# Ketju-project - Final Report

Version 1.1, 18.6.2020

**Initial distribution**

Sitra
Vaasa Hospital District
2M-IT Oy
Fraktal Oy
Futurice Oy
Reaktor Innovations Oy

# 1. Introduction, background and participants

Large scale social distancing, quarantines and other emergency measures due to the COVID-19 pandemic are costly both financially and on a humane level. The goal of the officials in Finland is to start opening society in a manner that prevents the virus from spreading uncontrollably but would still allow society to function normally.

**The track-test-isolate-treat** model has been chosen as part of the national strategy in Finland. Technological means of improving the manual tracking process is part of the strategy. Sitra, the Finnish Innovation Fund funded the Vaasa healthcare district in the beginning of April to pilot the use of a mobile application that senses the proximity of other mobile applications through the use of Bluetooth technology. The application was based on the de-centralized contact tracing model. The goal of the pilot was to test the Bluetooth technology in contact and proximity tracing as well as the use of a contact tracing application in real-life situations.

During the first phase of the project, different Bluetooth protocols and contact tracing models were evaluated. Many of the technologies were new, so both the protocols and the concerns related to legislation and privacy were shifting rapidly. Some of the evaluated Bluetooth protocols were abandoned at an early stage as they were considered unfit for EU and Finland (e.g. the centralized models like Singapore's Bluetrace and the European PEPP-PT). In April, the most promising protocol was the European DP-3T that was chosen as the basis for the pilot application.

DP-3T is a de-centralized model that emphasizes the users' data security and data protection, and the users can stay completely anonymous.

The project was started, financed and coordinated by the Finnish Innovation Fund Sitra. Other participants were the Vaasa Hospital District and IT companies 2M-IT, Fraktal, Futurice, Columbia Road and Reaktor. During the project, the Finnish Data Protection Ombudsman was informed and the National Cyber Security Centre was consulted in matters concerning data security and data protection.

# 2. Project schedule

The project schedule was heavily dependent on the schedules of different Bluetooth tracking protocols and their publication, and was therefore not set in the beginning of the project. However, the project was completed in a timely manner and the pilot was completed mostly during the set schedule in May.

| **Date** | **Project phase and tasks** |
|---|---|
| 24.3.2020 | Preparation and different technical approaches evaluated |
| 8.4.2020 | Project started, evaluation continues, technical development started |
| 30.4.2020 | Pilot planning started, technical development |
| 4.-15.5.2020 | Pilot participants briefed, detailed pilot planning, technical testing |
| 18.5.2020 | Pilot started |
| 5.6.2020 | Pilot ended |
| 15.6.2020 | Final report |

*Table 1: Project schedule*

# 3. Technical solution and its main functionalities

## Glossary

The application produced during the project includes several specific technical concepts relating to the solution and the chosen tracking model. This glossary describes the main concepts that are used in this document and in the context of tracking applications.

***Handshake*** *is a single, short-term detection between two devices that are close enough to each other.*

***Contact*** *is a longer period of connection between two devices. A contact consists of a sufficient amount of handshakes. A contact can be single-sided, and it must fulfill pre-set criteria for signal strength and duration.*

***Exposure*** is a contact that is deemed to be sufficient due to its strength (duration, distance) to lead likely to an exposure of the covid-19 virus. The exposure criteria are originally epidemiological but have been defined as technical parameters in the mobile application.

**Anonymization** means that a user of the system is represented as a meaningless, typically randomized string of characters that cannot be linked to the user's identity without additional information. This information can be held implicitly in the user's mobile device and will be shared outside the device afterwards, or held in a database outside the device.

**ID** in this context relates to an anonymized string of characters described above. The mobile applications exchange these IDs between themselves and based on these IDs the applications can later recognize possible exposures to mobile applications that have been marked in the system as having been diagnosed positive for covid-19.

**Centralized tracking models** contain a database that makes it possible to identify each user of the system explicitly.

**De-centralized tracking models** do not have a database that can identify users. Even if a user is diagnosed with covid-19, the contact data with other users is kept anonymous and the user's identity is not shared.

***Hybrid tracking models*** do not have databases that would identify users unless the user later voluntarily consents to sharing their information in the case of an exposure, thereby creating an exposure-chain. If a user chooses not to share their information after an exposure, they remain anonymous.

In practise, the difference between centralized and de-centralized tracking models are not that simple. There are many different hybrid solutions that fall between these two models.

## Choosing the tracking model for the pilot

As the covid-19 pandemic spread globally in the beginning of 2020 many countries started planning and implementing mobile applications and tracking protocols to observe exposures to the virus. These solutions utilized Bluetooth connections between mobile devices and especially the possibility of estimating the physical distance between mobile devices based on the attenuation of the signal strength over distance. New protocols were introduced almost weekly during late March - early April in 2020, so the project was concentrated on evaluating and re-evaluating these technical protocols before making the final choice.

The BlueTrace protocol developed in Singapore was one of the first tracking solutions that came into the public in the middle of March, and the codebase was made public in the middle of April. In the beginning of the project, BlueTrace seemed to be a good basis for the technical solution as practically the only protocol publicly available and first designs were started. Since it is a centralized tracking model, it also provided efficient tools to help the manual tracking of exposures for the health officials.

In the beginning of April, a European choice became available: the PEPP-PT protocol developed in Central Europe and led by Germany became public generating wide interest and sharing an early version of its codebase even before BlueTrace. PEPP-PT had also taken into consideration the compatibility between different countries within Europe offering additional value over BlueTrace.

However, soon after PEPP-PT’s appearance, a group of researchers led by Switzerland diverged from PEPP-PT and published their own tracking protocol called DP-3T that focused on the data privacy and anonymity of the users. Growing concern over the privacy issues of tracking applications in the public debate also affected the choice of protocol in the project. Additionally the fast and open development of DP-3T protocol was seen as an improvement over the closed development of BlueTrace and PEPP-PT. As a downside, the anonymous de-centralized tracking model offered less efficient tools for helping the manual tracking process.

Finally, in the end of April, a large international group of researchers published an open letter expressing their concern over the data privacy and possible abuse of the information collected through centralized tracking models. The letter led most researchers to back the DP-3T protocol.

All the different Bluetooth-protocols had problems with the application being limited or even non-functioning while running in the background, especially on iOS mobile devices. Possibly for

this reason, Apple and Google ended up developing their own de-centralized exposure notification protocol that seems to be heavily influenced by DP-3T. Apple and Google could develop the protocol on the operating system level, possibly solving the issue for running in the background and allowing for a more secure way to store the contact data. The protocol was publicly announced in the middle of April, but the release for use was set for May, which made it too late for the Ketju project pilot. DP-3T was chosen as the protocol for the pilot application.

Additionally, both Apple and Google restrict the use of their Exposure Notification protocol for only one application in each region, requiring a mandate of the health officials in that region or country.

