[Automated English Translation]

Technical and Legal Review of the Stopp Corona App by the Austrian Red Cross

Ulrich Bayer[3], Andreas Bernauer[3], Marco Blocher[2], Benedikt Gollatz[1], Aljosha Judmayer[3], Michael Koppmann[3], Christian Kudera[3], Thomas Lohninger[1], Georg Merzdovnik[3], Armin Ronacher, Max Schrems[2]

epicenter.works - Plattform Grundrechtspolitik[1]

Widerhofergasse 8/2/4
1090 Wien
Österreich

NOYB – European Center for Digital Rights[2]

Goldschlagstr. 172/4/2
1140 Wien
Österreich

SBA Research gGmbH[3]

Floragasse 7/5. Stock
1040 Wien
Österreich

Content

1. Introduction

1.1. Scope and disclaimer

With the *Stopp Corona App*[1] the **Austrian Red Cross** („ÖRK“)[2] provides a contact tracing app for the containment of new infections with SARS-CoV-2 ("coronavirus") in Austria. The source code of version 1.1, which was not publicly available at that time, has been made available to epicenter.works, noyb, SBA Research and Armin Ronacher (independently) for analysis under a Non-Disclosure Agreement (NDA).[3] The consequence of the NDA is merely that this report had to be sent to Accenture GmbH and the ÖRK 48 hours before publication. Our three organizations and Armin Ronacher received no financial compensation or other benefits for this review. We see it as our task to educate the public as neutrally as possible about this app, focusing on the technical and legal level with a focus on data protection and IT security issues.

The following documents/files were available to the organisations and persons involved in this project:

- The source code of the Android and iOS apps and the server application from 07.04.2020[4]
- Privacy policy *Stopp Corona App*[5]
- data protection impact assessment report
- “General” declaration of consent
- declaration of consent regarding the “Symptom-Checker”
- General terms of use, as embedded in the source code
- FAQs[6]

The evaluation of the epidemiological or medical usefulness of the *Stopp Corona App* is not a subject of this analysis. This analysis also does it examine compliance with consumer protection law, e-commerce law or other areas of law that affect the relationship between the ÖRK and the user of the app.

[1] https://participate.roteskreuz.at/stopp-corona/ (accessed on 18.04.2020).

[2] The Austrian Red Cross is the national Red Cross Society (according to the Geneva Conventions) in Austria and is constituted as an association according to the Austrian Association Act 2000 (ZVR 432857691).

[3] Full text of the NDA: https://epicenter.works/document/2465

[4] SHA256(StoppCorona1.1-QA_215-Android.zip)=
568f992d856ac3bd1b26ed5f09f8edce1316af5127a82775953afe43a93beb4a
SHA256(StoppCorona1.1-QA_580-iOS.zip)=
938b7833ba7a275277c02f81f412c0a4a9ccd7c9ce51a6fbf8bae4211ba9ce6c
SHA256(2020-04-09-CovidAppSources.zip)=
0e6c763178819b0c62cddc344497209ac5c94d3457c47315f0a889bd0a46c0da

[5] https://www.roteskreuz.at/site/faq-app-stopp-corona/datenschutzinformation-zur-stopp-corona-app/ (accessed on 18.04.2020).

[6] https://www.roteskreuz.at/site/faq-app-stopp-corona/ (accessed on 18.04.2020).

1.2. Executive Summary

The debate about contact tracing apps is only a couple of weeks old and is therefore still in flux. The Austrian Red Cross started developing an app comparatively early, while many other countries are just starting to discuss concepts. After reviewing the source code, we have the impression that many of the requirements for the app were only added after the start of development (e.g. automatic handshake). Although a privacy friendly approach has always been followed, the additional requirements and technical limitations on the smartphone operating systems of Google and Apple led to an architecture that has certain problems.

Our code review identified some serious privacy issues, some of which have already been taken care of by applying a hotfix. From a legal perspective, we have some suggestions for improvement, but in our opinion the concept of the app is compliant with data protection laws. The technical security check did not reveal any critical security vulnerabilities, but some suggestions for improvement were made.

Some mentioned privacy issues of the app can hardly be solved in the current architecture. In the international debate, very promising approaches have now emerged. Privacy-friendly protocols such as DP-3T are supported by the EU Commission[7] and international scientists.[8] They also support automated handshakes on Apple Smartphone devices - which is currently not functional.

This report pursues two goals: (1) to explain how the “Stopp Corona” app works, and (2) to suggest concrete solutions for its improvement. In this context we would like to thank the Red Cross and Accenture for a professional and solution-oriented exchange. Although the report raised some critical points, these problems were acknowledged and the concrete proposals for solutions were quickly addressed.

1.3. Overview of the recommendations

Of the 25 recommendations in this report, 16 were implemented by a hotfix, which is already available for download when the report is published. For 16 recommendations, an improvement was announced for the next release at the end of the 18th calendar week. The remaining 4 recommendations will only be solved with the migration to the new architecture - probably within 4 weeks.

[7] https://ec.europa.eu/info/sites/info/files/5_en_act_part1_v3.pdf (page 10, 21.04.2020)
[8] https://drive.google.com/file/d/1OQg2dxPu-x-RZzETIpV3lFa259Nrpk1J/view

Technical Recommendations

Chapter	Recommendation
2.2.2.	The Co-Epi Project defines important security and privacy recommendations for contact-tracing-apps. We recommend that these recommendations are followed in the chosen architecture.
	Feedback Red Cross
	The desirable security features of contact tracing apps are a cornerstone in the design of such apps. We find these features important and try to implement them in the best possible way.
2.2.3.	**Recommendation**
	At the beginning of the development process for the Red Cross application, no other approaches or reference implementations have been available. Therefore, the ÖRK and Accenture had to design their own architecture. In the meanwhile, there are multiple approaches which are debated in academia an expert communities (e.g., DP-3T, Co-Epi/CovidWatch). Medium- to long term, we recommend to switch to one of the decentralized architectures, which is recommended by international experts from various disciplines.
	Feedback Red Cross
	Solution with implementation of DP-3T We continue to clearly prefer a decentralized architecture approach. Therefore, we are in close contact with DP-3T1) and Google&Apple2) to ensure that their approaches meet our requirements, and have received very positive feedback. As soon as the approach reaches a practical maturity and availability, we will switch the architecture to DP-3T and pay attention to interoperability with other countries. (1) DP-3T: regular coordination meetings with Prof. Capkun (ETH Zurich) as well as Prof. Bugnion (EPFL) to include our requirements for different types of warnings, manual handshakes, tokens etc. in the DP-3T implementation. (2) Google and Apple in Austria, the global partnership of the implementation partner Accenture with Google&Apple regarding early adopted accesses of the announced solution; Apple's technical 3rd level support for mobile apps and solution of the iOS background limitations as well as clarification of a partial use

	of the Google&Apple approach for tracing devices without using the messaging functionality.
2.3.1.	**Recommendation**
	The statistic messages to the server have to be redesigned in a data protection friendly way or removed completely, otherwise it is theoretically possible to draw conclusions about contacts or chains of infection on the side of the server from the Red Cross.
	Feedback Red Cross
	Solution implemented in Release 22.4.2020 The recommendation was taken into account and the statistical report was removed. A new approach for statistic messages will only be carried out with the specifications described in chapter 2.3.1.
2.3.2.	**Recommendation**
	The tracking of individual smartphones through third parties, by means of locally recording the public keys in handshakes, as well as the associated possibility of creating movement profiles of app users, must be prevented technically, for example by changing key pairs.
	Feedback Red Cross
	Solution planned for release 30.4.2020 The suggestion was taken up and is currently being implemented. Note: The exchange of public keys is currently secured by various mechanisms and public keys are not directly provided e.g. via Bluetooth broadcast.
2.3.3.	**Recommendation**
	We recommend to reconsider the use of p2pkit for the automatic handshake and the use of Google Nearby messages for the manual handshake.
	Feedback Red Cross
	Solution with implementation of DP-3T As described in 2.2.3, the implementation of DP-3T and the Google&Apple mechanism for the direct exchange of handshake information between two devices replaces the mechanisms previously used for this purpose.

2.3.4.	**Recommendation**
	We recommend that no communication with the Red Cross server or a third party services takes place before the users have agreed to the processing of their data.
	Feedback Red Cross
	Solution implemented in Release 22.4.2020 The recommendation was taken up and already implemented in the current release.
2.3.5.	**Recommendation**
	The probability that infection messages can be linked to known public keys of users must be made as low as possible.
	Feedback Red Cross
	Solution planned for release 30.4.2020 and DP-3T conversion The recommendation is followed. It takes criminal energy combined with technical expertise to be able to analyse the information despite existing security measures. To further minimize the attack vector for theoretically existing statistical inferences, rotating keys will be used until the release on 30.4.2020.
2.4.4.1.	**Recommendation**
	When using cryptographic algorithms, we recommend following best practice recommendations regarding minimum key lengths and padding procedures.
	Feedback Red Cross
	Solution with implementation of DP-3T The recommendation is gladly taken up and an implementation with more suitable padding schemes, also available for iOS, is currently being analysed. With the changeover to DP-3T the topic is addressed in any case.
2.4.5.1.	**Recommendation**
	With regard to eavesdropping security, it is a sensible measure to limit trust in connection with certificate issuance to only one CA ("certificate pinning") in

	order to further increase security. We recommend incorporating this measure in the next version.
	Feedback Red Cross
	Solution implemented in Release 22.4.2020 The recommendation was taken up and already implemented in the current release.

Legal Analysis

In a brief legal analysis under the DSGVO (focusing on Articles 5, 6 and 13 DSGVO), the following problems and open issues have been identified.

Chapter	Recommendation
3.1.2	A clear identification of all sub(sub)processors in the data protection information must be made.
	Feedback Red Cross
	Will be included.
3.1.2	**Recommendation**
	Harmonisation of the purposes of data processing between the GPI and the Data Protection Directive.
	Feedback Red Cross
	Will be included.
3.1.2	**Recommendation**
	A clear separation of the two Google services (Nearby and Firebase) and the respective responsible parties in the data protection information must be ensured.
	Feedback Red Cross
	Will be included.
3.1.2	**Recommendation**
	Clarification is required in the privacy policy regarding the use of the Apple Push Notification Service.

