)
* The submission service's only task is to process uploaded diagnosis keys and persist them to the database after the TAN has been verified. The actual task of the verification is handed over to the verification server, which provides the verification result back to CWA. After verification was successfully done, the diagnosis keys are persisted in the database, and will be published in the next batch for distribution to the CWA CDN and to the Federation Gateway for keys that are applicable.

**CWA-Server: Distribution Service:**

* [cwa-server/DISTRIBUTION.md at main · corona-warn-app/cwa-server (github.com)](https://github.com/corona-warn-app/cwa-server/blob/main/docs/DISTRIBUTION.md)
* The distribution service's objective is to publish all CWA-related files to the object store, from which the clients will fetch their data. There are three types of files.
* The same 14 days retention period (like the database) is also enforced on the S3 compatible storage. Each distribution run will execute the retention policy.

**Event Registration Service:**

* QR kód lenne egy eseményre/étteremben, és akkor ezt a user be tudja scanelni, majd könnyebben lehetne összegyűteni, hogy ki és hol volt. Ha nincs app még a telefonján és bescaneli, akkor egy nginx-en futna egy weboldal (Appstore-website namespace - [https://e.coronawarn.app](https://e.coronawarn.app/) ), és közvetlen megmutatná, hogy hogy tudja az appot letölteni

**Appstore (-website/-s)**

* hash guid-ot csinálnak mögötte, redirectre is használják
* E (sima) - event registerre mutat
* S - quick test serverre mutat
* QR kódnak része ez a site és amögé pakolják a hash-t
* QR kódok ide fognak mutatni (redirectelni), amiket az appban használunk
* Quicktesthez most is vannk qr kódok, amik a testresulthoz tartoznak
* Eventhez is van, amiből generálódik egy QR kód, és a QR kódban benn van a link is
* URL tartalmazza a nevet, test-et, salt, hash..., base64-es kódolással
* Website-ok egy service-t használnak különböző routeokkal -> appstore-website-s megszűnt és minden a sima appstore-website alatt fut

**CWA Server-ex: Upload Service:**

* [cwa-server/FEDERATION-UPLOAD.md at main · corona-warn-app/cwa-server (github.com)](https://github.com/corona-warn-app/cwa-server/blob/main/docs/FEDERATION-UPLOAD.md)
* This is a spring boot [ApplicationRunner](https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/ApplicationRunner.html) service (running as a cronjob). The app will deal with the upload of DE keys to the federation gateway. The job will be configured to run periodically throughout the day to enable keys to be shared with the federation gateway and will run independently from the DE CDN distribution job. This job will source keys from a dedicated table populated with keys which are relevant for sharing. Once a key is published to the gateway via an upload batch it would be removed from this table to ensure it is not resent.
* Federation Gateway: Enables the sharing of keys across integrated app backends of nations within the European Union (EFGS – European Federation Gateway Service)

**CWA Server-ex: Download Service:**

* [cwa-server/FEDERATION-DOWNLOAD.md at main · corona-warn-app/cwa-server (github.com)](https://github.com/corona-warn-app/cwa-server/blob/main/docs/FEDERATION-DOWNLOAD.md)
* This is a spring boot [ApplicationRunner](https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/ApplicationRunner.html) service (running as a cronjob). The app will deal with the download, sematic validation, extraction, and storage of the keys from the federation gateway. The download service leverages the download API of the federation gateway and will trigger downloads of available batches provided by the Callback Service.
* In addition to callback-based downloads, the Download Service supports date-based downloads. For that, the download service will use the polling mechanism provided by the federation gateway based on batchTag and date combinations and it will keep track of its last processed state within the database. The polling mechanism would mainly be used for mass loading scenarios.
* On the download of keys from the federation gateway a process to derive a TRL from the DSOS needs to take place. This is done to enable the keys to be consumable by the DE CWA app as not all countries support the same approach which is required for the CWA app. The means the following:
  + For keys from countries which support the daysSinceOnset scenario will need to be converted into an appropriate transmission risk level based on values defined in configuration
  + For keys where we have no DSOS mapping configuration there will be a need to provide a reasonable global default otherwise we would need to reject the key download.
* These rules will allow the keys sourced from the federation gateway to be processed within the CWA App and be considered with the risk detection algorithms.
* Previously the download job was running once per hour and even though we were register to callback, we used the flag enforcedatebaseddownload as true. This would mean that the callback actually just store the new batchtags in  the federation\_batch\_info table, but the download job would delete them as first step and would use the actual date to fetch all batches again.
* Now , the old download job would run with that engforcedatebaseddownload on false which would mean that the batches that would be processed by the callback would be downloaded by the download job without any cleaning before. And the new download job (called: **download-databased**) would run once per day as it worked before.