**The main properties of the evaluated tracking protocols:**

BlueTrace

- User's contact information is stored on a centralized server during onboarding.
- Periodically changing contact IDs are fetched from the server.
- Information about contacts are initially stored only locally.
- In the case of a positive covid-19 diagnosis, the diagnosed user can share their contact information with the health officials, who can create an exposure chain based on the data on the centralized server and approach the exposed users.

PEPP-PT

- Equivalent in functionality with BlueTrace.
- Possibility of interoperation between the applications in different countries.

DP-3T

- Onboarding does not require users to register any personal information. No technical registration to the backend is made either.
- Periodically changing contact IDs are created locally.
- Information about contacts is stored only locally.
- In the case of a positive covid-19 diagnosis, the diagnosed user can anonymously share their contact IDs with the backend system. Other users' applications can use this information from the backend system to locally identify possible exposures from their own application's contact ID history and so become aware of the exposure and notify the user.

Apple ja Google Exposure Notification API

- Equivalent in functionality with DP-3T.
- Presumably better support for the application running in background, especially on iOS.

# Pilot application functionality

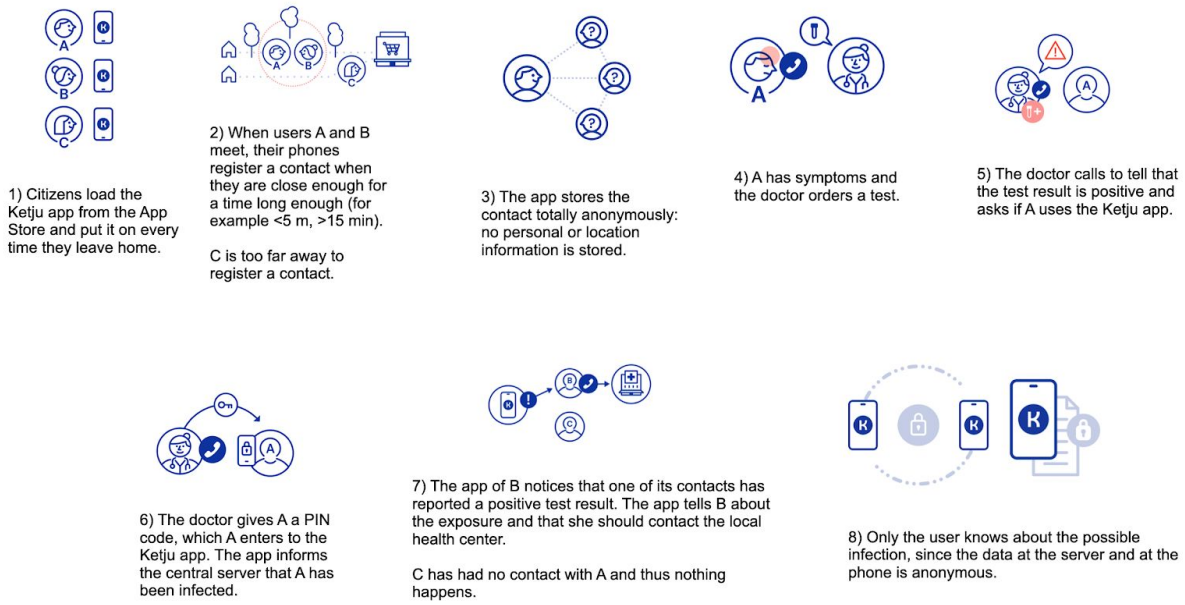


*Image 1: Pilot application functionality overview*

The DP-3T protocol used by the Ketju application, like all the other evaluated protocols, is based on using wireless Bluetooth technology in detection of close proximity to other mobile devices running the same tracking application.

The applications continuously send their own short term IDs that change every 15 minutes and are derived according to a specific cryptographic method from secret keys that change daily. These daily keys are derived from the previous day's key starting from a random secret key that is created at the time of onboarding. These short term IDs are seemingly random to outsiders and don't reveal the user's identity. An outsider is not capable of predicting a user's previous or future keys based on knowledge of the current one, blocking the possibility of following a user by tracking their Bluetooth traffic for a longer period of time.

Other mobile devices running the application within the range of the user's Bluetooth signal listen for and collect these short term ID's and save them in their local database as handshakes, including the timestamp and signal strength of the received signal and reported transmission power of the sending device. The difference of these two yields the attenuation of the signal between the devices and an estimation of the distance can be derived.

Enough handshakes with the same short term ID will result in contacts with a specific duration and an average signal strength that can be used to make an estimation of the average distance during the contact.

When a user of the application receives a positive diagnosis of covid-19, they can choose to share that information with other users of the application and thus have them be notified of possible exposures through the application. From a technical perspective, the sharing of a positive diagnosis is implemented by publishing the secret keys of the infected user starting from the first day of the user's estimated infectiousness. The other applications can then derive all the used short term IDs and compare those to their own internal contact database. If a user's daily contacts contain enough contacts that fit the distance and duration criteria to fulfill the parameters for exposure, the system can conclude that an exposure has happened. The identity of the infected user is not revealed to the exposed user, nor that an exposure has happened to anyone else than to the exposed users themselves.

Both the positive diagnosis of covid-19 and the first day of infectiousness is always defined by a health care professional.

## Technologies and architecture

The mobile applications were developed as native applications by using the typical technologies and software development practices for each operating system. The programming languages were Swift (iOS) and Kotlin (Android). The tracking functionality was implemented by using the open source DP-3T library. Some changes were made to the DP-3T library in order to identify pilot users from handshake and contact data. This allowed comparing the data gathered by the applications to the manual notes of contacts some of the pilot users collected to ensure that the application and the chosen protocol were working properly.

The back end system was developed by using modern technologies (TypeScript, Node.js, PostgreSQL) and the infrastructure was located in Google Cloud Platform Hamina data center.