	Feedback Red Cross
	Will be included.
3.1.2	**Recommendation**
	The use of alternative processors that are not subject to US law is recommended.
	Feedback Red Cross
	Recommendation is implemented for telephone numbers (TAN) in the current release. A solution is being worked on for pseudonymized data.
3.1.4	**Recommendation**
	It would be desirable to clearly identify those types of data that can be technically obtained from health and district administrative authorities, as well as the cases in which such information can be obtained under Austrian law.
	Feedback Red Cross
	So far there has been no request for information or efforts to do so. The issue was clarified in the data protection information.
3.2.1	**Recommendation**
	It is necessary to clarify the purpose for which IP addresses are processed and the legal basis of the data processing that takes place before consent is given.
	Feedback Red Cross
	IP addresses are not stored. The calls before consent were corrected.
3.2.1	**Recommendation**
	The storage period of IP addresses is unclear. The storage period of the "digital handshakes" must also be indicated in the data protection information.
	Feedback Red Cross
	IP addresses are not stored, information is added if not already clear.
3.2.3	**Recommendation**
	The need for a second, separate identical consent for the symptom checker should be reconsidered.
	Feedback Red Cross

	This was considered necessary after careful consultation with regard to Article 7(2) DPA.
3.2.3	**Recommendation**
	The deletion period (54 hours) should appear uniformly in all documents.
	Feedback Red Cross
	The deletion period is flexible in terms of Containment 2.0 according to the state of the art. The data protection information has been adapted accordingly.
3.2.4	**Recommendation**
	There are massive doubts whether the statistics function meets the requirement of data minimization.
	Feedback Red Cross
	The recommendation will be already implemented in the current release.
3.4.1	**Recommendation**
	The modalities and effects of a revocation of the consent given should be presented in a comprehensible manner. It should be possible to delete individual handshakes from the device.
	Feedback Red Cross
	Suggestions are gladly taken up and placed for prioritization in the backlog of possible updates.
3.4.2.	**Recommendation**
	The app must actually delete the handshakes immediately after the deletion period.
	Feedback Red Cross
	The suggestion is being implemented.
3.4.2.	**Recommendation**
	The storage period of IP addresses must be specified.
	Feedback Red Cross
	IP addresses are not stored, information will be added if not already clarified.
3.5.5	**Recommendation**

	Further "appropriate measures" (within the meaning of Article 5(1)(c) GDPR) seem to exist in order to avoid false information as far as possible.
	Feedback Red Cross
	From a technical point of view (see epidemiological explanations), false-positive reports are generally accepted or anticipated. This is identical to the analogous world, where many false-positive cases are also tested to get a negative test result.
3.5.6	**Recommendation**
	It is recommended that the transmission times are specified in the privacy policy or in the FAQs.
	Feedback Red Cross
	Currently, the message is transmitted with a maximum delay of one hour. Improvements can be incorporated with pleasure. Note on classification: The current official information flow "Symptom > testing > result > research of social contacts > notification" typically takes days - compared to 1 hour in the app.
3.6	**Recommendation**
	At least the own UUID has to be made visible to the user or an alternative possibility of clear identification has to be created to enable the exercise of data subject rights.
	Feedback Red Cross
	This wish for improvement is understood and prioritized. In fact, the UUID will no longer be used after the statistics message has been extended.

1.4. General functioning of the app

The Red Cross' "Stopp Corona" App is used to record contacts between mobile phones on which the application is installed to retroactively warn the recorded contacts in case of a later detected infection. For this purpose a random ID and a key pair consisting of a public and a private key is generated on each device.

The Stopp Corona App follows a model in which participating mobile phones exchange data with each other via a central infrastructure and then store it locally in the mobile phone. To detect contacts between participating devices, each of these devices tries to perform so-called "handshakes" with other devices in its environment. The user can choose whether

the handshakes should be performed automatically or manually. The goal of a handshake is to transfer the public key to the other device. However, due to limitations of the amount of data that can be transmitted using Bluetooth Low Energy (BLE), a central cloud service is also used for this purpose in addition to BLE (p2pkit for automatic handshakes and Google Nearby for manual handshakes).

In a simplified way, the existing architecture can be divided into two parts: on the one hand, the infrastructure required to exchange handshakes (or public keys) in the event of a contact, and on the other hand the infrastructure required to send infection messages to the affected contacts.

If the application registers a contact, i.e. another mobile phone with the application installed in the immediate vicinity, the respective public key is to be exchanged by means of the handshake. Two different methods are used to perform handshakes. Firstly, a manual handshake can be performed, in which the public (static) key is exchanged through the Google Nearby Cloud infrastructure using the Nearby Messages API. For this purpose, the public key is stored on Google servers and a token is exchanged based on ultrasound or Bluetooth, with which this public key can be looked up and downloaded there. Secondly, automatic handshakes can be performed. In this case the public key is exchanged by the p2pkit infrastructure of the service provider Uepaa. Functionally, this is analogous to the Google variant but without ultrasound.

In addition to the transmission of the public keys, both the iOS and the Android applications currently send a message to the Red Cross server independently of each other that such a handshake has taken place. This message contains the initially generated (static) ID of the respective client, as well as a timestamp rounded to the current hour.

If one of the parties involved in the previous handshake detects that he or she has been infected (Red Warning), or would like to send a warning about a suspected infection (Yellow Warning), a TAN code can be requested via the application. After entering the TAN code, the application can be prompted to send an encrypted message to all recent contacts. The suspicion (Yellow Warning) is determined via a self-test. If the user states that he/she suffers from recurring dry cough and/or a body temperature above 38°C, he/she will be prompted to send a suspected illness message to COVID-19. If this suspicion is later proven to be false by a test, the user can send an revoke the suspicion (Green Signal)l to his contacts.

In order to deliver the infection message, suspicion message or revoke message, it is sent encrypted to the server of the Red Cross. The server cannot read the message itself and does not know the public keys involved. The server merely acts as a contact point for all applications to store new encrypted messages or to periodically retrieve new encrypted messages. To reduce the number of messages to be downloaded, the total number of encrypted messages is reduced to 1/256 of all encrypted messages based on a prefix (first byte of the SHA256 hash of the public key) of the own public key. This prefix must also be transferred to the server together with the encrypted message during notification.

Now, to find out if an infection message exists for a participating device, each device tries to decrypt the downloaded data with its own private key and ignores the messages it cannot decrypt because they were encrypted for another device.

If a decryption attempt is successful, the application notifies the user that they are potentially affected. The message itself contains only the information that it is a suspicion (Yellow Warning), a potential infection (Red Warning), together with a time of the respective encounter rounded to the hour that is the reason for the notification. The application also sends a notification to the Red Cross server that a warning has been received on this device. This includes the initially generated (static) ID of the device and a timestamp rounded to the hour.

2. Technical Analysis

2.1. Introduction and methodological approach

The development of secure software is a basic requirement for an application that processes personal data and must withstand malicious attacks. For this reason, a large number of best practice recommendations and concepts have been established which promote secure development. A standard often referred to in the technical literature is the Security Development Lifecycle (SDL)[9] and the associated Process Guidance[10] (latest version 5.2 from 2012):

- Secure by Design: Ensures that safety-relevant aspects are taken into account as early as the planning phase of a software. This requires a secure architecture, which is derived from the collection of attack scenarios and a threat analysis.
- Secure by Default: Takes into account that despite careful planning, software can contain weak points. Therefore, the components of a software must always be operated with the lowest possible authorizations. Furthermore, a security concept must not be based on one measure alone, but must have multi-layered and far-reaching measures.
- Privacy by Design: The architecture and design of an application must provide for the minimum processing of data and data protection risks must be excluded in advance by the design. The user's consent must be obtained before processing. The collection itself must be transparent and clearly visible to the user. The collected data must be protected against access by third parties to the best possible extent, both during transmission and storage.
- Privacy by Default: Demands data protection through data protection-friendly default settings. This means that an application should be operated with the most data protection-friendly settings when it is commissioned. This concept is designed to protect users who are less technically inclined.

The classic method to ensure compliance with the presented concepts after development is an independent software security audit. The source code is searched for design flaws, security vulnerabilities and disregard of best practice recommendations. A software security audit is a complex procedure that would require at least 20 project days for the components provided by Accenture (Android source code, iOS source code, Push Service Backend, SMS Notification Backend and RCA CoronaApp Backend). Only a fraction of the required time was available for the analysis carried out. Furthermore, a separate test infrastructure would be necessary to be able to run malicious test cases without affecting production operations. For these reasons, the following results are only an initial assessment, which was created within the scope of a quick check. It is conceivable that if the source code is disclosed (open source release), further problems will be identified which are not addressed in the following analysis.

[9] https://www.microsoft.com/en-us/securityengineering/sdl
[10] https://www.microsoft.com/en-us/download/details.aspx?id=29884

2.2. Architecture

The basic architecture of the system has already been explained in the section "Allgemeine Funktionsweise". This section deals with more profound consequences of the architecture.

2.2.1. Handshakes

Devices exchange Public Keys with each other (with an indirection through the p2pkit cloud) during the automatic handshake. The idea or basis of the architecture can therefore be seen as decentralized if this exchange could happen directly from device to device.

It is important that the public key does not change in the course of time and can therefore be seen as a unique identification feature. This is undesirable from a data protection perspective. A passive attacker can use this feature to unambiguously recognize persons if they stay at a location several times. The application also eliminates security features of the Bluetooth stack in mobile phones: normally the Bluetooth stack automatically rotates the MAC address (the identification feature of a Bluetooth transceiver) to prevent such recognition.

The fact that messages are encrypted for a specific recipient also causes a certain scaling problem. The downloaded part of the database represents about 1/256 of the whole database and is determined by the first byte of the SHA256 sum of the own public key, which is exchanged during handshakes. This means that if the application were to be adapted to rotate public keys exchanged during a handshake, increasingly larger parts of the database would have to be downloaded. There is thus a risk that the system will be overloaded and have scalability problems at the peak of the pandemic.

In general, it should be noted that the system architecture is not inherently insecure. However, it is not desirable from the perspective of scalability and dependence on central servers. Due to the limitations of BLE, it is questionable whether the existing architecture can be implemented compatibly without having to resort to a cloud service. In order to exchange handshakes between devices, a connection would have to be established between the devices, which cannot be implemented under iOS based on BLE alone. However, using third-party cloud services to perform handshakes breaks with privacy-by-design principles (see section "Privacy").

2.2.2. Security Characteristics

The Co-Epi-Project[11] has defined a list of desirable security features of contact tracing apps, which are presented below:

- **Server Privacy**: Server Privacy means that the central server infrastructure cannot draw any conclusions about the users (e.g. their contacts or position data). This feature is desirable because it reduces the consequences of an attack on the central

[11] https://github.com/TCNCoalition/TCN (accessed 19.04.2020).

infrastructure. The less information the server knows about the user, the less can be stolen when compromised by an attacker. Naturally, the server privacy is reduced for users reporting infections, as more information must be disclosed.

