WeTrace: A Privacy-preserving Tracing Approach

Muriel Franco, Bruno Rodrigues, Christian Killer, Eder John Scheid,
Alessandro De Carli, Andreas Gassmann, David Schönbächler, and Burkhard Stiller

Abstract: **For the protection of people and society against harm and health threats — especially in case of the COVID-19 pandemic — a variety of different disciplines needs to be involved. The data collection of basic and health-related data of individuals in today's highly mobile society does help to plan, protect, and identify next steps health authorities and governments can, shall, or need to plan for or even implement. Thus, every individual, human, and inhabitant of the world is the key player — very different from many past crises'. And since all individuals are involved his/her (a) health and (b) privacy shall be considered in a very carefully crafted balance, not overruling one aspect with another one. Privacy remains key.**

The solution of the current pandemic's data collection can be based on a fully privacy-preserving application, which can be used by individuals on their mobile devices, such as smartphones, while maintaining at the same time their privacy. Additionally, respective data collected in such a fully distributed setting does help to confine the pandemic and can be achieved in a democratic and very open, but still and especially privacy-protecting manner.

Therefore, the WeTrace approach and application designed utilizes the Bluetooth low energy (BLE) communication channel, many modern mobile devices offer, where public-key cryptography is being applied to allow for deciphering of messages for that destination it had been intended for. Since literally every other potential participant only listens to random data, even a brute force attack will not succeed. WeTrace and its Open Source implementation ensure that any receiver of a message knows that this is for him/her, without being able to identify the original sender.

Index Terms: **Bluetooth low energy (BLE), contact tracing, COVID-19, privacy-preserving, mobile.**

I. INTRODUCTION

SOCIAL distancing is one of the essential measures to prevent the spreading of COVID-19 in a population. The pandemic situation during 2020 and 2021 indicated that any step toward re-establishing society's regular economic activities shall be performed carefully by health authorities and governments to prevent new infection waves. Thus, the use of novel and already rolled-out technology is essential and almost the only way *(a)* to crowd-source information concerning the health of individuals and *(b)* to ensure that social distancing rules are being respected. These two requirements stem from the perspective of a general epidemic analysis and urgently need to be combined with technical support such that *(i)* the individual's privacy, *(ii)* the personal freedom of an inhabitant of Switzerland and many other countries — in contrast to a selected list of differently organized countries of the world —, and *(iii)* the avoidance of "finger-pointing" to select humans can be reached technically and efficiently in the shortest possible time frame.

For instance, the infection's spreading rate based on currently available data [1] (e.g., in how many days does the number of infected individuals doubles) imply that, as observed in some cases in Italy or Spain, individuals cannot be examined because the hospital's infrastructure may be overloaded. Thus, awaiting for a doctor's signaling that an individual is infected may be too late to prevent further infections from that individual. Furthermore, the asymptomatic period, where individuals are not aware of his/her infections, can imply that a system may emit distance alerts regardless of whether a mobile device flagged an unidentified human being infected is nearby or not. As of today, individuals carry such mobile but different devices with different communication choices being integrated. These devices include phones — generally termed smartphones —, tablets, and laptops. Each of them is connected through partially selectable communication technologies, such as Bluetooth (BT), Bluetooth low energy (BLE), Wi-Fi (IEEE 802.11 family of protocols), Wi-Fi direct, global system for mobile communications (GSM), universal mobile telecommunication system (UMTS), high-speed downlink packet access (HSDPA), long-term evolution (LTE), or LTE advanced (LTE-A) (often termed 3G, 4G, and partially already moving into 5G technologies). Many of those alternatives allow for the communication of a mobile device's position to nearby devices, even further on.

WeTrace, as introduced here as a technical response to the COVID-19 pandemic, is one of the first of its kind as of April 2020 to address a fully-preserving approach and applications [2], [3]. The focus was laid on BLE, since all 3G to 5G communication technologies deployed already allow for an identity tracking of a device used. The subscriber identity module (SIM) card-based identification of communications, for example, can always reveal the current user's (subscriber) identity, who was registered with the respective contract. Thus, a clear demand beyond public telecommunication system-based communications for full privacy-preserving tracking and tracing was urgently required [4]. Note that WeTrace offers as well a solution to meet major European union (EU) general data protection regulation (GDPR) requirements [5].

The WeTrace application developed fulfills exactly this key requirement on privacy-preserving for arbitrary mobile devices, communicates via BLE, and is used by their owners in a once-used, once-associated manner. This means that the underlying

Manuscript received February 19, 2021; revised June 16, 2021; approved for publication by Zhu Han, Guest Editor, June 22, 2021.