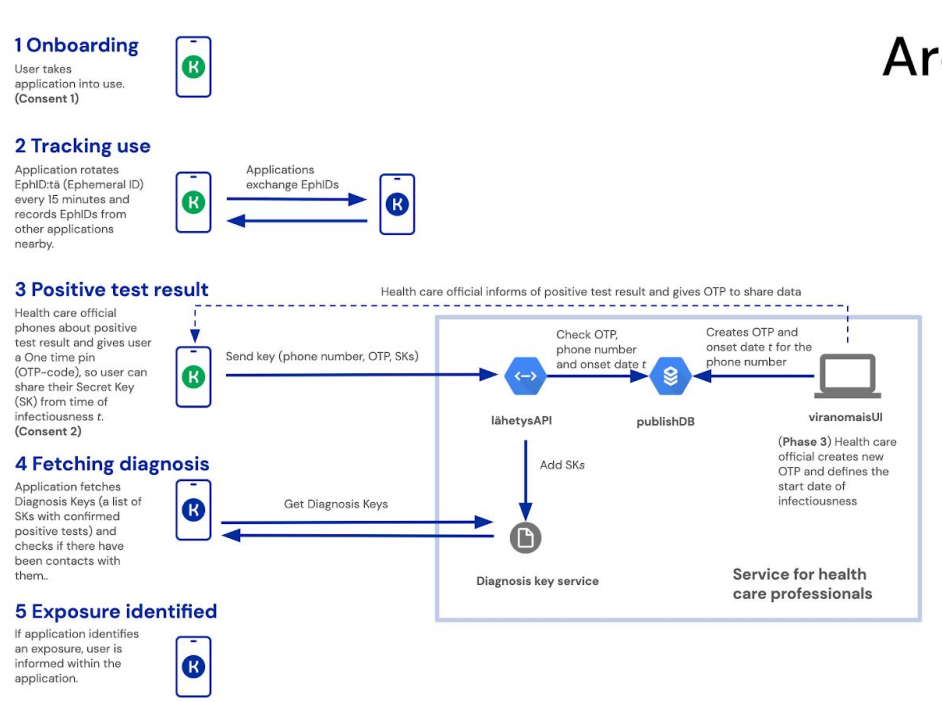


*Image 2: Overview of solution architecture*

# 4. Main principles of the application design - the user interface and visual appearance

When designing the user interface, appearance and user experience of the Ketju app, the project team was guided by established user-centered design principles and the specific goals of the project. Since the basis of the app is quite technical and the presumed user-base is rather large and contains subgroups whose understanding of technology varies greatly, the main principles for the design were defined as clarity, accessibility, ease of use, trustworthiness and humanity. This meant among other things that

- established conventions were used when designing the UI functionality
- all texts have sufficient size and contrast to ensure their readability
- it was ensured that visually impaired users can successfully use screen readers or other supporting technologies to interpret the content of the application correctly and to interact with it
- different sizes of displays, phone models and operating systems were taken into consideration
- the screens were designed to be serene, the texts guiding the user to be terse, and the guiding images to be unambiguous
- the appearance was designed as visually polished, balanced and consistent, for example all the communications related to the application had to be recognizable and adhere to the brand
- although the application is critical for the society, important and official, it had to be humane and easy to approach


*Image 3. An example of the visual appearance.*

# 5. Data security and data protection

It was known already at the beginning of the project that contact tracing applications can cause huge data security and data protection risks for individuals, if these risks are not taken into account in designing the architecture and features of the application, choosing its technologies, and ensuring the high quality of its implementation. It was imperative for all the parties in the Ketju project that the application would become secure from the viewpoint of its user. The data protection ombudsman of Finland was regularly informed about the aspects of the application design.

To guide data security and data protection, a set of principles was defined in the beginning of the project. The project team took security threats into account by identifying and using threat scenarios as a part of the implementation process.

## Guiding principles

The following principles guided the data security and data protection architecture of the Ketju application.

### Anonymity between users

The application must not include any way to expose the identity of a user to another user. The Ketju mobile apps exchange temporary ids that are changed many times during the day. Following the DP-3T algorithm, these ids are derived mathematically from the internal id unique to each installed application using hash functions that do not permit to deduce the internal id based on the temporal ids.

When an application user gets a positive COVID-19 diagnosis from the health officials, she may voluntarily send her own id to the backend system. The apps of other users fetch these ids regularly and can thus inform their users if there has been a contact with the id of the person fallen ill. The anonymity is not endangered in this scenario, since no personal information nor the exact time of the contact are revealed to the user in the exposure message.

### User anonymity towards system administrators

The application offers no means for system administrators to connect real identities of users to ids that have not been voluntarily published to the backend in the case of a positive covid-19 diagnosis.

Ketju mobile apps collect, process and store all the contact information independently and send none of this information to the backend service.

The mobile apps update regularly their list of infected user ids from the backend system. The comparisons of the infected ids to the contacted ids happens internally in the mobile app.

### Minimizing data processing

The app processes only the data that is necessary for deducing the length and distance of user contacts.

Each application stores its contacts only locally in the mobile device. The app does not use nor store any location information based on satellites (GPS) or wifi positioning. The contact data is permanently destroyed when the app is removed from the mobile device, and it cannot be restored from any backend system.

The contacts are stored only for a limited time (3 weeks in the DP-3T implementation), after which the application deletes them automatically. This ensures that the app does not store any contact history unnecessary for tracing infections. In addition the pilot application contains a kill switch that terminates the application on July 1st, 2020 latest, if the pilot users won't remove the app after the pilot study has ended.

The users cannot be deceived to expose any part of their contact information, since the application offers no way to open nor copy the data.

The users are identified only based on pseudonymized ids. As has been described above, a user's real identity can only be connected to the id if the user has allowed it voluntarily after an infection.

## Identified threat scenarios

When the system was designed, a considerable amount of effort was put into identifying threats and vulnerabilities endangering data security and data protection. The list below is a summary of the identified threats. Each realized threat could lead to exposure of confidential information, unauthorized changes to information, or denial of use of the data or the service. The threats were managed through a variety of measures, including taking data security into account in system architecture, technology choices, and procedures for programming and testing.

### Threats to the mobile app

The project team identified the following threats to the mobile app and the device:

- Unauthorized access into the app, when the device is not locked.

- Unauthorized access to the app's resources, like the possibility to turn off Bluetooth.
- Decompiling the app and changing its behavior, like changing network operations or faking ids or copying them from one device to another.
- Exposing internal information of the app.
- The confidentiality of the logs and analytics of the app.
- Deceiving the user to hand over information or to change it.

## Threats to communication between systems

The project team identified the following threats to the communication between the mobile app and the backend system and the communication between mobile devices:

- Eavesdropping and altering the data communications between the mobile app and the backend system.
- Eavesdropping, altering, repeating and transferring the BT communications of the mobile apps.
- Cross positioning the location and movements of the mobile apps based on their BT communications.

## Threats to backend APIs

The project team identified the following threats to the backend system APIs:

- An attempt to corrupt the data in the system.
- A denial of service attack.
- Exposing personal information, like a person's diagnosis, its date and time, or the exposed persons.

## Threats to the health authority UI

The project team identified the following threats to health authority UI:

- Unauthorized access to the UI and the information it presents.
- Unnecessary viewing of personal information.
- Unauthorized changes in data.

## Threats to software development and distribution

The project team identified the following threats to software development equipment or software distribution.

- Dependencies of 3rd party libraries.
- Unauthorized changes to code.
- Changes in compiled components.

- Distribution of a fake app.

### Threats to backend administration

The backend of the app was implemented using public Google Cloud Platform services in the data center of Hamina. The project team identified the following threats to the administration level of the service:

- Unauthorized access to resources, like the database.
- Unauthorized changes to cloud resources, like deleting resources or information.