- **Source Integrity:** Source integrity refers to the inability to send infection messages to users with whom the user has not been in contact. Furthermore, there must be no possibility for a user to fake the identity of another user.
- **Broadcast Integrity:** Broadcast Integrity refers to the impossibility for a user to send information about an identity other than himself during the handshake. Furthermore, a user must not assume a false identity during the handshake.
- **No Passive Tracking**: No Passive Tracking refers to the fact that passive listening to Bluetooth transmissions does not reveal location information about users until they have generated infection reports.
- **Receiver Privacy:** Receiver Privacy means that the recipient of an infection message does not disclose information about himself/herself to others.
- **Reporter Privacy:** Reporter Privacy refers to the fact that only the minimum of information is shared by infected persons. Specifically, no data should be shared with persons with whom one has not been in contact. Persons with whom one has had contact should only receive the approximate time of the encounter.

In the following, we consider the security features presented for the architecture of the *Stopp Corona App*.

2.2.2.1. Server Privacy

In the case of the ÖRK App, server privacy is maintained for the exchange of infection messages *from the perspective of the protocol*. However, due to the use of the infrastructure of p2pkit and Google Nearby, as well as the implemented statistical functions in the app, the *overall Server Privacy is violated*. The former is problematic, because due to the technical limitations of BLE, the public key must be transferred to either the infrastructure of p2pkit or Google Nearby for each handshake. Since neither service has end-to-end encryption (E2EE), the public keys are stored on the servers in plain text. This makes p2pkit and Google Nearby Cloud an interesting target for attackers. With the combination of access logs and the public keys, the complete social graph ("who had contact with whom and when") can be reconstructed. More details on the data protection-relevant use of these data services can also be found in section "2.3.3. Use of third-party services".

In addition, the system currently has statistics messages that also violate server privacy in the current implementation (see section "2.3.1. Statistics messages")

2.2.2.2. Source Integrity

Since a public key, which determines the recipient of a message, can also have been transmitted by third parties (see Broadcast Integrity) and infection messages are not signed, it is also possible that infection messages are transmitted to persons with whom an infected person has not been in contact. Source integrity is therefore not guaranteed.

2.2.2.3. Broadcast Integrity

Since only the public key of one person is exchanged in handshakes, a malicious user can exchange his key for the key of another user and thus provide a false identity. Broadcast integrity is not given with this. In concrete terms, this means that a user can collect public keys from third parties and thus fake contacts by distributing the public keys of these third parties instead of his own.

2.2.2.4. No Passive Tracking

Although rotating identifiers are exchanged on the BLE layer according to the BLE specification itself, the currently non-rotating public key of the user is stored in the p2pkit cloud during the automatic handshake. These can be "exchanged" for static keys in the p2pkit cloud. Theoretically, this behavior could be detected in the p2pkit cloud and restricted by rate limiting, but it is generally assumed that the system can be exploited for passive tracking.

For more on the problems caused by passive tracking, see section "2.3.2. Movement profiles by tracking in physical space".

2.2.2.5. Receiver Privacy

This is maintained in the architecture with a sufficiently large number of users, but is broken with in the concrete implementation (see section " 2.3.5. Traceability of notifications within a short period of time to known public keys"). In case of database accesses by the users, the user discloses the first byte of the SHA256 sum of his public key. The user can thus be classified into one of 256 classes of users. As long as the number of users in the system is large enough, users should not be identifiable in this way. With a small number of users, however, this information disclosure would be problematic.

2.2.2.6. Reporter Privacy

Architecturally, the same considerations apply here as for Receiver Privacy. This concrete implementation, however, breaks with Reporter Privacy (see section " 2.3.5. Traceability of notifications within a short period of time to known public keys"). Incidentally, an infected user must in principle also authenticate himself with a TAN. TAN and telephone number are transmitted to the ÖRK when the infection is reported.

➔ **The security features for contact tracing apps presented by the Co-Epi-Project are important recommendations from the perspective of data protection. We recommend that the chosen architecture takes these recommended security features into account.**

2.2.3. (Distributed) Contact Tracing Architectures

In the last weeks a variety of centralized and decentralized architectures for contact tracing were discussed, which are characterized by different conceptual and technical features. There is currently a lively debate in the academic environment about which architectures and

conceptual approaches are best suited for contact tracing applications. In general, the topic is a very young science where experience and evaluation are still lacking. For this reason there are currently no best practice recommendations for the development of contact tracing architectures.

However, it is slowly becoming clear that compatibility with the BLE stack is an important requirement. To fulfill the compatibility, a message must not be longer than 20 bytes, since this is not provided for in the BLE stack. Both the DP-3T protocol and the Google/Apple Privacy Preserving Contact Tracing protocol support this requirement and basically only exchange random identifiers that rotate together with the MAC address of the Bluetooth stack. This means that there are no further attack vectors compared to what a mobile phone already transmits on its own. Only when an infection is reported secret keys are revealed via a central infrastructure, from which the identifiers can be recalculated and downloaded by other devices.

This system is different from the *Stopp Corona App*. The *Stopp Corona App* also downloads infection messages from a central infrastructure; however, these are specially encrypted for the recipient. With DP-3T or the Apple/Google system, the device would instead download data with which all - pseudonymised - possible infections can be calculated. These approaches were not yet available when the ÖRK app was first developed, but today they represent a viable alternative.

In the following, the *Stopp Corona App* architecture is compared in detail with other architectures.

2.2.3.1. BLE Layer Data Exchange

At the Bluetooth level, the *Stopp Corona App* exchanges an identifier that is used to exchange data through Google Nearby or p2pkit via a cloud service. This data transfer currently consists of the exchange of a public key and, in the case of a manual handshake, a four-digit random number that is displayed to the user. In comparison, decentralized systems such as DP-3T, Co-Epi/CovidWatch or Apple/Google exchange a random token without using a cloud service.

2.2.3.2. Infection Exchange

Infection messages are exchanged via a central backend both in the *Stopp Corona App* and in decentralised systems. In this respect, the term "decentralised system" is only applicable to a limited extent. With the *Stopp Corona App*, devices store infection messages which are encrypted for a specific other end device. With DP-3T and others, secrets or seeds of infected persons are usually exchanged, from which the random tokens that may have been exchanged in case of contact can be calculated. In both cases, contacts are only recognised on the user's device and cannot be recognised by the central system.

2.2.3.3. Source Integrity

The *Stopp Corona App* approach would theoretically allow infection messages to be sent only from one person to another person when they have been in contact with each other. Provided that it is ensured that the public keys of both devices are exchanged in case of contact, infection messages could be signed and this signature checked by the recipients. In the current version of the app, infection messages are not signed. A problem that would arise in the current architecture through the use of signatures is that users could determine the identity of the sender when receiving an infection message, which is contrary to the approach of Reporter Privacy.

In a decentralized architecture only temporary tokens are exchanged. In such a system, a user is therefore able to distribute the tokens of other users over a certain period of time. The Apple/Google protocol restricts this by making tokens valid for only about 30 minutes. If it is possible to deduce from the infection messages for which period of time exchanged tokens were valid in each case, recipients can ignore tokens that were not received in the appropriate 30-minute window.

2.2.3.4. Infection Messages

The *Stopp Corona App* currently requires that when an infection is reported, the infected person must give the ÖRK their telephone number. This aims to prevent the misuse of an infection report and, if necessary, makes it possible to contact the affected person. However, depending on what interactions with the person are necessary, this can be avoided. It would be possible to separate the infection message itself from the authentication for one and thus not allow a conclusion from telephone number to concrete infection messages at the backend.

DP-3T and other protocols assume that an infection message can only be triggered by a positive Covid-19 test. As a suggested protocol, inactive authentication codes are transmitted here, which are only activated with a positive test result. Since the authentication codes are entered into the system by doctors, for example, this can create a fuzziness that does not allow any conclusion to be drawn about the specific person. The current structure of messages based on symptoms (yellow) and test results (red/green) offers an added value and should be retained if possible.

➔ **At the start of the development of the ÖRK App, no architectural approaches were available, forcing the ÖRK and Accenture to implement their own architecture. In the meantime, different approaches (e.g. Co-Epi/CovidWatch, DP-3T, NOVID20, Pepp-Pt) are discussed in the professional community. We recommend the change to an architecture that is recommended by international experts from different scientific disciplines.**

2.3. Data protection

The development of an app that processes sensitive data such as contacts and the state of health of the users is to be measured by whether privacy-by-design principles have been observed, which should ensure that data protection aspects are taken into account during development. The central principle is "preventive not remedial", i.e. that data protection risks are excluded in advance and are not to be mitigated by further measures afterwards. In the present context, the risk that contacts that have taken place and possible chains of infection can be reconstructed by the ÖRK or third parties, or that significant changes in the assessment of the probability of such occurrences can be made with respect to previously existing knowledge, must be excluded.

Basically, it can be said that this approach has been followed to a certain extent in the development of the *Stopp Corona App*. First of all, the system is designed in a way that it is possible that the backend of the ÖRK does not know who has been in contact with whom and what chains of infection might have been created, while still being able to forward appropriate infection messages in case of infection. Unfortunately, this guarantee is violated in the current version of the App, as communication metadata, such as IP addresses and times of transmission, which allow conclusions to be drawn about persons, the nature and content of the communication, have not been taken into account, particularly in the operation of the App. This is a breach of privacy-by-design principles.

2.3.1. Statistical messages

The backend provides an API endpoint */Rest/v3/track-events*, which allows installations of the app to promptly report a handshake or the receipt of a successfully decrypted infection or all-clear message. The unique device number and the exact time of the handshake, as well as the received and successfully decrypted infection and all-clear messages, are transmitted to the server. In combination with the resulting metadata of the communication (IP addresses of the reporting devices, times), conclusions can be drawn about the persons between whom handshakes have taken place and which infection chains are possible.

We strongly recommend, as far as the use of such statistical reports is necessary at all, that these reports are transferred

- not in time with the event that has taken place, but from each device approximately daily at a certain time,
- without the transmission of unique identifiers,
- not in the form of timestamps, but in the form of summary statistics (i.e. number of messages issued), and
- not in the form of real values, but noisy data using a local differential privacy mechanism.

