EINS

DP-3T Integration

14th July 2020

EINS By VIA

Introduction

This document details how the EINS system interfaces with the DP-3T contact tracing protocol, and other entities using such. It looks at this from a high-level architectural perspective before focussing in on how specific user interactions trigger particular flows and how those flows work. Lastly, it takes a look at all third party services and how these are kept separate from the contact tracing solution.

What/Who Is DP-3T

As defined in the DP-3T documentation, DP-3T is:

"a international consortium of technologists, legal experts, engineers and epidemiologists with a wide range of experience who are interested in ensuring that any proximity tracing technology does not result in governments obtaining surveillance capabilities which will endanger civil society."

It is independently funded by Prof. James Larus's discretionary funds at EPFL and in no way is funded by either Google or Apple. For more information on the individuals involved in the project examine the readme file in the DP-3T Github repository.

DP3T has come to refer to the decentralized tracing protocol, in which contact keys can be managed/stored/exchanged in such a way to preserve maximum privacy of all users.

System Architecture

There are two main parts to any system that make it DP3T compatible. This is the mobile application (or more accurately the DP3T SDK, run completely on the user-owned device) and the positive key storage API/server.

Both the EINS application and EINS API code are proprietary and will soon be open-sourced for public viewing on the Via Data Github organisation.

The DP-3T open source code for the mobile application SDK and the backend API/Server can be found on the respective DP-3T organisation.

These systems were developed separately from each other and are integrated using software development kits (SDKs) created by DP-3T. The mobile application SDK for DP-3T runs within the

actual EINS mobile application itself. All contact tracing functionality is managed by the DP-3T app SDK, the EINS application simply processes the information returned by the SDK in order to provide an improved experience for the user.

Each standalone contact tracing application (at this stage, each country) operates its own DP-3T backend positive test key management server. This server receives the positive contact keys from app users who test positive, and disseminates them to app users to check for contact with infected persons. There is a server for Switzerland, operated by SwissCovid on behalf of:

The Federal Office of Public Health of Switzerland

Schwarzenburgstrasse 157

3003 Berne

Switzerland

EINS will employ the strategy of utilizing these servers per country where possible, to ensure less fragmentation of the key management backends. However, it is planned that EINS could own and maintain a server to ensure compatibility with other systems into the future if needed.

There is no direct interaction between the EINS server API and these DP-3T APIs in any way. The DP-3T server APIs interact with the DP3T mobile application SDK deployed within the EINS application, as well as other DP-3T servers as required by interoperability as detailed in this specification.

Any DP-3T server requires a database to store all information that is passed to it. This database is hosted by each owner.

Configurations

In order to operate a DP-3T system, there are certain items that need to be configured. These are as follows:

- **Server Key Pair**: This must be updated from the default values in order to ensure the security of the DP-3T system.
- **Server Data Retention Period**: This by default was set to 21 days. In order to comply with the widely regarded standard recommended by health authorities, this was changed to be 14.

- **Server Database Details**: The server must point to its own database for proper functionality. The database was created on the EINS system architecture and these database details entered into the DP-3T configuration.

Contact Tracing Flows

Now that a higher-level overview of the DP-3T integration has been detailed, it is important to look at some of the flows triggered by the user work in order to gain a deeper understanding of the interaction between EINS and any DP3T based backend server, in this case SwissCovid.

Device Contact is Logged

As soon as contact tracing is enabled, this process starts happening in the background of the application. A full detailed explanation of this process can be found in the DP-3T documentation. However, in brief, the following process occurs:

- The device activates a Bluetooth beacon that is detectable by other devices, by enabling the device level Apple/Google APIs.
- The device generates unique (ephemeral) identifiers that are rotated routinely based on time. These are generated by the DP-3T app SDK and broadcast through the beacon.
- Should the device come into contact with another supported device, both devices store each other's identifiers alongside a rough timestamp on the device. This process is completely managed by the DP-3T application SDK.
- Note that all information stored on the device is done through DP-3T's own cryptographically secure storage solution meaning that only DP-3T is able to access this information through the mobile app SDK.