**CWA Server-ex: Callback Service:**

* Expose a callback API which can be used by the EFGS to notify CWA when new key batches are available for download

**CWA Server-ex: Download-CH Service:**

* Ugyanazt csinálja, mint a sima Download service, viszont ez nem az EFGS-el kommunikál, hanem a Svájci Gateway-el (CHGS).

**CWA Server-ex: Upload-CH Service:**

* Ugyanazt csinálja, mint a sima Upload service, viszont ez nem az EFGS-el kommunikál, hanem a Svájci Gateway-el (CHGS).

**Verification Server:**

* [cwa-verification-server/architecture-overview.md at master · corona-warn-app/cwa-verification-server (github.com)](https://github.com/corona-warn-app/cwa-verification-server/blob/master/docs/architecture-overview.md)
* The primary scope of the component is to provide a proof whether users were tested positive for SARS-CoV-2. The secondary scope is to provide information about the status of a SARS-CoV-2 test and to provide the ability for privileged users (Health Authorities) to obtain a proof document on behalf of tested users.
* **Detailed:** The verification server supports the user's journey beginning at scanning the QR code printed on the documentation of the SARS-CoV-2 test until the upload of diagnosis keys when the user was tested positive. Testing labs will provide the results of SARS-CoV-2 tests to the [test result server](https://github.com/corona-warn-app/cwa-testresult-server), which in turn provides an interface to the [verification server](https://github.com/corona-warn-app/cwa-verification-server). Since the GUID contained in the QR code is linked to a test, the mobile application is able to fetch the results from the verification server and provide a notification to the user. After users have given their consent to upload their diagnosis key, a TAN is fetched from the verification server. This TAN will be used as an authorization token when the user uploads the diagnosis keys of the past 14 days.
* Therefore, from a CWA Server perspective, the Verification Server provides an endpoint for TAN verification.

**Testresult server:**

* [cwa-testresult-server/architecture-overview.md at master · corona-warn-app/cwa-testresult-server (github.com)](https://github.com/corona-warn-app/cwa-testresult-server/blob/master/docs/architecture-overview.md)
* The primary scope of the component is to provide the verification server with information of lab test results of SARS-CoV-2 tested people. The Test Result Server provides test results of SARS-CoV-2 tests in a pseudonymized form to only the verification server as it requests such information.

**Verification Portal:**

* [cwa-verification-portal/architecture-overview.md at master · corona-warn-app/cwa-verification-portal (github.com)](https://github.com/corona-warn-app/cwa-verification-portal/blob/master/docs/architecture-overview.md)
* The primary scope of the component is to create a proof certificate (teleTAN) whether users were positive tested for SARS-CoV-2 or not. The proof is generated by hotline employees based on the information the can obtain. The scope of the Verification Portal is just the creation of the proof certificate - the teleTAN. Not is scope is the support of validating the request of people calling the hotline.
* The Verification Portal is part of the Portal Server. The Portal Server consists of this component and the Verification IAM and is a logical grouping of services and not a "Server" in the traditional sense.

**Röviden/Összefoglalva:**

In the world of the Corona Warn App the Verification *Identity and Access Management (IAM)* ensures that only authorized health personnel get access to the Verification Portal. The parts of the verification component cooperate in the following manner:

* *Verification Server* (repository: cwa-verification-server)
  + helps validating whether upload requests from the mobile App are valid or not.
* *Verification Portal* (repository: cwa-verification-portal)
  + allows hotline employees to generate teleTANs which are used by users of the mobile App to upload their diagnostic keys.
* *Verification Identity and Access (IAM)* (repository: cwa-verification-iam)
  + ensures that only authorized health personnel get access to the Verification Portal.
* *Test Result Server* (repository: cwa-testresult-server)
  + receives the results from laboratories and delivers these results to the app via the verification-server.
* *One Time Password Generator (OTP Generator)*
  + used to provide the 2nd factor for authentication

**Vault:**

* Secreteket tároljuk itt. TLS certek itt találhatóak.