➔ **The statistics messages to the server must be redesigned or removed in a data protection-friendly manner, otherwise it is possible to draw conclusions about contacts or chains of infection.**

2.3.2. Movement profiles by tracking in physical space

As explained in the chapter on "No Passive Tracking", it is possible to track an installation of the app in physical space and thus create movement profiles. To do this, the app must be in automatic handshake mode, which is the default setting on Android devices. On iOS devices this tracking does not work at the moment due to limitations of the operating system. Due to the app's automatic exchange of the public key and the static nature of the public key, a single device can be recognized for an unlimited period of time. Using sensors in public places or in means of transport, tracking and the creation of movement profiles of individual persons would be possible.

The Bluetooth stack has been preventing this by default in smartphones since around 2014 by randomizing the MAC address; however, the current protocol overrides this security feature by always sharing the same key. In addition, as already mentioned in the section on "Server Privacy", all public keys are now stored on the p2pkit cloud.

To better visualize the problem, imagine the case where a supermarket uses a passive Bluetooth sniffer to recognize people who shop frequently.

➔ **The tracking of individual devices in physical space and thus the creation of motion profiles of users of the app has to be technically impossible.**

2.3.3. Use of third-party services

2.3.3.1. p2pkit

Automated handshakes are handled via the Internet using the third-party service p2pkit. This creates metadata of the communication between the devices as well as statistical messages of the involved devices within p2pkit. Information about the time of the handshake, the fact that the *Stopp Corona App* is used, a p2pkit-own pseudonymous user ID, operating system, operating system version and model of the device logging in are transmitted to p2pkit. This data, in combination with other metadata of the communication, can be used to draw conclusions about contacts that have taken place.

2.3.3.2. Google Nearby Messages

Google Nearby Messages is used for the purpose of manual handshakes. This service is delivered through the Google infrastructure. Meta data such as IP addresses allow conclusions to be drawn about manually registered contacts.

➔ **We recommend to reconsider the use of p2pkit for the automatic handshake and the use of Google Nearby Messages for the manual handshake.**

2.3.4. Data processing prior to consent

The first time the application is started, the user is shown a short explanation of how the app works. Afterwards he is asked to agree to the data processing in order to use the app. However, several requests are sent in the background when the app is initialized:

- Download of a JSON object from Microsoft Azure that contains the configuration of the application
- Download of infection messages from Microsoft Azure
- Subscription to a Google Firebase topic that is used for push notifications

At least the download of the infection messages and the subscription at Google Firebase are not necessary from a technical point of view before the consent of the data processing. But also the download of the configuration can be solved technically in a way that the user has to agree to the data processing first.

➔ **We recommend that no communication with the ÖRK server or any third party service takes place before the user has consented to the processing of data.**

2.3.5. Traceability of notifications to known public keys within a short period of time

Every encrypted infection message that is published for download on the ÖRK servers is also accompanied by a prefix (first byte of the SHA256 hash of a public key fingerprint). If we now consider a small time interval, it is possible to exclude individual known public keys as potential recipients by downloading all new messages added during this interval from the server, or under certain circumstances to draw conclusions about possible senders and recipients of infection messages if the public keys of all recipients are known.

Example 1: Person A has performed a manual handshake with person B and received the public key PK_B from person B. This enables person A to assign the PK_B key to person B. If person A receives an infection message in the future, he or she can check all newly published messages on the server at that time, and determine whether an infection message has also been stored for the prefix of the PK_B key. If this prefix was not among the newly published messages, person A can rule out that person B has also just received a warning.

Example 2: The fact that an infection message was filed in the prefix of person A during a certain time period increases the probability that person A had contact with an infected person. If a group of persons A, B, C, D, ... has performed handshakes among themselves within a short period of time, each member of this group has the public keys of the other persons PK_A, PK_B, PK_C, PK_D, … in the group. If infection messages are created for the prefixes of PK_A, PK_B, PK_C, PK_D, …in a certain short period of time, it is now more likely that an infection message has been sent to a person within this group.

Example 3: Since a person reporting an infection does not receive an infection message to himself, it would even be possible in the scenario of example 2 to determine which person has reported the infection. If his public key is assigned his own prefix, i.e. in analogy to example 1 the receipt of an infection message can be excluded.

The recognition of temporal correlations of this kind can be made more difficult by changing public keys and the mixing of new messages on the server (cf. so-called mix networks[12]). However, such a concept is currently not used.

➔ **It must be excluded as far as possible that infection messages can be assigned to known public keys of third parties.**

2.4. Security

2.4.1. Android App

2.4.1.1. Operating system permissions

For an app to be able to use certain sensitive operating system permissions, these must first be granted by the user. However, an app may not ask for permission for all possible sensitive operating system permissions, but only for those defined by the development team. Access to the Internet is required for the app to function in general. The app also has permission to disable battery optimization so that the app can run in background and sleep mode. This function is required for the automatic handshake.

The program library of p2pkit requires the following permissions:

- android.permission.ACCESS_WIFI_STATE
- android.permission.CHANGE_WIFI_STATE
- android.permission.CHANGE_NETWORK_STATE
- android.permission.ACCESS_NETWORK_STATE
- android.permission.BLUETOOTH
- android.permission.BLUETOOTH_ADMIN
- android.permission.ACCESS_COARSE_LOCATION

The app is thus able to read and change the state of the network and the WiFi interface. In addition, access to the Bluetooth interface is possible and a Bluetooth connection to other devices can be established. The last authorization allows access to location data, whereby only information from the WiFi interface and the mobile network ("mobile phone network") is processed for this purpose. The accuracy of the location therefore corresponds approximately to that of a block of houses. This authorization does not allow access to the GPS.

[12] https://dl.acm.org/doi/pdf/10.1145/358549.358563

The app also uses Google Nearby Messages, which uses Bluetooth, Bluetooth Low Energy, WiFi information and the microphone (signals in the ultrasonic range) to exchange short messages and then allows a longer exchange of messages over the Internet. Since Google Nearby Messages on Android is a system service, the permissions do not have to be defined by the developers. When using Google Nearby Message for the first time, the user will be asked if he or she agrees to access location data and the microphone.

All in all, the analyzed source code did not show any malfunction of the app with regard to the requested permissions. The requested authorizations are used purposefully. No indication was found that location data or sound recordings are recorded or extracted from the app.

2.4.1.2. Storage of the key pair on Android

The RSA key pair is stored in a protected area of the mobile device via the Android Keystore Provider. All cryptographic operations are called up through special system interfaces, so that not even the app itself has access to the raw key material. The Keystore Provider also ensures that only the app itself is authorized to perform operations with this key pair; other apps are not allowed to access it. The public keys of the devices collected during the handshakes are stored in an SQLite database locally in the app's data directory.

2.4.1.3. Four-digit number of the manual handshake

During a manual handshake, users are shown their own four-digit number and all four-digit numbers of users in the vicinity. Due to the relatively small number of possibilities for the four-digit number, there is a low probability that the same number will be generated for two users.
In this case, these two users could not be distinguished from each other by others looking to complete a handshake. This could be solved by restarting the app.
With a four-digit number, the probability that this event will occur during a certain manual handshake is only 1:10.000. The probability that this will never happen is only about 50% after 7000 handshakes. Considering the number of users, it would therefore make sense to increase the number of digits and/or switch to alphanumeric values.

2.4.1.4. Note on potential exposure to operating system vulnerabilities

Since the application relies on BLE and needs Bluetooth enabled, older unpatched Android versions may be attacked via Bluefrag (CVE-2020-0022)[13].
This is actually a problem of the underlying operating system and the supply of patches from the manufacturer and not the application. However, the application could check the security patch level of the device and refuse service if no patch is installed. In this case the patch level should be at least 2020-02-05 or later[14].

[13] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0022.
[14] https://source.android.com/security/bulletin/2020-02-01.

2.4.2. iOS App

Dynamic tests of the *Stopp Corona App* were performed on an iPhone 7 with iOS 13.3. A deeper insight was gained by applying a jailbreak using checkra1n 0.10.1 beta.

2.4.2.1. Operating system permissions

For an app to be able to use certain sensitive and data protection-relevant operating system functions, these must first be granted by the user. A popup is then displayed in which the user can select whether or not an app may use this function. The popup is then displayed when the app wants to access this particular operating system functionality for the first time after installation, but the decision can be changed later in the operating system settings. A user can thus deny an app access to a certain function, but in many cases the app will then no longer be able to be used properly in practice because central functionalities are based on interaction with the environment (location determination, sound recording via microphone, interaction via Bluetooth, etc). In the specific case of the *Stopp Corona App*, the handshake would then no longer function smoothly.

Furthermore, a particular app may not ask for permissions for all possible sensitive operating system functionalities, but only for those defined by the development team. These permissions can be viewed by looking at the entries in the Info.plist file in the program directory of the installed app, with an additional textual justification by the development team. The following permissions are stored for the *Stopp Corona App*, for which the app can ask for permission if necessary:

- NSBluetoothAlwaysUsageDescription = "The permissions are needed to improve the accuracy of the digital handshake.";
- NSBluetoothPeripheralUsageDescription = "The permissions are needed to improve the accuracy of the digital handshake.";
- NSLocationAlwaysUsageDescription = "The permissions are needed to improve the accuracy of the digital handshake.";
- NSLocationWhenInUseUsageDescription = "The permissions are needed to improve the accuracy of the digital handshake.";
- NSMicrophoneUsageDescription = "The permissions are needed to improve the accuracy of the digital handshake.";

The first two entries concern the use of Bluetooth, although the two entries are there for older iOS versions to provide backward compatibility. According to the documentation, NSBluetoothPeripheralUsageDescription informs all iOS versions from 6 to 13 (not included) about the use of Bluetooth, from iOS version 13 onwards the entry NSBluetoothAlwaysUsageDescription is required.

The entry NSLocationWhenInUseUsageDescription allows a request for the location when using the app, NSLocationAlwaysUsageDescription allows a request for the location while the app is in the background. The latter entry is most likely necessary for the automated handshake introduced in the newer version of the app to work.

NSMicrophoneUsageDescription allows access to the microphone, so that the manual handshake, which is also based on the exchange of sounds in the ultrasonic range via Google Nearby, is possible.

All in all, the analyzed source code did not show any malfunction of the app with regard to the requested permissions. The requested permissions are used purposefully. No indication was found that location data or sound recordings are recorded or sent to the Internet.

2.4.2.2. Storage of the key pair on iOS

The RSA key pair is stored in the keychain, the password vault embedded in the Apple ecosystem. Due to the access attribute kSecAttrAccessibleWhenUnlocked, access to both the public and private keys in the keychain is only possible when the device is unlocked.