## Comparison to recommendations by authorities

The European Commission and the European Data Protection Board have released their own recommendations for applications supporting the prevention of the COVID-19 pandemic during Spring of 2020.

The project team assessed the recommendations of the European Commission and the European Data Protection Board as well as the applicable legislation (like the EU General Data Protection Regulation (GDPR) and the Finnish national legislation) for the application. Based on the results, the architecture, features and principles for data processing of the Ketju app follow the legislation and the recommendations. Depending on the chosen principles, the national app can be implemented in a way that it fulfills the requirements of the applicable legislation and the possible updates to it.

## The fulfillment of data protection rights

The EU General Data Protection Regulation (GDPR) guarantees certain data protection rights to a registered person. Based on the results of the assessment in the project, the Ketju app is in compliance with these rights. The user may also manage her own information by removing the app from her mobile device, which removes also all the information the app has stored and cannot be restored from any backend system.

# 6. Communications

Clear, inspiring and trustworthy communications is crucial for distributing a contact tracing app and acquiring users for it. There were two important target groups for communications of the Ketju app pilot test: the pilot participants using the app, and the wider audience of Finnish citizens that have heard about the pilot through the media. The goal was to tell why it's important to track contacts digitally and to convince the target group that the app fulfills the strict security and privacy requirements.

In the interaction between the app users and the media, the citizens' concerns about the privacy and security of the app were pronounced. The following questions were repeatedly answered in different channels:

- Do the users see their own or someone else's contact history?
- Does the app monitor or record its user's location?
- What information about the users can the authorities see?
- Does the app significantly increase battery use of the device?
- How many users are needed to make the app useful for containing the epidemic?

These frequently asked questions were answered in internal commications for the pilot users, in the media as well as in the ketjusovellus.fi web page.

There was a particular challenge in designing the communications for the pilot users concerning Bluetooth and location information. The app does not monitor nor store the user's location, but in the Android OS it's not possible for an app to request to use Bluetooth without the user allowing the app to use location information, as well. This had to be explicitly pointed out in the material meant for the pilot users. The communications have to be in line in the application, the marketing materials, and in the media. It was important to follow closely what the media wrote about the subject and to influence that the media had the most up to date information from the widest possible range of aspects, though the field is extremely large. The notifications sent by the app are a part of the communications and the impressions they produce are essential.

Since the pilot participants were arranged beforehand, the project produced no expreriences on user acquisition.

# 7. Pilot - description, observations, results

## Pilot arrangements

The pilot participants consisted of 33 voluntary users from the Vaasa Central Hospital emergency department and the director of Vaasa Hospital District. The participants worked according to their normal schedule in three shifts at the department during Monday 18th May - Friday 5th June, 2020. The devices they used were iPhones and Androids of several different models and different versions of the OS.

Four participants of the pilot made notes manually of the contacts that happened during their shift in addition to using the app. These four users got “sick” during the pilot, which allowed the development team to see how the the exposure notifications worked (when did the users get them, did they notice them etc.). In the pilot, a contact leading to exposure was defined as at least 15 minutes per day at a distance of 4,5 m or less.

The pilot app was based on the distributed model and DP-3T protocol, which uses Bluetooth to detect contacts and stores the contacts distributedly and anonymously only on users’ own devices. Hence the pilot participants were asked to send the contact information from their apps regularly to the development team.

The make the analyzing the results easier, the DP-3T implementation of the pilot app was in debug mode, which permitted to store a permanent unique identifier of each detected device. This allowed the development team to see all the participants’ contacts and to compare them with the manual notes. If the DP-3T implementation had been in production mode, only the contacts leading to exposure would have been found, and the granularity would have been a day (24 h).

### Device variants used in the pilot

iPhone (18 pcs)

- models 6, 6s, 7, 7plus, 8, XS, XR, 11
- OS versions 12.4.5, 12.13, 13.3.1, 13.4, 13.4.1

Android (17 pcs)

- models Samsung Galaxy A50 / J5 / S8+ / S10e, Nokia 8, OnePlus 5 / 6T, Huawei P30 lite, Huawei Nova 3
- OS versions Android 6, 8.0, 9, 10, OxygenOS 9.0.11

# Observations and results

When trying to generalize the results of the pilot, one should take in to account its small size, the specifics of the testing environment and especially the chosen technology and the model for contact tracing. The pilot was done at a time when is was not yet possible to use a DP-3T version based on the API of Google and Apple. Thus the results are valid only for a distributed model application that is based on DP-3T's own implementation of Bluetooth contact tracing.

**Measurements:**

- The app worked in real use and all the participants' apps were able to collect data as intended.
- There were 33 pilot participants from the Vaasa Central Hospital emergency department and the director of Vaasa Hospital District.
    - Two participants had two different devices (iOS, Android) with them during the whole test. (The duplicate devices have been removed from the analysis where necessary.)
    - There was an extra phone in the break room that had the app always on for connectivity testing.
    - In addition, a member of the development team visited Vaasa with two different devices (iOS, Android) which also collected data.
- The participants sent data of their contacts to the development team in total 240 times.
- There were more than 6200 contacts during the pilot.
    - About 1400 contacts between the two devices that the same person was carrying.
    - About 850 contacts between a participant and the phone in the break room.
    - About 4000 contacts between pilot participants.
- Four participants made manual notes of all of their contacts that they estimated to last more than 10 min at a distance less than 5 m. These contacts were later compared to the ones obtained from the app.
    - There were about 250 manually registered contacts.

## Results

- Images 3-8 offer an intuive idea of the nature of the contacts.
- There was a lot of variation in how the devices detected each other:
    - There were quite a few one-sided detections of another device. The amount of one-sided detections is a relatively good measure of detection inaccuracy, but it is also affected by the contact time limit. The small amount of iOS devices in the pilot and the issues with running the app in the background in iOS make the the big picture harder to grasp. When using the most strict 5 minute time range criteria, about 37% of the contacts detected by Android phones were one-sided,

the remaining 63% were detected by both sides. When the criteria for asyncronous detections is loosened to 0.5-1 h, two-sided detections rise to 70-75%. iOS-iOS contacts are registered perhaps on the same level, but there are too few events to make definite conclusions (image 10).
    - The problems with the iOS app running on the background have most likely reduced the amount of iOS-iOS contacts, making them more rare than Android-Android contacts (image 10).
    - Occasionally the devices had very different interpretations of their distance (based on signal strength), like 1 m vs. 4 m (image 11). It would be very laborious to take these device-specific differences into account with calibration parameters, since there are so many different device models and sometimes the components and configurations of a phone model depend even on its batch.
- So called false positive contacts through walls and other structures did not turn out to be a significant issue in the pilot. It has to be noted though that in an application like this, one must always make a compromise between ghost contacts (false positives) and unnoticed contacts (false negatives): reducing one increases the other. The balance has to be defined during development through the boundary parameters for a registered contact.
- Increased battery use did not seem to be a problem for the participants.
- There were some problems with Bluetooth connectivity with other devices (sports watches, car radios) when the app was running.
- When the app was running in the background, it did not always send a notification of exposure.
- The users did not always notice the notifications of exposure.
- The pilot app did not register contacts, if the user had not allowed it to use location data in Android. The other devices noticed these devices normally.
- It turned out to be very difficult for the users to make manual notes of their contacts on a sufficiently detailed level: when did the contact occur, how long did it last, who were there.