**Log-viewer/Log-upload:**

* [cwa-log-upload/Overview.md at master · corona-warn-app/cwa-log-upload (github.com)](https://github.com/corona-warn-app/cwa-log-upload/blob/master/doc/Overview.md)
* The log upload service is the counterpart of the log upload in the app. It enables the App developers to analyse the log data uploaded to the CWA infrastructure to indentify the root cause of bugs that only occur in rare conditions are not easy to reproduce. The log upload and viewer service follows the privacy preserving paradigm of the corona warn app, only allowing authorized personell to access the logs. The means to restrict access are the same that also protect TeleTAN generation from abusive usage.
* Error log sending, ha valami nem megy, akkor egy gombnyomással feltöltheti az error logokat egy URL (logupload.coronawarn.app) a user. OBS bucket-be berakja a logokat, és egy Postgre adatbázisban rendszerezve vannak ezek a logok. Validációt is csinál, ami átmegy a CWA-data-ba (mint a google és ios verifikáció cwa-data esetében)

**Website:**

* This repository contains the source files of the official website for the Corona-Warn-App as it is available at [coronawarn.app](https://coronawarn.app/).

**CWA-ppa-server/CWA Data Service:**

* [corona-warn-app/cwa-ppa-server: Data Donation Server for receiving and storing users usage data and event driven user surveys. (github.com)](https://github.com/corona-warn-app/cwa-ppa-server)
* Telefonokról szeretnének adatokat begyűjteni (megkérdezni a usereket, hogy bele egyeznek-e az adatgyűjtésbe). Hogyan lehet használhatóbb az app, milyen szokások vannak... stb. Biztosítani kell, hogy egy valós telefonról jönnek és mondjuk nem csak botról. Kap egy tokent a telefon és akkor ez kommunikál google-el és apple-el, hogy valós-e. Ez a service ezeket az adatokat gyűjtené, validálná és RDS-be tenné, illetve RKI-nek is szolgáltatnánk adatokat. Két API. Telefon, google, apple egy api, rki felé mégegy.

Működés:

1. A user kiválszatja, hogy részt akar e venni (ebből lesznek a magenta színű telefonok)
2. Tokent küld a telefonra a data service, mivel ez kell az autentikációhoz, hogy nem egy bot küldözgeti
3. Tokennel beautentikál vagy a Google vagy az IOS autentikátorba
4. Miután sikeres volt az autentikáció, akkor ténylegesen el is küldi az adatokat a data-ba feldolgozásra
5. Megtörténik a feldolgozás és az RKI-nak megy statisztiaki adatok miatt

**EDUS (Event Driven User Survey) Service:**

* [cwa-ppa-server/EDUS.md at main · corona-warn-app/cwa-ppa-server (github.com)](https://github.com/corona-warn-app/cwa-ppa-server/blob/main/docs/EDUS.md)
* This service will expose an API for handling One Time Passwords (OTPs). The role of this service is to redeem OTPs and to generate OTPs for testing purposes. During productive usage, the OTPs are created solely by the PPAC service.
* Jelenleg ki van kapcsolva (deployment replica 0-ra véve)

**ELS-verify Service (Error Log Sharing):**

* Ez a service ellenőrizni, hogy a logokat valós eszközről küldik-e
* kapcsolatban az log-viewer/log-upload-al

**Retention Service:**

* [cwa-ppa-server/RETENTION.md at main · corona-warn-app/cwa-ppa-server (github.com)](https://github.com/corona-warn-app/cwa-ppa-server/blob/main/docs/RETENTION.md)
* This service is responsible for the periodic database table clean-up.

*Privacy-Preserving Access Control (PPAC) Service* (Will follow soon.)

**CWA E2E Test:**

* Newman pod-al vizsgálja az endpointokat (GET és POST requesttel).
* curl post commandot használ = HTTP Post -> benne van a newmanjunitparser-ben
* Választól függően 3 opció lehetséges:
  + Success: Sikeres volt a test
  + Failure: nem sikerül futtatni egyáltalán
  + Error: futott, de hiba volt valamelyik lépésben

**CWA-Log-upload:**

* Ha a felhasználó hibát észlel az alkalmazás működésében, akkor eldöntheti, hogy megosztja-e és feltölti a kapott hibát, amit majd a fejlesztők ki fognak vizsgálni.
* ELS- verify ellenőrzi a készülék hitelességét.