The public keys of the devices collected during the handshakes are stored in a SQLite database locally in the app's data directory. The same is true for the own public key; this key is also stored in the same SQLite database, but this is only done the first time an illness message is sent.

2.4.2.3. Note on four digit number of manual handshakes

During a manual handshake, users are shown their own four-digit number and all four-digit numbers of users in the vicinity. Due to the relatively small number of possibilities for the four-digit number, there is a low probability that the same number will be generated for two users.
In this case, these two users could not be distinguished from each other by others looking to complete a handshake. This could be solved by restarting the app.
With a four-digit number, the probability that this event will occur during a certain manual handshake is only 1:10.000. The probability that this will never happen is only about 50% after 7000 handshakes. Considering the number of users, it would therefore make sense to increase the number of digits and/or switch to alphanumeric values.

2.4.3. Backend

The backend is divided into the three components: Push Service Backend, SMS Notification Backend and RCA CoronaApp Backend.

- Push Service Backend: Processing of Push Notifications with the Google Firebase Service
- SMS Notification Backend: Sending TANs via SMS
- RCA CoronaApp: REST endpoints for processing the requests of the mobile applications

A short code review of the source code shows that, with regard to security, it was developed according to modern best practice recommendations. In the review, we focused primarily on high-risk and easily exploitable standard vulnerabilities. The following is a brief summary based on the highest risk vulnerability classes according to OWASP (OWASP Top 10):

1. Injection vulnerabilities: Due to the consistent use of prepared statements and ORM libraries, no SQL injections were discovered. OS command injections are not possible because the backend does not call OS commands with entered data.
2. Error in authentication: The app does not support users or passwords.
3. Loss of confidentiality of sensitive data: On the server side, mobile phone numbers are stored for devices that request a TAN. Furthermore, infection messages are stored on the server side. With the infection message, the mobile app sends a suspected or confirmed case of COVID-19 to the server at the user's request. In addition to the type of message (yellow - suspected case, red - confirmed case, green - health message after suspicion), it contains the mobile phone number and an encrypted message for each stored contact (digital handshake). This procedure was discussed in more detail in section 2.2. Architecture.
4. XML External Entities: No locations could be found where the backend application receives XML data from the mobile apps. Therefore, this method of attack is not possible.
5. Error in access control: The mobile apps have full access to the REST API. No further access control takes place here. REST calls require that the HTTP header AuthorizationKey be set to a specific value. This value is stored with the mobile apps and is therefore quasi public. The header serves its purpose of making unintentional calls more difficult. It should not be given any more importance than this.
6. Safety-relevant misconfigurations: There was no access to configuration data. This point can therefore not be assessed.
7. Cross-Site Scripting (XSS): The back end validates data at the REST interface as recommended in Security Best Practices. Outgoing data complies with the REST standard.
8. Insecure Deserialization: No serialized Java strings are processed in the backend. There is therefore no danger in this respect.
9. Use of components with known vulnerabilities: Using the Maven build files (pom.xml) we were able to track which third-party libraries are used in which versions. A review of these dependencies did not reveal any exploitable vulnerabilities in the third-party libraries used.
10. Insufficient logging and monitoring: This property cannot be assessed in the course of a code review.

Beyond the findings mentioned above, individual safety-relevant problems were identified. These were reported to Accenture and, due to the signed NDA, can only be disclosed after a 15-day Responsible Disclosure Process. The problems identified have no impact on user safety.

2.4.4. Cryptography

2.4.4.1. Deviation from best practice recommendations

In the use case of the application, the encryption of infection messages mainly serves the purpose of proving that the public key was known to the sender of the message. The content of the message does not contain any relevant data except for the rounded time stamp of the encounter and the type of warning (red/yellow/green) and a random UUID for this message.

In the application RSA (RSA/None/PKCS1Padding) with a key length of 1024 bit is used for this purpose. In order to comply with the state of the art, this algorithm would have to be used with a key length of at least 2048 bit (this would make the messages larger and the decryption slower by a factor of about 7)[15]. Also the padding scheme PKCS#1 v. 1.5 (defined in RFC 8017[16]), which is mainly still available for backward compatibility reasons, is used. New applications should, if RSA has to be used, at least rely on an OAEP Padding Scheme, also defined in RFC 8017, to reduce the probability of various attacks on the encryption system (cf. [17]).
It has also become best practice, if RSA is to be used for data encryption, to use hybrid encryption systems, where RSA is only used to encrypt a symmetric key, which is then used to encrypt or decrypt the actual data. In the case of the Stopp Corona App, RSA is directly used to encrypt messages consisting of (padding of 37 "\0" bytes, warning type (Red,Yellow,or Green), timestamp rounded to the hour, random UUID for this message). Apart from that, no data is encrypted by the application using this method.

If, for example for compatibility reasons, no other methods (e.g., ECC) can be used or the protocol can be changed, it is recommended to use at least 2048 bit RSA with OAEP padding which is also available in the javax.crypto.Cipher[18] library used in the app.

2.4.4.2. No verification of warnings

To send a warning to a subscriber, it is sufficient to know their public key. If an encrypted message retrieved from the server could be successfully decrypted by the application, there is no longer a plausibility check whether this warning is really based on a past (even locally recorded) contact, or whether the time contained therein is really plausible. The public key required for valid encryption is thus the only proof of a previous encounter. However, if these keys are made public or, as in the case of p2pkit or Google Nearby Messages, sent via a central location, these keys can be misused to send false warnings to their respective owners even though no actual physical contact has taken place. To do this, an attacker only needs a valid TAN. With a TAN he can send a maximum of 500 such messages of 512 bytes each to the server, which would then deliver them to the respective clients. (See also the remarks on Source Integrity and Broadcast Integrity in Section 2.2.2. Security Characteristics).

In order to make it more difficult to track a single static key and to reduce the probability of such false positives, the key pair used should be constantly changed and the key pair used in the decrypted timestamp should be matched with the key pair used for that time window.

[15] https://www.keylength.com/en/compare/
[16] https://tools.ietf.org/html/rfc8017#section-7.2
[17] https://www.iacr.org/archive/eurocrypt2000/1807/18070374-new.pdf
[18] https://developer.android.com/reference/javax/crypto/Cipher

2.4.5. Communication between client and server

2.4.5.1. Certificate Pinning

When a connection is established to the server, the *Stopp Corona App* checks the TLS certificate of the web service interfaces against a list of public certificates stored in the operating system of the smartphones and issued by Certificate Authorities (CAs). With regard to security against eavesdropping, it is a sensible measure to limit trust in connection with certificate issuance to only one (optionally self-administered) CA ("certificate pinning") in order to further increase security.

If an attacker compromises a single CA, he can perform man-in-the-middle attacks on all TLS connections that rely on the trustworthiness of the entire list of CAs. This is because a TLS server certificate is not hardcoded to a particular CA, but potentially allows any one of these approximately 150 CAs to issue TLS certificates for each domain.

Certificate pinning can significantly reduce this danger, because in this case the client (in this case the *Stopp Corona App*) no longer trusts a whole list, but only one or two CAs or even a specific certificate.

➔ With regard to eavesdropping security, it is a sensible measure to limit trust in connection with certificate issuance to only one CA ("certificate pinning") in order to further increase security. We recommend to include this measure in the next version.

2.4.5.2. Server side TLS configuration

The app communicates with the backend via HTTPS in both the Android and iOS versions. The following TLS configuration applies.

The connection can be established via the following protocols:

- TLS 1.2

The following protocols are *not* allowed:

- SSL 2
- SSL 3
- TLS 1.0
- TLS 1.1
- TLS 1.3

The following cipher suites are allowed for TLS 1.2:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp384r1 (corresponds to 7680 bits RSA)
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (corresponds to 3072 bits RSA)

- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) DH 2048 bits
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) DH 2048 bits

The current TLS configuration is rated A overall by the SSL Labs[19] suite of tests, with the highest rating being A+. The certificate in use and protocol support achieve the full score, while key exchange and cipher strength reach 9/10 points. Overall, the configuration complies with best practice recommendations and is configured satisfactorily.

[19] https://www.ssllabs.com

3. Legal analysis

3.1. Involved actors - data protection law roles

3.1.1. Austrian Red Cross (ÖRK) as controller

Personal data of the users are processed in connection with the download and installation of the *Stopp Corona App*, with the individual data collection processes ("digital handshake"), with the notification (of suspicion) of a Covid-19 illness and with an all-clear signal regarding of a reported illness.[20] The data controller (Article 4(7) GDPR) for these processing operations is the ÖRK, as expressed in their privacy policy (Article 13/14 GDPR).

3.1.2. Processors and technical service providers

The ÖRK does not provide the *Stopp Corona* App on its own, but uses various technical service providers who are (sub-)processors under data protection law (Article 4(7) GDPR) or who are to be qualified as such. Specifically, these are:

Accenture GmbH („Accenture“)

Development, operation (hosting/backend) and maintenance of the software were outsourced to Accenture. Thus, Accenture is processor (Article 4(8) GDPR) for the ÖRK. Some other service providers are sub-processors for Accenture - others are directly data processors for the ÖRK.

Microsoft Corporation („Microsoft“)

Accenture uses the *Azure* cloud service from Microsoft Corporation ("Microsoft") as a sub-processor. Microsoft is subject to the EU-US Privacy Shield, which means that a transfer of data is in principle legal under Article 45(3) GDPR. Point 5.3.1 of the Privacy policy indicates that there are other sub-processors in addition to Microsoft, but leaves open which ones they are, since only Microsoft is mentioned ("*Accenture will provide the hosting and technical operation of the App and the server* [...] *using other individually approved service providers*").

Article 13(1) GDPR allows the disclosure of only "categories of recipients". If these are known, however, it is to be assumed that the individual recipients (i.e. also all sub-processors) must be named. Upon request, Accenture provided a list of Microsoft's sub-sub-processors.

➔ **All sub-(sub-)processors must be clearly identified in the Privacy Policy.**

[20] In addition, there is processing for statistical purposes, which is only dealt with superficially in this report, since the result of aggregation to statistics usually no longer includes personal references.

Uepaa AG („Uepaa“)

The automatic "digital handshake" is implemented by the software p2pkit from Uepaa, which acts as a sub-processor of Accenture.[21] Uepaa uses Amazon Web Services Inc. ("AWS") as sub-sub-processor. A data transfer to Uepaa is generally permissible, since there is an adequacy decision for Switzerland according to Article 45(3) GDPR. A data transfer to Amazon is also subject to an adequacy decision, since Amazon is subject to the EU-US Privacy Shield.