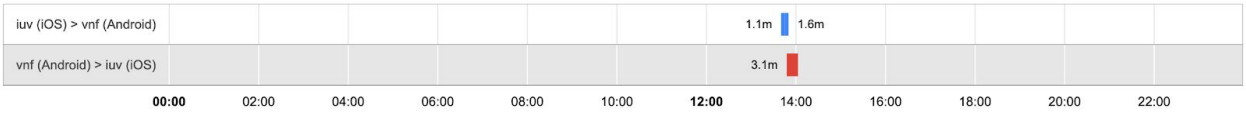


*Image 3. Two-sided contact with a small asynchronity in time (see image 10).*


*Image 4. One-sided and two-sided contacts.*


*Image 5. Android devices detect iOS devices less frequently than the other way around (see also image 9). This is most likely due to the issues with running in the background on iOS.*


*Image 6. One-sided iOS contact are numerous: iOS has detected the Android device but not vice versa. (Other timeline is not depicted as it would be empty).*


*Image 7. A probable false positive (middle row).*

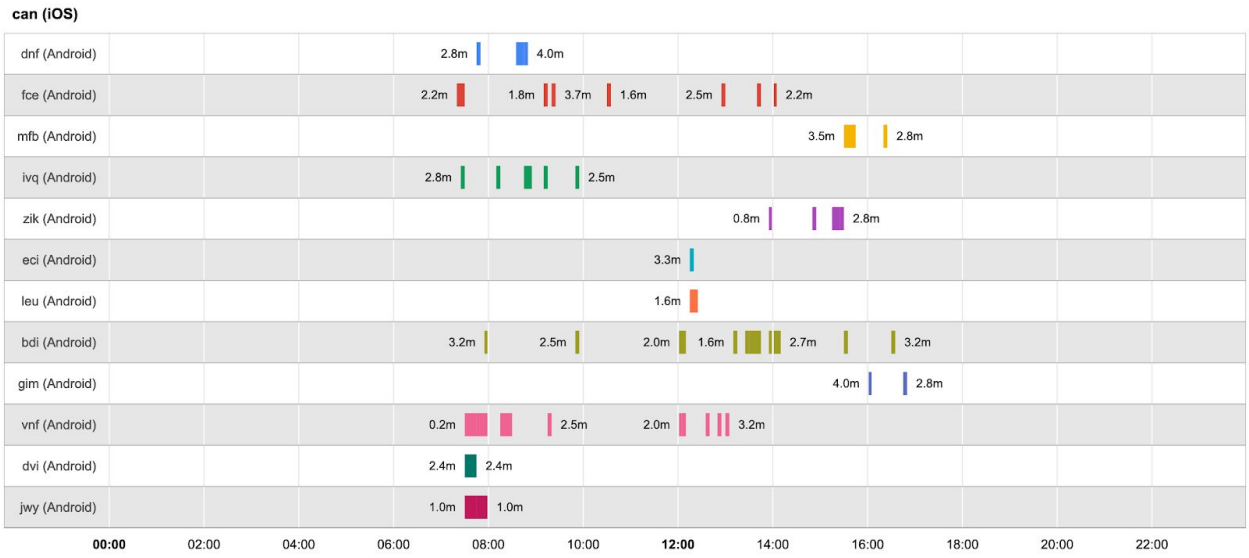


*Image 8. Single user's contacts during a work shift.*

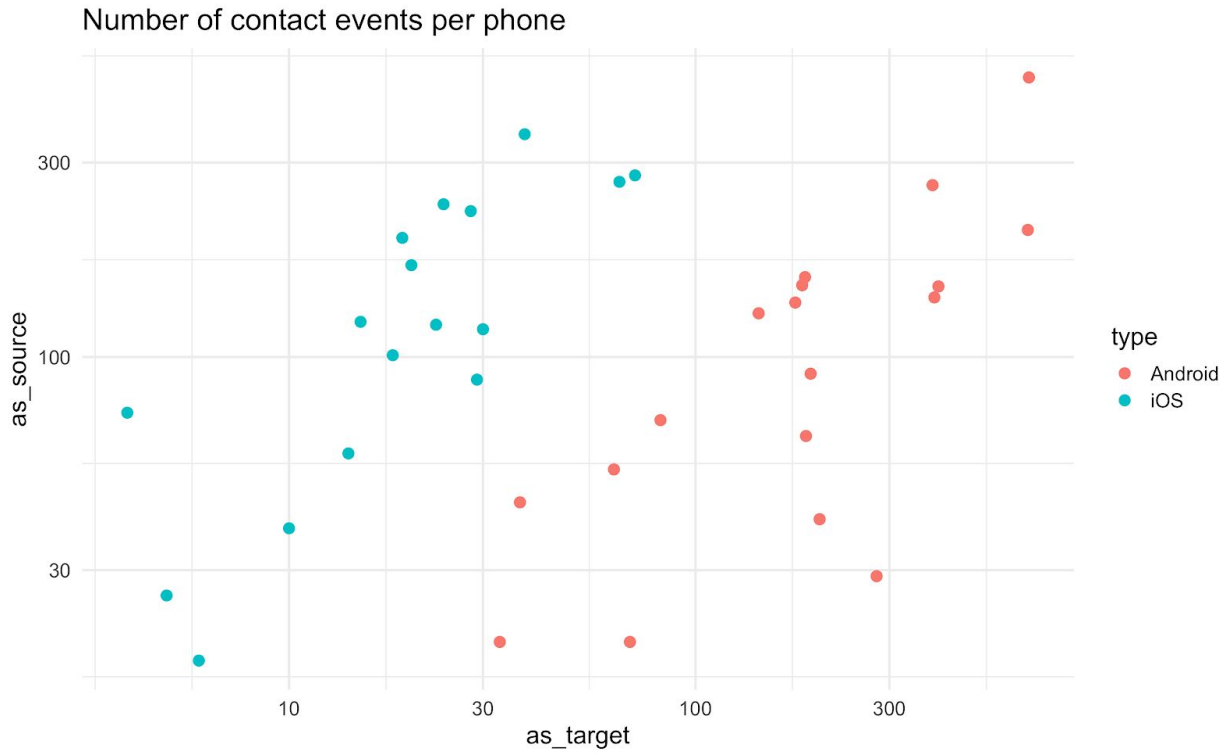


*Image 9. The contacts registered by devices varied significantly (the axes are logarithmic). The skewed distribution of the number of contacts most likely signifies the sensitivity differences between antennas and other technical properties, but they might also be a result of the differences between human behaviours. iOS was detected less often as a target due to the issues with the app running in the background (x axis). iOS devices were able to detect all types of other devices well (y axis).*