**Quicktest:**

* Németországban különböző helyeken (üzletek, gyógyszertárak… stb) vásárolhatók lesznek gyorstesztek. A gyorsteszt megtörténne a vásárolt helyszínen, ahol egy erre jogosult személy fel is töltené az eredményt az appba. Ezt bemutatva be lehetne menni a helyekre (mozi, étterem…)
* schnelltestportal.de -> quicktest
* iam.schnelltestportal.de -> quicktest-iam
  + Keycloack, amiben IAM-et intézzük.
* **Quicktest-proxy:** a portal-al fog kommunikálni, ami egy nginx lesz, ez rakja le a testresultba azokat az adatokat amik a testrestulnak kellenek, a többi adat (személyes) két adatbázisban lesz tárolva, az egyikben 14 napig a másikban 10 évig lesz elérhető, amik azonnal titkosítva lesznek
  + quicktest-long-rds
  + quicktest-rds

**Registration portal:**

* Partnerek tudnának ide regisztrálni és feltölteni a szükséges infókat (szükséges papírok, név, contact...), mert eddig a kb 80 fős onboarding team csinálta.
* **Registration-portal-IAM** ugyanúgy mint Quicktest-IAM esetében
* This portal is used to provide a portal for the rapid test partners to register, upload the contracts, privacy statements and certificates. In a later step the process would be automated with ServiceNow connection. Up to now this was done via a functional mailbox, but since within 3 weeks more than 10,000 requests came in and that would be difficult to manage with 80 people, Digisol decided to build a portal for this.
* **Registration-portal-API:**
  + Endpoint amibe ha beadja valaki a megfelelő linket, akkor le lehet tölteni az összes registration portalon bejegyztett információt egy bizonyos időtől kezdve
  + export végű URL, ahova be kell lépni (keycloackos credentiolok loginhez),
  + CSV-be tölti le az adatokat

**Kontakttagebuch:**

* Napló, ahova az appon belül a userek felírhatják, hogy kivel találkoztak. Kb mint egy Notepad. Csak frontend-en jelenik meg.

**Map:**

* Tulajdonképpen egy Google térkép, ahol a felhasználók tudnak keresni olyan partnert akik gyorstesztet végeznek (akár DCC hiteles tesztet is).

**Map-Admin:**

* Map portal admin felülete
* Két letőség van a helyek felvitelére:
  + Ezen a map-admin portálon keresztül (CSV/Excel formátummal)
  + vagy most már az új deployal már a quicktest portal-on keresztül is fel lehet tölteni ezeket a helyeket, ahol tesztelnek
* Mass uploaddal is lehet feltölteni dolgokat
* és ha a quicktestportálon kitörlik, akkor nem fog megjelenni a térképen (mapon)

**Map-IAM:**

* Keycloack, amiben IAM-et intézzük. Teszthelyeket regisztrálunk ide.

**DCC verification**

* szeretnénk a tesztet validáltatni
* Verification server (tokennel) és testresult server alapján elküldi aláírásra a tesztet. A Partner (orvosok, laborok) hash-el aláírja, majd az IBM (hivalatos aláíró backend) is aláírja, és majd akkor érkezik vissza a warnApp-hez. PCR tesztek és gyorstesztek tényleg valósak-e.
* Ez van összekötve DGCG szerverrel -> feltöltött DSC-vel (Document Signing Cert) írják alá ezeket
* Oltási kiskönyv is hasonlóan megy (IBM fejlesztés).

**DCC-proxy:**

* a portal-al fog kommunikálni, ami egy nginx lesz, ez rakja le a testresultba azokat az adatokat amik a testrestulnak kellenek, a többi adat (személyes) két adatbázisban lesz tárolva, az egyikben 14 napig a másikban 10 évig lesz elérhető, amik azonnal titkosítva lesznek
* Azért volt rá szükség, mivel nem mindegyik labor akar a quicktesten keresztül kommunikálni.

**DCC rules:**

* It is downloading busines rules from the DCCG and acts as distribution point for the wallet apps and verifier apps
* no interaction with any other CWA component (Vault is of course needed)
* development done by DigiSol
* A tervezett National rule-ok helyett egyelőre CCL-config, ami egy újfajta megoldás, mint az eddig használt rule-ok esetében és tervezetten csak ideiglenes megoldás, amíg ki nem alakítják megfelelően a National rule-okat.