However, the Uepaa privacy policy linked by the ÖRK appears problematic.[22] It is unclear whether this privacy policy is applicable at all if Uepaa only acts as a processor. However, the policy also includes information on “end user data". It was written some two years before the introduction of the GDPR and does not contain all the information required under Article 13 GDPR, nor a clear separation between the role as a processor and that as a controller.

Of particular concern is the following clause: "*You grant Uepaa the right to use, reproduce and distribute Your Use Data and the End User Data in connection with Your use and the End User's use of the Service. Such data will be used to measure, customize, and improve our service*. This would allow the use of data of the persons concerned for the purposes of Uepaa and would probably contradict the allocation of roles according to the GDPR.

The data processing agreement ("DPA") between Accenture and Uepaa was only submitted to us in two short excerpts, otherwise it is subject to a confidentiality clause.[23] The purposes defined in the DPA only cover the core functionality (discovering and communication of two devices). The DPA is thus narrower than the clause in the ÖRK's privacy policy. Use as indicated in the privacy policy is therefore legally prohibited.

➔ **An alignment of the processing purposes between the DPA and privacy notice is urgently needed. Currently, the privacy policy of the ÖRK links to a privacy policy of the Uepaa, which indicates incorrect (because excluded by the DPA) processing purposes.**

Google Group ("Google")

Google (or parts of the Google Group) is used to provide Nearby and Firebase Cloud Messaging services. Nearby is used to handle the manual "digital handshake", Firebase Cloud Messaging is used to send push notifications in the event that a contact partner of the user has announced a Covid 19 illness, suspected illness or an all-clear.

It is irritating that Google is mentioned as a processor in point 5.3.2 and point 5.3.4 of the data protection notice, but sometimes as a separate controller in the data protection impact assessment.

[21] Clarification in an e-mail from Accenture, 14.04.2020.
[22] http://p2pkit.io/pdf/160718_PrivacyPolicy.pdf (accessed on 18.04.2020).
[23] E-mail from Accenture dated 14.04.2020.

When asked about Google Nearby, Accenture stated that Google is "*not a processor but a service used by users on their end devices*". Firebase Cloud Messaging is in turn a processor for the ÖRK.[24] This requires clarification or correction of the privacy policy.

In particular, the blanket reference to Google's privacy policy is therefore incorrect, as this policy applies to cases in which Google is controller for data protection. According to the ÖRK concept, this policy would only be applicable to Google Nearby. For Firebase cloud messaging, however, the privacy policy of the ÖRK, i.e. the controller, would be relevant. Between the ÖRK or Accenture and Google, however, the relevant document is the concluded DPA.

➔ **A clear separation of the two Google services and the respective controller in the privacy policy must be ensured. Only service providers who are actually processors within the meaning of Article 4(8) GDPR may be designated as such.**

Apple Inc. („Apple“)

Firebase Cloud Messaging in turn forwards push notifications for iOS devices to the "Apple Push Notification Service". The role of this service (processor or independent controller) is unclear. Accenture has not yet answered a question about this.

➔ **A clarification in the privacy policy is urgently needed.**

Note on the use of US suppliers

There was general criticism that the ÖRK primarily uses US providers, who may in particular also be subject to US surveillance laws (such as the Cloud Act, EO 12.333 or FISA 702). An obligation to disclose under US law is independent of the company's status as a controller or processor. Especially data on a global pandemic is highly relevant for US intelligence services and fulfils the requirements of US surveillance laws.

When using iOS and Android, there is currently no alternative due to the practical distribution of these operating systems for smartphones. However, it is not to be assumed that data stored locally in the app will be accessed by these corporations and processed further.

Online services such as Microsoft Azure, Google Firebase Cloud Messaging, Google Nearby, Amazon Web Services via p2pkit or the Apple Push Notifications receive at least some user metadata. This could be reduced or avoided altogether with other system architectures.

Even if this processing is legal under the current legal situation (in particular under the EU-US Privacy Shield), the integration of US services seems to be problematic in practice, also because of potential conflicts between the GDPR and US law. Avoidance of these processors is therefore urgently recommended.

[24] E-mail from Accenture dated 14.04.2020.

➔ **The use of alternative processors that do not require data transfer to the United States is recommended.**

3.1.3. Other users as data source and receiver

Finally, it should be considered that in connection with the use of the *Stopp Corona App,* a data transfer to the individual users also occurs. User IDs (see point 3.2.2.) of other users are collected via their end devices when a "digital handshake" is performed or suspicious cases, illnesses or all-clear are reported. These User-IDs are (albeit strongly pseudonymised) personal data.[25]

3.1.4. “Health authorities” and district administrative authorities as data recipients?

Point 5.2 of the privacy policy mentions the "[...] *possibly legally required transmission of information on specific cases of infection to the health authorities at their request* [...]" and refers to § 10 Austrian Data Protection Act (data processing in disaster situations). In addition, "[...] *at the request of the district administrative authorities, the controller* may *be obliged to provide information about suspected cases and infections pursuant to § 5 para. 3 Epidemic Law 1950* [...]".

Although such data transfers would be justified under Articles 6(1)(c) and 9(2)(i) GDPR, the privacy policy leaves open whether such transfers actually take place or whether they are technically foreseen and possible at all; there is no forseen interface for such data transfers. It is also unclear which specific bodies are meant by "health authorities".[26]

In the light of Articles 12(1) and 13(1)(e) of the GDPR, it is advisable to specify the specific recipients or at least the categories of recipients. As the ÖRK, according to its own information, only stores a randomly generated User ID and a telephone number for each user if a Covid 19 illness or suspected illness is reported, it is questionable how useful this data would actually be for the public authorities mentioned above.

➔ **It would be desirable to clearly specify the data that are specifically technically informed and the cases of transmission of information known under Austrian law.**

3.2. Processed data, corresponding processing purposes and legal basis (Articles 5(1)(b), 6 and 9 GDPR)

3.2.1. Download and installation of the app

For the use of the app, no personal data (such as name or date of birth) of the user is required. When the *Stopp Corona App is* installed, the user is assigned a randomly

[25] See in detail point 3.
[26] § 10 DSG speaks of "those responsible for the public sector and aid organisations".

generated static code (Unique Identifier ID, "UUID"), which individualises the user or his terminal device and is to be qualified as a pseudonymous data (see point 3.3.).

Furthermore, an asymmetric key pair is created on the user's terminal device, whereby the public key is used for the exchange during the "digital handshake" and the private key for decrypting received messages (suspicion, confirmed illness and revoking the suspicion).

The purpose of these processing operations is to enable the encrypted exchange of information with other terminal equipment. The ÖRK bases these processing operations on the legal basis of (explicit) consent under Articles 6(1)(a) and 9(2)(a) of the GDPR. The user will be asked to give the following consent when opening the app for the first time:

> *"I agree that the Austrian Red Cross (ÖRK) may process my personal data (unique identification number [ID], the IP address of my terminal device, my telephone number, suspicion and reporting of my COVID-19 illness [=health data]) for the purpose of quickly interrupting the corona infection chain. In addition, my pseudonymous ID is exchanged with my intensive contacts for the purpose of later warning me of a medically confirmed risk of infection".*

The comments in the data protection impact assessment report that this consent does not violate Article 7(4) GDPR can be conformed. Although the usability of the app is linked to the granting of consent, the consent is - as far as can be seen - limited to what is necessary for the proper use of the app.[27]

It is unclear, however, why consent to the processing of the user's IP addresses is obtained. The privacy policy does not indicate for what purpose and in what way the IP address is processed by the ÖRK (in any case for the purpose of communication). It is only stated that neither Uepaa nor Accenture save the IP address. The data protection impact assessment and the FAQs do not provide a more precise explanation for obtaining specific consent either.

It should be mentioned that data processing is already taking place before consent is obtained: Configuration information and infection messages (reports of suspected cases, illnesses and all-clears) from Microsoft Azure are downloaded and registered with Google Firebase. These processing operations are not mentioned in the consent declaration or in the privacy policy linked to it, nor are they referred to in the terms of use. A correction of the privacy policy is necessary here.

It should be borne in mind that consent would not be mandatory for these processing operations. Since no special categories of data within the meaning of Article 9(1) GDPR are processed for this purpose, the processing may be based on other elements of Article 6(1) GDPR.[28]

[27] Data Protection Impact Assessment Report , page 34, 35.
[28] A contractual basis already exists before consent is obtained: According to the general terms of use, "*the license agreement for the use of the software copy of the app [...]* begins *with the download and ends with the deletion of the app on the user's terminal device*".

➔ **It is necessary to clarify the purpose for which IP addresses are processed and the legal basis of the data processing operations that take place before consent is given.**

3.2.2. Acquisition process (“digital handshake”)

During the automatic "digital handshake" (only possible between Android devices), randomly generated codes are created via the *p2pkit* of Uepaa ("User-ID";[29] not to be confused with the static UUID) and transmitted to the servers of Uepaa together with time information, the device model, as well as operating system information. After confirming that the two devices are sufficiently close (via Bluetooth and Wifi Direct), the User IDs are exchanged between the two devices involved, using the asymmetric key pair mentioned in point 3.2.1. The User ID of the contact partner is stored locally on the user's terminal device.

With the manual "digital handshake", Google Nearby is used instead of p2pkit to exchange data, whereby Bluetooth, WLAN and also ultrasonic signals can be used.

The purpose of these processing operations is the technical registration of the contact persons of each user, thus creating a contact diary on the basis of which notifications can be sent in two directions: On the one hand, if the user himself falls ill with Covid-19, suspects an illness or revokes the suspicion, on the other hand, if this is the case with one of his registered contacts (for details see point 3.2.3.). As a legal basis, the ÖRK again relies on the consent mentioned in point 3.2.1.

3.2.3. Incident (reporting of suspected cases, illnesses and all-clears)

If an user is qualified as a suspected case of Covid-19 illness based on his (non-verifiable) information in the Symptom Checker ("suspected case", yellow), she/he can voluntarily report this result via the app. The same applies if a medical certificate confirmed a Covid-19 illness to the user ("illness", red), whereby here again no verification of the user's details is provided. Finally, an user can revoke a suspicion ("all-clear", green), which is particularly useful for taking a negative Covid-19 test result into account in reported suspect cases.

The purpose of these processing operations is evident and consists in the notification of these events to the user's contact persons within the last 54 hours, which is done via Google Firebase cloud messaging by push notification. The recipients of the notification are then required to take appropriate measures (on their own responsibility) to break the chain of infection. It is noteworthy that the general terms of use specify a different, or significantly more vague, notification period: "*Only those contact persons with whom the user has been in contact in the past 3 calendar days will be notified."It is* recommended to specify this more precisely.