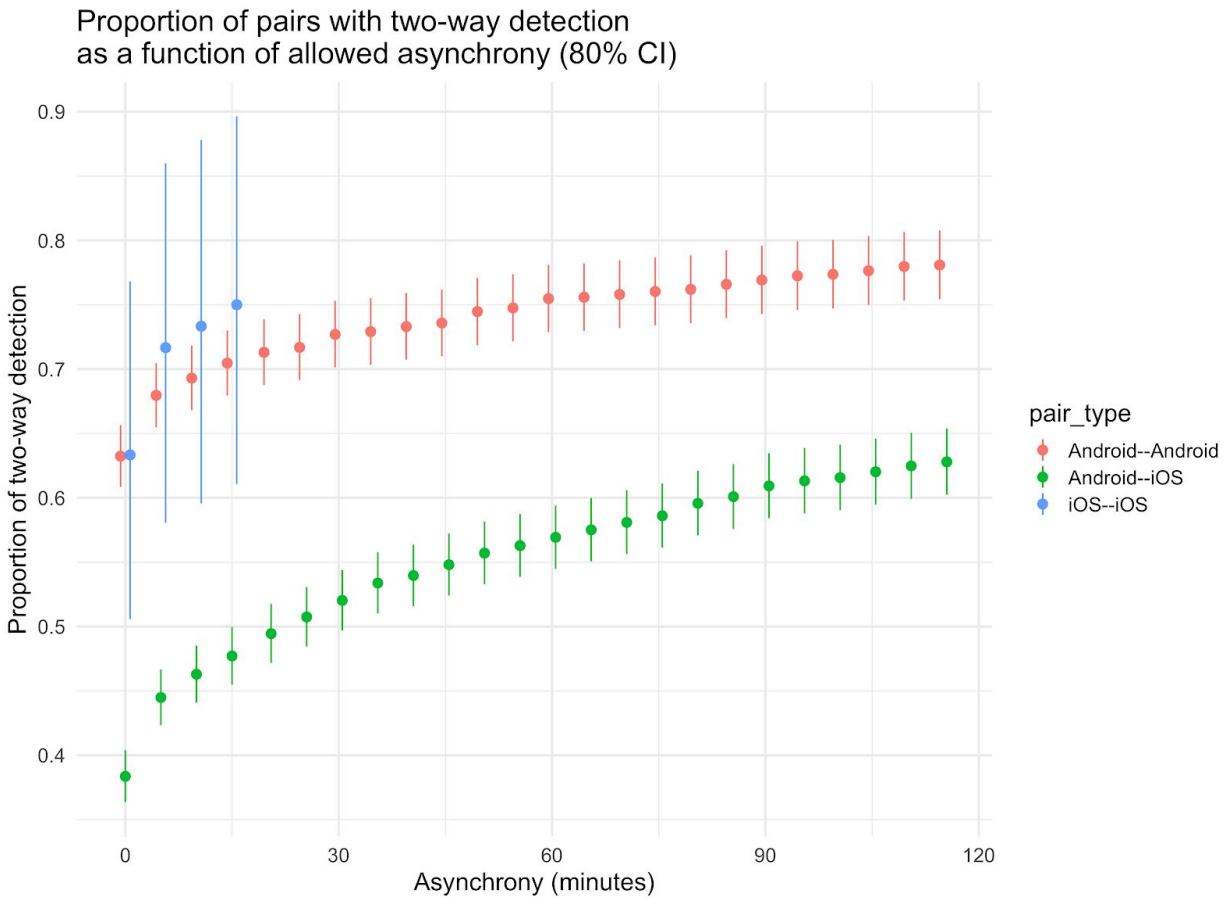


*Image 10. In a contact, both phones should detect the situation symmetrically. The asymmetry of the contact detections can be taken as an indicator of inaccuracy (see e.g. image 4). On the other hand, when people move freely, a contact is not a clearly defined occurence in time (compare image 3). Image 10 shows the percentage of symmetrical two-way detections from all detections relating to the pair with x axis depicting the allowed asynchrony between detections. 63% of the Android-Android pairs detect each other both ways simultaneously (granularity 5 minutes), the rest, approximately 37% are one-sided. The percentage of two-way detections increases, if temporal inaccuracies are accepted. iOS-iOS pairs are too few to make conclusions. iOS-Android pairs detect each other less reliably than Android-Android pairs. Confidence intervals have been calculated from the case amounts; due to clustering on user level the results are likely to contain higher uncertainty.*