This paper was supported partially by *(a)* the University of Zürich UZH, Switzerland and *(b)* the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, the CONCORDIA project.

M. Franco, B. Rodrigues, C. Killer, E. J. Scheid, and B. Stiller are with Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH, email: {franco,rodrigues,killer,scheid,stiller}@ifi.uzh.ch.

A. D. Carli, A. Gassmann, and D. Schönbächler are with Papers AG, Switzerland, email: {a.decarli,a.gassmann,d.schoenbaechler}@papers.ch.

M. Franco is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2021.000021

assumption of this work is, as many others do, but do not necessarily state that explicitly: "One mobile device belongs to one individual and is used by this individual only during the pandemic". Furthermore, the application of low-range BLE communications determines a highly suitable coincidence between the COVID-19 "social distancing" requirements and the communications technology: Since only those individuals in a range of a few meters (if staying that close together for approximately 10 to 15 min) potentially are subject to contagious infection, their mobile devices can, if the WeTrace application is enabled and configured, exchange health data in a fully privacy-protecting manner, such that infection status information can be exchanged fully anonymous and in a secured manner.

Finally, WeTrace as a tracing approach and application will not depreciate over a short period of time, since the fully privacy-preserving characteristic can serve for future health-related as well as other privacy-preserving data collection applications, such as high-density events in entertainment cases or natural disasters, like earthquakes or tsunamis, where many independent individuals can report, and a respective view on locally residing individuals can be fed back.

Therefore, the WeTrace approach shows a measurable technical and societal contribution for the support during the combat of pandemic situations as the one observed during the COVID-19 crisis. The main contributions of WeTrace are:

- A detailed analysis of the technical and legal requirements of contact-tracing applications;
- The design of a decentralized and fully anonymous approach based on BLE, global positioning system (GPS), and public-key cryptography mechanisms to report infections between users who had been in close contact for a certain period of time;
- The summary of major implementation details of an open-source WeTrace prototype that can be integrated with COVID-19 contact tracking and tracing applications; and
- Evaluations and discussions concerning crucial aspects of the WeTrace approach, especially privacy-preserving, security, energy consumption, and scalability.

This paper is organized as follows. While Section II summarizes major-related work and compares those against the key dimensions of utmost importance set for a privacy-preserving tracing application, Section III determines the essential requirements needed to meet the goal of hiding in full the privacy of an individual participating. Derived from those requirements, the WeTrace approach is introduced in Section IV and relevant technical details are provided. Section V follows with major specificities of its technical implementation. While Section VI discusses the trade-offs taken as well as observations obtained with respect to proximity and privacy, Section VII provides evaluations and findings on the communication channel used, attacks mitigated, storage overhead, energy consumption, and scalability. Finally, Section VIII summarizes and concludes WeTrace as one essential step into a fully privacy-preserving tracking and tracing application world.

II. RELATED WORK

Since the formal declaration of COVID-19 being a worldwide pandemic, progressing at a swift pace, a set of different projects and tools have been proposed to implement automated notification systems based on contact tracing [6] and approaches to help in the social distancing [7]. These systems act as a supporting tool to ensure the re-establishment of society's everyday activities while safely maintaining human health as the primary goal. For example, the Swiss government implemented the SwissCovid App [8] based on DP3T [9], which allows for users' infection self-reporting after a COVID-19 case confirmation. Of course, systems for tracking and tracing exist, and various mobile devices show a variety of different but related functionality, including the search option for a displaced smartphone, once it had been misplaced or stolen. However, many if not all such approaches are known to trade-off the individual's privacy or the human's data privacy in one way or another. Thus, the notification or alerting option of today's tracking and tracing systems has not reached an acceptable standard concerning user privacy and user data privacy.

Thus, it is highly important, especially with respect to an application collecting health data, to observe security and, more precisely, privacy concerns of all solutions proposed so far, which address COVID-19 tracing anonymously. Privacy includes the privacy of the user and the privacy of user data, here a medical status. Additionally, this includes for sure non-negotiable guarantees that these systems cannot be used as *(a)* attacking tools to the availability of other systems, *(b)* jeopardizing the privacy of users themselves, *(c)* violating the privacy of user data, and *(d)* revealing at any step performed private and to be secured personal (health) data. In this sense, Tables 1 and 2 collects the major characteristics of related systems in order to clarify the state-of-the-art concerning tools and proposals related to COVID-19-driven tracing and tracking, thus providing an overview of its applications, privacy, and technology characteristics. An extensive report of further related work collections performed by several contributors can be found at [23]. A global report on approaches by governmental and private projects to use personal data to combat COVID-19 can be found at [6]. However, since this work here on WeTrace addresses user privacy and user data privacy as its priority, the constructive work to define and specify an appropriate solution is considered to be more relevant at this stage than collecting yet another complete view of country-specific approaches underway these days.