Before sending a message, the user is asked to enter a mobile phone number to which a TAN is then sent by SMS. The message will only be sent once the TAN has been entered correctly. The purpose of processing is to prevent abusive (intentionally false) messages

[29] These User-IDs are automatically changed every 14 days.

(inhibition threshold by disclosure of the telephone number), as well as the possibility of contact by the ÖRK for assistance.

Once again, the consent referred to in point 3.2.1. serves as the legal basis for the submission of reports. In cases of abuse, the telephone number is processed on the basis of Article 6(1)(f) GDPR (legitimate interests) and Article 9(2)(f) GDPR (establishment, exercise or defence of legal claims).

However, prior to the first use of the Symptom Checker, separate consent is obtained:

> "*I agree that the Austrian Red Cross (ÖRK) may process my personal data (unique identification number [ID], the IP address of my terminal device, my telephone number, suspicion and reporting of my COVID-19 illness [=health data]) for the purpose of quickly interrupting the corona infection chain. In addition, my pseudonymous ID is exchanged with my intensive contacts for the purpose of warning of a risk of infection and for the later all-clear or confirmation of the infection.*
> "

This consent differs only slightly in the last sentence from that mentioned in point 3.2.1., in that revoking the suspicion or confirmation of an infection is also mentioned. Why separate consent is obtained for processing in connection with the Symptom Checker is not apparent and is not clear from the documents provided. The second consent does not provide additional transparency: it is almost identical to the first consent; both consents refer to the same privacy policy. The avoidance of bundling several consents cannot be decisive either, since the consent obtained at the first use already refers to several (and to a large extent the same) processing operations. Finally, no separate revocability of the second consent is required, since the answers provided by the user (health data) are discarded immediately after completion or discontinuation of the questionnaire anyway. Here it would be worth considering to explain the processing operations in connection with the Symptom Checker in the privacy policy in a more understandable way and to use only one single declaration of consent.

- ➔ **The need for a second, separate consent for the symptom checker should be reconsidered.**
- ➔ **The notification period (54 hours) should appear uniformly in all documents.**

3.2.4. Statistics

Finally, for statistical purposes, the number of app installations as well as the number of handshakes and messages are recorded together with hourly time stamps.

The legal basis for these processing operations is § 7(1)(2) Austrian Data Protection Act (DSG) in conjunction with Article 9(2)(j) GDPR. On the basis of the statistics, the aggregated usage behaviour of users and the distribution of handshakes throughout the day will be determined. It will also be determined whether the average user reacts to warnings.

As described in the technical part of this report, there are concerns as to whether the implementation of this function complies with the requirement of data minimisation (Article 5(1)(c) GDPR). If technically possible, a more data-efficient implementation is required.

➔ **There are concerns whether the statistics meet the requirement of data minimization.**

3.3. Pseudonymisation and data minimisation (Article 5(1)(c))

As explained (point 3.2.1.), the user does not provide the ÖRK with any data such as name, date of birth, etc. Only when submitting a report (suspected case, illness, revoking a suspicition) a mobile phone number must be provided in order to receive a TAN via SMS, which is used to release the report.

Data generated by the ÖRK to uniquely identify the user or to enable encrypted communication with other users (UUID, User ID, asymmetric key pair) are all to be qualified as pseudonymous data within the meaning of Aricle 4(5) GDPR, which only contain an indirect personal reference.

In this respect, the comments in the data protection impact assessment report and the remarks in the FAQs are to be endorsed: In the light of the CJEU's case law in *Breyer*[30] pseudonymous and not anonymous data must be assumed, despite the low probability of inference to the person behind the identifiers provided by the ÖRK. The GDPR is therefore fully applicable; the use of pseudonymous data is, however, a sensible technical measure within the meaning of Article 32 GDPR which contributes to minimising the risks of data processing.

However, the telephone number possibly provided by the user is to be regarded as directly personal data, as the user can be contacted directly. It should be noted that according to the information currently available, the UUID is apparently not linked to the telephone number provided by the user to obtain the TAN.

The implementation of the app without requiring the disclosure of directly personal data for the use of the basic functionalities is very welcome from the point of view of data minimisation (Article 5(1)(c) GDPR) and in this respect also takes into account the requirements of Privacy by Design pursuant to Article 25 GDPR.

3.4. Storage limitation (Article 5(1)(e) GDPR) and data deletion by the user

3.4.1. Withdrawal of consent and deletions by the user

Since the majority of data processing is based on the consent of the user, a withdrawal of consent must lead to the deletion of these data, unless there is another legal basis on which

[30] CJEU 19.10.2016 C-582/14.

the data are processed (see Article 17(1)(b) GDPR). According to the information available, this must concern the UUID, the User IDs and the asymmetric key pair, but not the data generated or disclosed in the course of a notification (on these see point 3.2.3.).

The declaration of consent and the privacy policy do not transparently explain how a withdrawal can be made. Point 4.2. of the privacy policy merely states that uninstalling or deleting the app implies a withdrawal. A partial withdrawal is also possible by deactivating the automatic "digital handshake", for which there is a button in the app.

However, this possibility is only discussed in the data protection impact assessment report,[31] but is not included in the privacy policy, which would need to be supplemented in this respect. According to the data protection impact assessment report, consent can also be withdrawn "*at any time after the transmission of the COVID-19 notification of illness or suspicion [...] to the data protection officer of the controller by e-mail, telephone or post.* "

It is completely unclear what effect such a revocation directed to the data protection officer will have on the basic functionality of the app (which in this case remains on the user's terminal device). It is also unclear how the data subject should identify himself or herself to the ÖRK (see in detail point 3.6.). In the interest of maximum transparency, the privacy policy would have to be supplemented accordingly. Moreover, the modalities of withdrawal should already be indicated in the consent form(s).

The following passage in point 6 of the privacy policy is also misleading: "*As a rule, your personal data is only stored in the app for the duration of use. You can delete this data yourself at any time on your end device.*" On the one hand, if this wording is used, the exceptions to the rule mentioned would also have to be mentioned. On the other hand, the sentence suggests to the user that he/she can manually remove individual data objects (such as individual handshakes) from his/her terminal device. However, as far as can be seen, this is only possible by uninstalling or deleting the entire app.

➔ **The modalities and effects of a withdrawal of the consents granted should be presented in a comprehensible manner. Misleading formulations that hold out the prospect of manual deletion of individual data should be rewritten.**

3.4.2. Data deletions apart from withdrawal / retention periods

In addition to the user's option to request the deletion of personal data by withdrawing her/his consent, personal data will be deleted according to the present documents after the following retention periods have expired:

The randomly generated User-IDs are transmitted to Uepaa together with a time information, the device model and operating system information and stored there for up to 14 days (point 5.3.3. and point 6. of the privacy policy).

[31] Data protection impact assessment report, pages 31-35.

However, the privacy policy does not indicate how long individual "digital handshakes" are stored; page 83 of the data protection impact assessment report states: "*The user's digital handshakes are available for the last 7 days and are automatically deleted after that.*"

This cannot be brought into line with the functioning of the app in reality. "Handshakes" (no matter if manual or automatic handshakes) were still visible in the listing in the app. None of the handshakes disappeared - not even after more than 17 days. However, the older handshakes were not included when a suspicion was reported. It seems that the local list does not include a deletion routine, but only the contact details of the relevant period are informed. Upon request from Accenture, they assured that the relevant deletion routine will be included in the next release.[32]

Data transmitted in connection with the notification of a suspected case or illness will be stored for 30 days after the notification is made. This includes in particular telephone numbers provided by the user. If there are concrete indications of unlawful or abusive behaviour, the data will be stored for a period of up to three years after the notification of the illness has been made (point 4.4. and point 6. of the privacy policy). Point 7.1.6. of the privacy policy also points out that the processing of a telephone number can be objected to within the 30-day storage period in accordance with Article 21(1) GDPR. Consequently, this must also apply during the possible three-year storage period, since Article 6(1)(f) GDPR is also used as the legal basis here. The privacy policy would have to be supplemented in this respect.

After the end of the epidemic, all data are to be deleted, although it is acknowledged that an end is not foreseeable at present.

It is not clear for how long IP addresses are stored. As explained under point 3.2.1., the user agrees to their processing, although the concrete purpose of processing remains unclear.

- ➔ **Like the purpose of processing, the storage period of IP addresses remains in the dark. This needs to be improved.**
- ➔ **The storage period of the "digital handshakes" must also be indicated in the privacy policy. The app must actually delete the handshakes immediately after the deletion period.**
- ➔ **Finally, the data privacy policy must indicate that there is a right of objection under Article 21 DSGVO with regard to all processing operations carried out on the basis of legitimate interests.**

3.4.3. Conflict of objectives between storage limitation and data accuracy

The current storage periods of the app are very short (also in international comparison). In practice, it is quite conceivable that people will only recognize an illness much later than an infection with SARS-CoV-2. Data may already have been deleted at this time. It therefore seems that a differentiated storage period would also be possible in the sense of data accuracy (see below). In this case, the period of relevant information could be chosen

[32] E-Mail from Accenture dated 15.04.2020.

depending on the individual case (including the progress of the illness) in order to select a correct time window.

3.5. Problem of data accuracy (Article 5(1)(d) GDPR)

3.5.1. General points

According to Article 5(1)(d) GDPR, the controller (here the ÖRK) must also ensure the accuracy of the data within the data application. Particularly in the case of information on health and data on the spread of SARS-CoV-2, a high level of data quality must be applied here.

3.5.2. Consequences of information

In practice, one has to think, for example, of the threats of punishment in §§ 178 and 179 StGB (Austrian Criminal Code), which punish an intentional or negligent endangerment of people with contagious diseases with up to three years imprisonment. An user who does *not* initiate quarantine measures after being informed about the possible infection would probably have to account for himself according to these regulations if it later turns out that she/he was SARS-CoV-2-positive during the relevant period. Civil liability would also be conceivable (for example, if an employee continues to work despite a warning, thereby bringing a company to a standstill). Users would therefore probably have to (at least) indirectly accept the information displayed in the app.

In the case of misinformation, however, the user would be restricted in his freedom without reason, since she/he must assume that the information is correct in case of doubt and cannot verify it (also because of the anonymity of the information).

In summary, there are possible massive interferences with the rights of app-users in case of misinformation by the app.