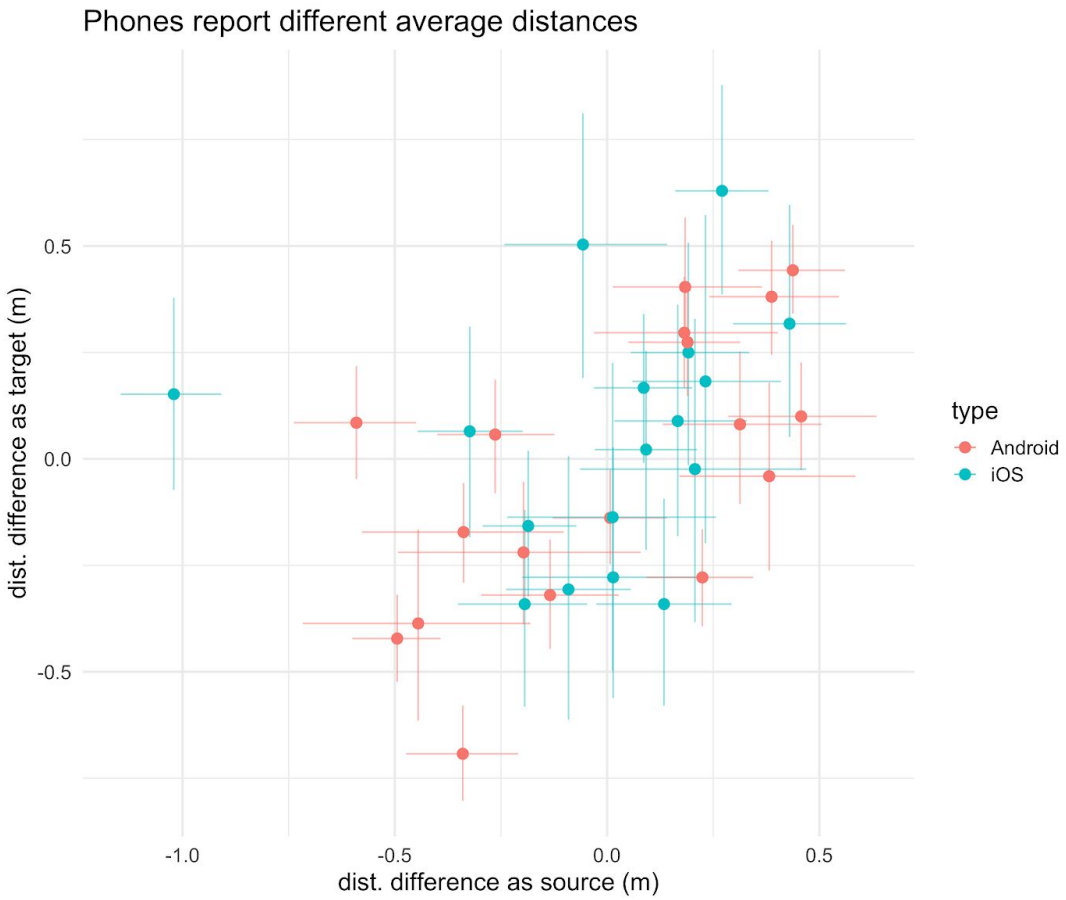


*Image 11. The reported distances between mobile devices were not on average the same but there were no noticeable differences between the different operating systems. Partly the differences were due to the technical properties of the devices as well as the ways in which users carry and hold their phones (pocket, purse, hand, table) but they can also be due to the way the users keep their physical distance in social situations. (Confidence intervals 80% have been calculated from a hierarchical regression model, ultimately from the case amounts and the similarity of the phones). Without any errors in the measurements the phone would be seen both as a source and target equally well and the points in the graph would be located on a rising diagonal.*

## Pilot user experiences

After the pilot had concluded, a final survey was made for the pilot users to get feedback. 28 out of 33 pilot users took part in the survey.

| | **How easy was the Ketju Pilot-app to use?** (average on scale 1-5, 5 = extremely easy) | **How did the Ketju Pilot-app increase battery usage?** (average on scale 1-5, 5 = not at all) | **Did the use of Ketju Pilot-app affect the normal use of your phone?** | **Did the use of Ketju Pilot -app affect the use of other accessory devices that connect via Bluetooth?** |
|---|---|---|---|---|
| **Android** | 4.6 | 3.5 | 4/28 Answered Yes | 1/28 Answered Yes |
| **Apple iOS** | 4.8 | 4.4 | 2/28 Answered Yes | 4/28 Answered Yes |

*Image 12. The final user survey at the end of the pilot included questions about how the tracking application affected normal mobile device use, battery usage and the use of other Bluetooth-using accessory devices.*

iOS users experienced the use of the application to be easier on average (although the difference is small enough to be included in the sample chance variation; this and other results below have not been tested for statistical significance). There were more problems on iOS with other Bluetooth accessory devices than with Android. Android users experienced more battery usage with the use of the tracking application and felt that using the application affected the normal use of their mobile device.

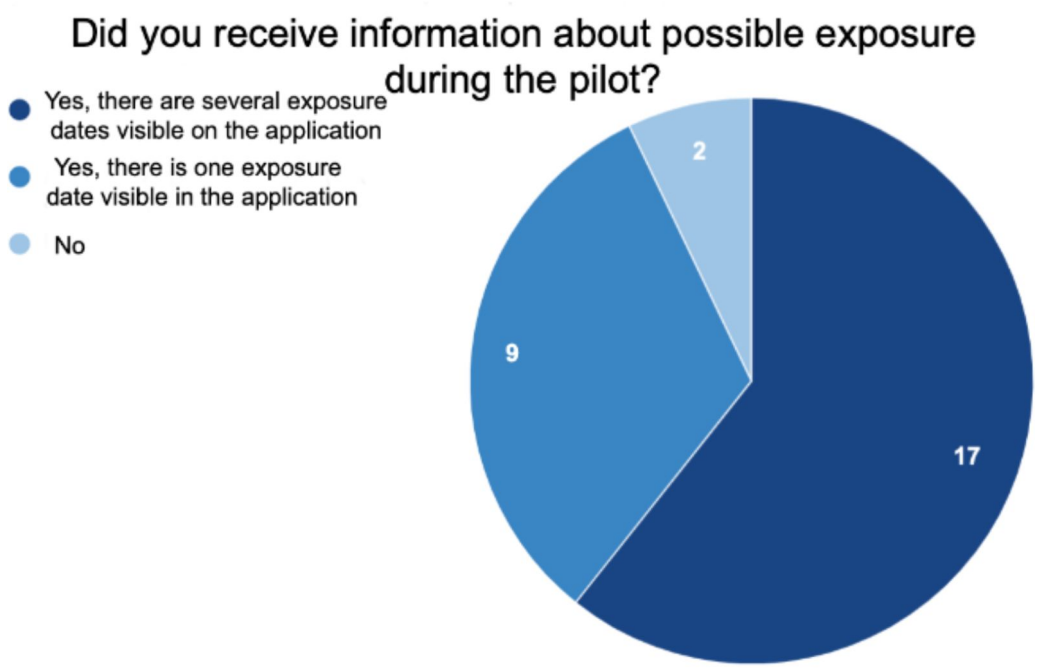


*Image 13. During the pilot, positive covid-19 diagnosis were simulated and inputted into the system. Most pilot users were "exposed" to these simulated cases during the pilot.*