As can be seen in the above flow, this process is entirely managed by the DP-3T SDK with all data involved kept within either the device OS or the DP-3T SDK or Server. None of this data is processed or managed directly by any EINS code in any way.

Positive Test Flow

This flow is triggered on the EINS application by the user updating their COVID-19 status to positive (and verifying this positive test). Once again a detailed explanation of this process can be found in the DP-3T documentation, a brief high-level description of the flow is:

- Once prompted by the EINS user via the EINS UI, the DP-3T app SDK pulls together all ephemeral Ids on the device for the time that the user has been contagious.
- The user is required to insert a verification code given to them from the health professional of the region in which they were tested positive. This code in Switzerland is known as a CovidCode. Each territory's codes provide the required authentication to upload to the relevant DP-3T server, operated by the relevant authority (in this case SwissCovid).
- The DP-3T app SDK then prepares the positive keys along with the auth code, and sends it to the relevant DP-3T server. In the Swiss case, operated by SwissCovid.
- The SwissCovid DP-3T server stores this information in its central database that contains a list of all Ids that are contagious.
- All devices on the system (and other systems via server interoperability coming soon to the DP-3T network) pull data from the server on a regular basis and check to see if any of the Ids stored on the local contact log match any Ids pulled from the server. This all happens within the DP-3T mobile app SDK.
- Should a match be found that meets the criteria for user notification, the DP-3T SDK returns a notification to the EINS mobile app which in turn passes this on to the user.

As can be seen above the DP-3T source code manages the entirety of this process, and the responsible party for each step is either EINs or the DP-3T server operator (SwissCovid). The EINS application simply initiates it and passes on any notification of exposure risk to the user.

Contact Tracing Responsibility Scope

See below for a table view on each step of the contact tracing process and which parties and external services are responsible for each step.

Device Contact Logging

This table follows the flow for general contact logging of nearby devices on the applications itself.

Action	Location	Responsibilities
User gives permission to the app for tracing purposes	Device	**EINS:** Initiates DP-3T tracing functionality based on Apple & Google device APIs

Device generates its ephemeral Ids that it broadcasts	Device	**Apple & Google OS**
Device listens and logs other nearby device information	Device	**Apple/Google Tracing:** Responsible for the low- level technical implementation of the actual Bluetooth interaction **EINS App DP-3T SDK:** Responsible for the storing and handling of information of all logged contact keys.

Positive Test Flow

This table follows the flow for submitting device tracing information to the DP-3T server upon a user testing positive and choosing to notify other users.

Action	Location	Responsibilities
Ephemeral Ids during contagious period are sent to a DP-3T server	Device -> Server	**DP-3T SDK:** Used for the collection and transmission of Ephemeral Ids **EINS App:** Responsible for conveying the contact codes to the relevant server **SwissCovid DP-3T Server**: Responsible for accepting and storing the contact codes
Server stores contagious information	Server	**SwissCovid DP-3T Server:** Responsible for the processing and storage of the data

User Exposure Check

This table looks at the process of the application regularly checking contagious device info and checking against the local contact list to check user exposure risk.

Action	Location	Responsibilities

Device requests contagious information from server	Server -> Device	**EINS App DP-3T SDK:** Initiates the data request on the application side. Also responsible for the authentication and sending of data on the server-side **SwissCovid:** Responsible for the hosting of the DP-3T server.
Comparison of pulled data to locally stored data	Device	**EINS App DP-3T SDK:** Responsible for comparison of pulled information to locally stored information. This involves calculating whether a user has been exposed or not.
Notification of exposure to user	Device	**EINS App DP-3T SDK:** Initiates notification process based upon exposure calculation **EINS App:** Passes notification onto the user