3.5.3. Detailed information

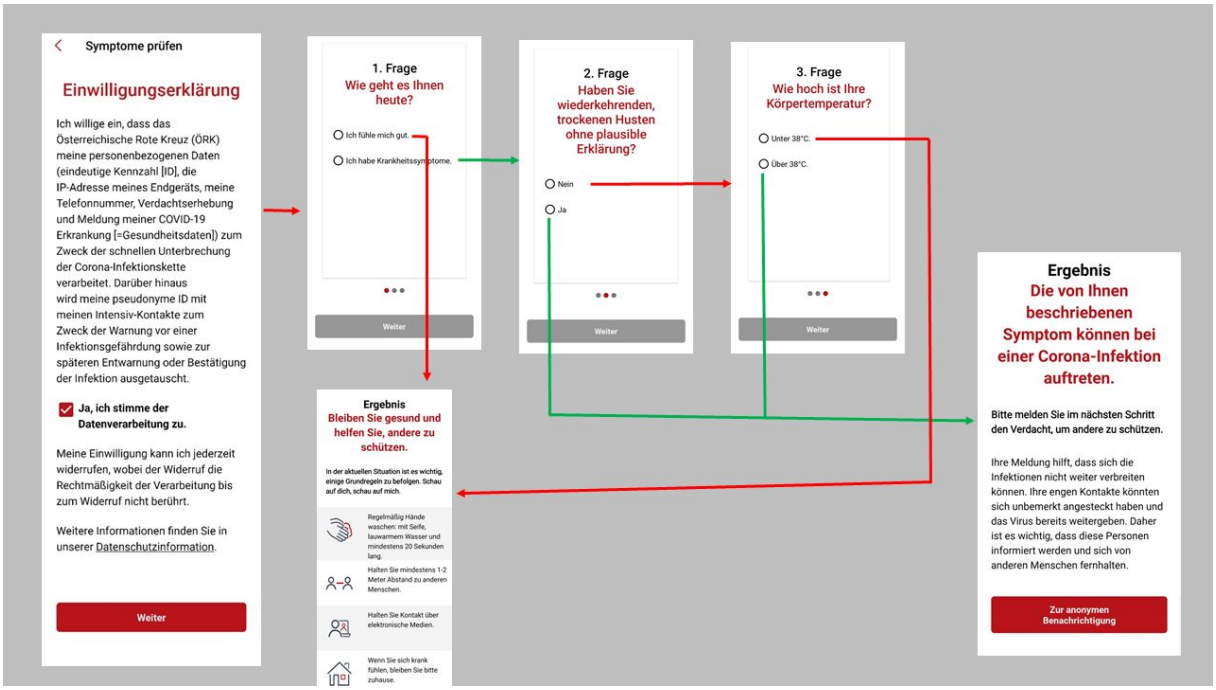
On the positive side, the ÖRK's concept of allowing differentiated reporting according to (1) suspected case, (2) confirmed illness (positive test and) and (3) the revoking of suspected cases is welcomed. With these different types of information, an user can actually provide correct information or even correct incorrect information. Other concepts so far only assume a warning, which may no longer be corrected.

3.5.4. Lack of verification of user data

If a person is *de facto* not SARS-CoV-2 positive, but the contact person receives such an information, the processed data is not " accurate" as defined in Article 5(1)(d) GDPR. As the

ÖRK leaves it up to the users to enter a warning, positive message or revoking of a suspicion, there are concerns whether in practice data accuracy can be guaranteed:

Currently, the logic of the self assessment (see below) leads to the instruction that the user should trigger the warning function as soon as the questions on a "dry cough" or "high fever" are answered positively. A further checks by a third party do not take place. App-users who are convinced that they are SARS-CoV-2 positive will probably regularly answer one of the questions positively and will then be encouraged to trigger the warning function.



Even at the peak time of corona spread, when only very restrictive and targeted testing was possible, only about 20% of the tests were positive. Currently, this value is about 15%. It can therefore be assumed that the population on a large scale merely suspects an infection, but is in fact not infected. According to the ÖRK, such a "high sensitivity" (i.e. the inclusion of as many people as possible) is what is sought to ensure that every suspected case and contact is investigated.[33]

The logic of the app also does not require the user to take a Corona test or revoke a suspicion later. Users mAY therefore not take any further steps after a warning has been issued. According to the ÖRK, the proposed quarantine is limited to 7 days, as it is assumed that either a test has been performed or the user has forgotten about it "*A suspected case which is not confirmed or rejected within 7 days* [...] *has probably been forgotten* [...]."[34] This raises the question of whether misinformation leading to a massive encroachment on the rights of the warned person is not knowingly accepted by the current logic.

[33] E-mail from the ÖRK, 19.04.2020.
[34] E-Mail from Accenture dated 15.04.2020

3.5.5. Reasonable steps?

Article 5(1)(d) GDPR does not require absolute accuracy of data, but the taking of "reasonable steps" to ensure accurate data. In view of the massive consequences of false information (lost freedom of movement), a high standard must be applied.

The verification of the user by means of a telephone number (TAN) seems to be unsuitable in most cases, as overly cautious users cannot be blamed and even in the case of abuse it is not clear how a willful false information should be revealed in the case of information that is anonymous to the recipient.

Other concepts are based on "tokens" (codes) provided by a third party (health hotline, doctor, etc.) when a suspicion is realistic or when a test is positive. This "token" would then be necessary to send the notification via the app. This would also have another advantage: These tokens can be assigned randomly, which would avoid the registration of users with a mobile phone number, while at the same time massively increasing protection against misuse.

The encroachment on the rights of the warned person could, for example, also be minimised by providing more concrete information and thus allow for more planning. For example, a warning could also include the concrete date of a test and the date of the expected test result instead of indicating a blanket (and thus probably often wrong) deadline of seven days. This date could also be encoded in a token that is issued when the test date is agreed.

This would ensure that the suspected case (1) has made a test appointment, (2) has described its symptoms to at least one third party and (3) the person being warned is shown a more correct quarantine time by the app.

According to information from the ÖRK,[35] the planned Containment 2.0 strategy for the epidemiological containment of the disease seems to accept a high number of false warnings. As a result, it is considered acceptable to quarantine a larger number of people through the app, even if 90% of them are not infected, rather than subjecting the entire population to similar measures. However, options such as reminding user to revoke the warning or a shorter quarantine period in case of a warning once faster testing is available have been brought into play.

➔ **Even though it is clearly outside our expertise to evaluate epidemiological concepts, there do seem to be "reasonable steps" (Article 5(1)(c) of the GDPR) to avoid or to correct as far as possible false information that were not taken.**

3.5.6. Duration of the notification

In the test case, notification of a suspicion lasted about 15 minutes and was proactively indicated by a pop-up on the contact person's device. When the all-clear was sounded, no

[35] E-Mail from ÖRK dated 19.04.2020.

message was received on the contact person's device for over an hour. Only after the app was opened again did the all-clear signal appear.

According to information from Accenture, information should be visible on the device within one hour, through a "silent push". If this does not work, there must be a technical problem.[36]

➔ **Since app-users sometimes have to make fundamental decisions, it is important to ensure that information is transmitted without delay. It is recommended that the transmission times be specified in the privacy policy or in the FAQs.**

3.5.7. Automated individual decision-making under Article 22 GDPR?

In this context, it should also be considered whether the information provided by the app does not also constitute an "automated individual decision" under Article 22 GDPR. In the current state of the app, however, the existence of a "decision" must be denied by the app itself for lack of a certain logic. The app still appears to be primarily a neutral information system.

However, as soon as the app makes a (welcome) more precise selection that incorporates various factors such as the proximity, duration and time of an infection and thus makes a selection of the warned contact persons, this question would need to be examined further. In its Guidance on Apps supporting the fight against Covid 19 pandemic in relation to data protection, the European Commission has also drawn attention to this problem.[37]

3.6. Assertion of rights of data subjects

Point 7 of the privacy policy lists - as required by Article 13/14 GDPR - the rights to which users are entitled as data subjects in accordance with Article 15 et seqq. GDPR. Without going into detail about these rights, the central question is whether and how an user can actually exercise these rights vis-à-vis the ÖRK:

The own UUID assigned during installation or the User-IDs used for handshakes are not known to the user; they are neither displayed in the app nor is there a possibility to export them. Other identifiers (such as name, date of birth, address or device ID of the terminal device) are not known to the ÖRK, nor are the phone numbers provided linked to the user's UUID.

The question therefore arises as to how an user can clearly identify himself or herself to the ÖRK in order to exercise her or his rights under Article 15 et seqq. GDPR. Both the privacy policy and the data protection impact assessment report do not provide an answer to this question; in point 7.1.8. of the privacy policy, only the data protection officer is mentioned as

[36] E-mail from Accenture dated 15.04.2020.
[37] See page 7 https://ec.europa.eu/info/sites/info/files/5_en_act_part1_v3.pdf (accessed on 19.04.2020).

the contact point for data subjects' rights and an e-mail and postal address and a telephone number are provided.

According to Article 11(2) GDPR, the rights of data subjects under Articles 15 et seqq. GDPR do not apply if the controller can demonstrate that it is not in a position to identify a data subject. However, the ÖRK cannot rely on this provision, as the static UUID is an unchangeable identifier, which allows a unique identification of each user. The only problem here is that the own UUID is not accessible to the user.

Article 12(2) GDPR requires the ÖRK to facilitate the exercise of the rights of data subjects under Articles 15 to 22 GDPR. It is therefore essential to allow access to one's own UUID. If this is not implemented, there is also a serious problem with regard to the principles of transparency and fair processing pursuant to Article 5(1)(a) GDPR, as the privacy policy suggests to users that they can easily exercise their data subjects' rights.

This identification problem also arises by analogy with the possibility of withdrawal vis-à-vis the data protection officer (see above, point 3.4.1.).

In general, it should be considered to enable the user to exercise his rights to information (Article 15 GDPR) and data transferability (Article 20 GDPR) by allowing the export of the app's current local actual data stock at any time (in particular download UUID, User-IDs used, messages and contacts of the last 54 hours).

➔ **At least her/his own UUID has to be made visible to the user or an alternative possibility of clear identification has to be created to enable the exercise of data subjects' rights and the revocation of given consent. Otherwise, there is a risk of legal protection deficits, as no user can identify herself/himself in practice to exercise her/his rights.**

4. Previous Versions

Version	Date	Comment
0.3	20.04.2020	Overview of recommendations added
0.2	19.04.2020	Inclusion of 1.1, 1.2 und 2.3.5; Editing of open questions
0.1	18.04.2020	Raw version, for the Austrian Red Cross and Accenture