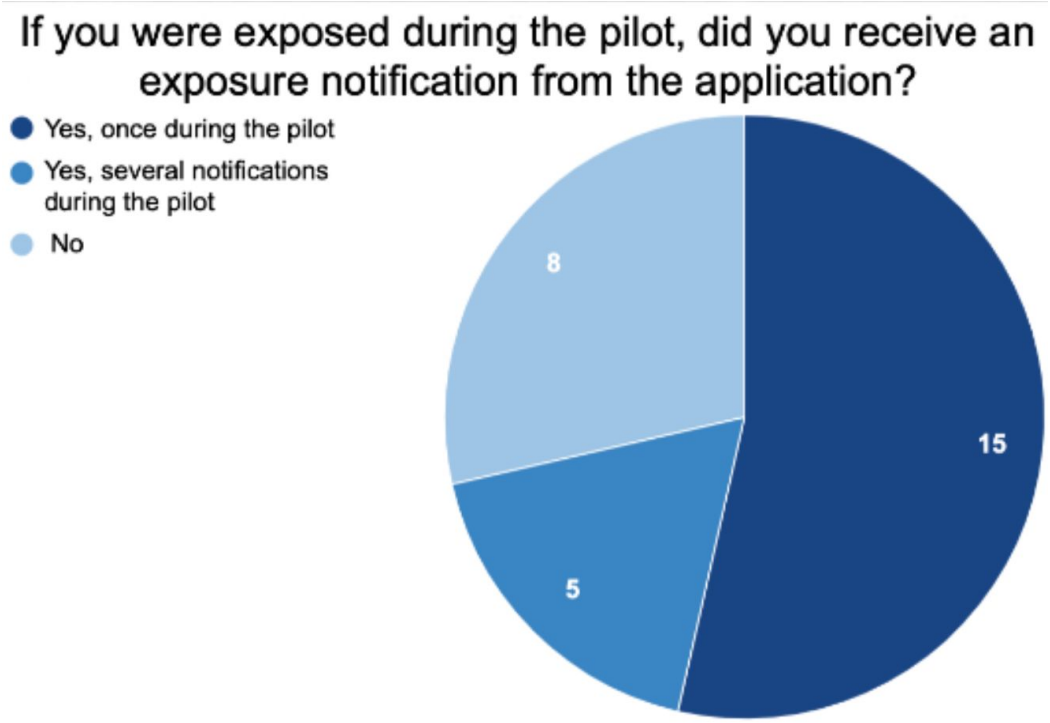


*Image 14. Most of the pilot users both received and noticed the application notification about possible exposure.*

Images 13 and 14 give an overview of how well the users thought the exposure notifications worked. Most users received at least one notification, when the simulated covid-19 diagnoses were shared. It seems that while many users noticed the first exposure notification, the later ones went either unnoticed or were not received. Eight users reported no exposure notifications.

Some open comments from the final survey concerning the notifications:

- Notification seemed a good method, but there was only one notification.
- Notification came only when I opened the application. The notification is not necessarily seen very easily. A more explicit way to notify would be better and so that it would always work. If I had not opened the application I would have not known about the exposure.
- It was not noticed easily. If it was cleared away with all the other notifications, you would not see it again.
- Notification did not come. I didn't open the Ketju daily and it might take a long time until I would have noticed the notification.
- I got one exposure notification, it worked
- Sending a notification seemed like a good method. I noticed it right away when I held my phone for the first time after the notification had been sent.

## Conclusions from the pilot

**The Apple and Google Exposure Notification API should be tried out in the next version of the tracking application.**

The Bluetooth implementation in DP-3T proved to be challenging in regards to accuracy and reliability. The issues with running in the background on iOS meant that these devices did not send out a signal regularly and some contacts with other iOS devices were left unregistered.

As of the time of writing the DP-3T consortium is also letting go of their own Bluetooth implementation and moving towards using the Apple/Google API.

**Reliable registration of contacts is challenging.**

The pilot data shows how significantly different devices interpret the distance between one another. The measurement of distance is based on signal strength, which is affected by both the device components and configuration as well as the environment, device orientation and its placement. When developing the tracking application further, time and effort should be spent on testing the reliability of contact registration and possible adjustments in parameters.

Reliability can be affected in principle by using device specific calibrations, optimizing the exposure parameters and creating a suitable compromise between sensitivity and accuracy (false positives vs. false negatives). If it is possible to adjust contact and exposure parameters,

calibration can be optimized to support either the prevalent epidemiological rule of thumb (e.g. 15 minutes at most 2 meter distance) or actual contagions. As the epidemiological rules of thumbs are in and of themselves heuristic in nature and contain some uncertainties, the best way would be to calibrate against real contagions. It is yet unclear whether the de-centralized model and especially the Exposure Notification API by Apple and Google allow for national calibrations or optimizations. On the other hand, both Apple and Google have good grounds as global operators to optimize these parameters.

The achieved reliability of contact registrations will heavily affect how the user is notified about possible exposures. It remains for the health care officials to estimate what is the sufficient level of reliability and the suitable angle for communication when the decision for a national launch is being made.

**Notification of exposure is a critical part of the application.**

The notification about exposure to the user needs to be unmistakable. In the pilot application, this was hindered by having to do all the data-processing and notifications on the user's device locally. Because of issues with running in the background, the notification was sometimes only seen by the users when they opened up the application and made it active again. Many of the users wished that the exposure notification could be sent a text message or by email, but this was not possible due to the de-centralized and anonymous model and the issues with running in the background. As the application has no information about the users identity, the notification must be sent through the application.

When using application notifications, it must be noted that each user can set their notification preferences on their devices as they wish. This might result in the notifications being off or being made known to the user in a manner where they will go unnoticed by the user. This could be affected by giving clear instructions to the users about how to set appropriate notification preferences for the tracking application.

**The de-centralized tracking model does not offer significant help for manual tracking. Within the de-centralized model, an exposure notification can only lead to the user coming voluntarily forward to the health care officials, getting tested or setting themselves voluntarily into quarantine-like conditions. Based on the exposure notification, the user who was infected cannot be known.**

**To track the chain of infection unambiguously would require a hybrid tracking model.**

The pilot application used the DP-3T protocol in debug mode, which made it possible to track the contacts to specific users during the pilot phase. This was necessary to compare the Bluetooth contact results to the manual notes the pilot users did during the pilot.

However, when using the DP-3T in production mode or using the Exposure Notification API by Apple and Google, the contacts cannot be traced to specific users. The user only gets a notification of a **possible** exposure with approximately **a day's accuracy window of exposure**. If these are the only factors known about the exposure, it is extremely difficult for the health care official doing the manual tracking to connect this exposure information to other facts known. Further, without more detailed information on the exposure, a doctor cannot order an official quarantine which would result in the user's getting i.e. financial aid. The only actions left for the user after an exposure notification are to get voluntarily tested and/or staying in voluntary quarantine.

The strong anonymity causes some problems as the application will inevitably result in some false positives of exposure from situations that have not contained exposure risks. The strong data privacy required by the de-centralized model (no specific time or exposure source) prevents health care officials from establishing an unambiguous exposure identification and evaluation.

If the use of hybrid models would be possible in the tracking application so that more detailed contact data could be revealed to the health care officials by users' consent, it would be possible to connect exposures to the known diagnosed cases and remembered contacts within the current manual tracking process. By interviewing both the infected and exposed user, it would be possible to e.g. ascertain that the users have been close to each other in a bus but were facing in different directions. Based on this more detailed situational information, the health care officials could make a more informed decision on the need for quarantine. However, a hybrid model is not possible if using DP-3T or the Exposure Notification API by Apple and Google.

Globally, centralized and mandatory applications like Singapore's Bluetrace have been in use for a long time, and thus results of their efficiency in helping to control epidemics are available. At the time of writing this report, de-centralized applications have been used for such a short time period that data on their efficiency and benefits is lacking. Overall, the role of tracking applications in controlling of global epidemics is complementary rather than comprehensive in aiding the manual tracking processes and other tools and measures.

# 8. Publication of results

The results of the project will be published for free use after the project as per the terms of the funding agreement by Sitra. Application codes are published as open source on GitHub.

Additionally, other materials produced during the project will be available on GitHub including this report, architectural schema and images of the user interfaces.

GitHub: https://github.com/ketjusovellus