**DCC-validator:**

* ticket.io lufthansa... partnerek szeretnék legfőképpen használni
* IBM-el közös fejlesztés
* Eseményeken használható dcc-validálás, hogy ne a hatósági dcc-validatort kelljen használni hanem commercial validation legyen.
* Jelenleg nem tudják megállapítani, hogy tényleg valid-e a dcc
* Mindegyik vállalat kap egy külön namespace-t
* 1 nodepool lenne, ahol tervezetten node autoscaling működne
* Kapna mindegyik partner saját Grafanát is, valamint saját vault-ot is, ahol titkosítva tárolnák a partnerek a secreteket
* Redis-t használna a háttérben

**Digital Covid-19 Certificate Gateway (DCCG/DGCG):**

* <https://github.com/eu-digital-green-certificates/dgc-gateway/blob/main/docs/software-design-dgc-gateway.md>
* <https://github.com/eu-digital-green-certificates/dgc-overview/blob/main/guides/certificate-governance.md>
* szintén ACC, PRD és Test környezet
* 4 tomcat, mysql, load balancer (kb mint a sima EFGS)
* Különböző nemzeti backendekhez csatlakozik
* 3 fajta QR kódot lehet hitelesíteni az itt (adatbázisban) tárol certekkel:
  + Recovery
  + Test
  + Vaccination
* Validation rule-oka vihetnek fel az országok
  + Ezek olyan szabályok, hogy az adott országok miket fogadnak el a belépéshez (milyen oltást, PCR kell/nem kell...)
* Gateway amit csinál: certificateket ellenőrzi, hogy másik országé jó-e valamint publikálja ezeket
* Operáció részről a feladataink: kb ugyanaz lesz a feladat, mint EFGS esetében (key change-k működnek-e, országok onbordolása, cert-ek cseréje..., toolok ugyanazok, mint EFGS)
* Revocation List:
  + Az egész EU oltási kiskönyv az arra alapult, hogy fogtak egy kiállító certet, és azzal csinálták az összes oltási igazolványt, cserébe vagy az összes valid vagy az összes invalid amit azzal csináltak. Mert az EU 500 milla lakosáról nem tarthattunk egy egész sql sort, hogy ki kapott már QR kód-t, és ki nem. De mégis felmerült az igény, hogy 1-1 certet blokkolni kéne, úgyhogy minden cert valid kivéve a következők.... és ott jön a lista (revocation list), mert ez a lista már sokkal rövidebb lesz... valszeg nem 500 millió sor

**European Federation Gateway Service (EFGS):**

* Országok (MemberStates) oszthatják meg egymás között a fertőzöttségi adatokat. Ha az adott országban a fertőzött beleegyezik, hogy megosztja az adatait, akkor felkerül az EFGS adatbázisba, és a többi ország le tudja tölteni ezeket az adatokat.

**MTLS Kommunikáció a service-ek között:** ([Confluence link, ahol megtalálható a kép](https://seu30.gdc-bln03.t-systems.com/confluence/display/APPAGILE/mTLS+map))

**Diagram

Description automatically generated**

Várható

**Registration-portal új funckió:**

* Új gomb lesz a portalon, amivel hogy ha valaki eventet szervez és ott kiderül, hogy valaki fertőzött (volt/lesz), de az adott személynek nincs CoronaWarnApp-ja, akkor a szervező megteheti azt, hogy ő a saját applikációjában bejelenti az adott személyt. Bejelentést követően egy hivatalaos egészségügyi szervezet hozzáférhet ezen adatokhoz és ők fogják felkeresni a többi résztvevőt.

**Rule management service**:

* RKI által küldött rule-okat ne kelljen manuálisan feltölteni
* Keycloackban lesznek a userek kezelve (5-10 user) - 2faktor
* Bővítve lesz 3 faktorra - proxy segítségével (cert-el lesz megoldva)

**DCC-Rules:**

* kap egy GUI-t ami országokra bontva megjeleníti, hogy milyen rulok érvényesek. (Van már egy EU-s, de kell egy német.)
* külön namespace és endpoint
* Lesz még egy fajta rule:
  + German acceptance rule/National rule: Nem feltétlenül csak úgy lesz "oltott" valaki, hogy megkapott x oltást, hanem lehet csak kevesebb oltást kapott, de már volt beteg is.... stb

**Pcr proxy:**

* ugyanolyan mint a többi proxy, csak PCR-al
* február közepére készül majd fel a CWA app, addig PRD on hold

EFGS esetében ECDC fogja átvenni a kommunikációt Evelináék (Evelina, Georgios, Ander) helyett.

**Lesz új endpoint**

* vagy egy új proxy vagy quicktest-nek lesz egy másik service mögötte.
* api endpoint, ahova JSON formátumba fel tudják tölteni, és ez valahogy átkerül majd az analytics bucketba, és ebből csinálnak majd riportokat
* ügyfél március elején szeretné már live-ban