Nevertheless, since the pandemic is relatively recent and is spreading in some countries of the world at a swift pace (an exponential growth was observed in certain countries [1], in many others it had slowed down slowly at the time of writing [24]), it is important to note that most of these approaches as referred to above are theoretical proposals or are still under development. Thus, the detailed information on these technological solutions and approaches might depreciate over a short period. Therefore, more general and generic applications will help sustain the work and research invested. As such, the Sismo approach [22] was proposed for earthquake notifications and now is being used to operate for COVID-19 notifications.

Different survey map solutions focusing on COVID-19 tracings, such as of [25], [26]. However, key dimensions are still missing for a clear view of the state-of-the-art. Therefore, this paper does not repeat such surveys, rather it determines nine key dimensions, which had been selected to classify and com-

Table 1. Solution's overview and privacy comparisons.

Solution	Open source	Reporting	Data collected	Privacy-preserving mechanism applied
WeTrace	Yes	Self-reporting	GPS location history, encounter timestamp	Public-key cryptography, GDPR-compliant privacy, fully anonymous
CoroTrac [10]	No	Self-reporting	GPS location history	Data anonymization model
CovidWatch [11]	Yes	Self-reporting	GPS location history	GPS anonymization model
Pandoa [12]	Yes	Self-reporting	GPS location history	GPS anonymization model
NextTrace [13]	No	Provided by Labs and self-reporting	Location and proximity data	Data anonymization model
geoHealthApp [14]	No	Claimed to be AI-based	GPS location history	Blockchain, claimed GDPR compliant
CoronaTrace [15]	No	Self-reporting	GPS location history	User data anonymization, not publicly visible individual information
TraceTogether [16]	No	Self-reporting	Location and location of nearby devices	Encrypted BLE channels
DP3T [9] part of PEPP-PT [17]	Yes	Self-reporting	GPS location history, encounter timestamp	Ephemeral identifiers, pseudonymous for research volunteers, GDPR
NextStep [18] part of PEPP-PT [17]	Partially	Self-reporting	Only P2P Bluetooth encounters	Encrypted (not disclosed)
NOVID20 [19]	Yes	Self-reporting	GPS location history, Bluetooth, and Google	Encrypted (not disclosed)
Luca-app [20]	No	Self-reporting and direct connection to health authorities	QR code for self-checking into locations, GPS location history	Encrypted, but available to health departments
StopCorona [21]	No	Self-Reporting	Mobile number for 30 days, Bluetooth, nearby audio	Pseudonymous, collects user information for research purposes
Sismo [22]	No	Self-reporting	GPS current location	Pseudonymous, collects user information for research purposes

pare relevant related work (*cf.* Tables 1 and 2), thus, providing a focused view on practical, open-source, and privacy-preserving research and development aspects:

1. **Solution**: Proposed tool or respective solution, including its current name(s) and currently available key reference(s).
2. **Open-source**: Determination of whether the code is or will become publicly available. This is essential for verification of the privacy-preserving property of the approach as well as other security metrics
3. **Reporting**: Concerning whether users applying this approach within their smartphone are able to flag themselves being in one certain medical state, e.g., "not infected", "close contact with infected", "infected", "infected, being with symptoms", "indifferent", or "healthy". There is no "cured" state, as this state is considered as "not infected". However, since actual states are not important for the WeTrace application's operation (any data can be integrated into a message unless it grows too large for the Backend to publish), medical states required have to be defined with epidemiologists.
4. **Data collected**: Determining the type of data being collected and processed, generally that is possible from the user and/or a Backend (*cf.* below) if involved. These data may include GPS data for geo-localization, timing-related information, medical status (*cf.* before), communication addresses, or phone numbers.
5. **Privacy-preserving mechanism**: As it is mandatory that any solution or tool does not violate the users' privacy, e.g., by revealing their identity, health conditions, or geographical location, security, and risk analysis of this or these mechanisms foreseen or deployed does determine the level of privacy reached.
6. **Communication technology**: The technical communication solution selected enables the collection of relevant (or irrelevant) information on users' encounters. The focus is on BLE and GPS. Approaches with Wi-Fi communications or any 3G to 5G communications are not listed, since *(a)* all of them do limit by definition the users' privacy due to the use of Internet protocol (IP) addresses or phone numbers since Internet service providers are required by law in selected countries to keep track of the "owner" of an IP address for a certain time or *(b)* the use of SIM cards legally requires registration of users with their full identity, respectively.

Table 2. Solution's technology comparisons.

Solution	Communications technology	Data storage	Backend
WeTrace	BLE	Decentralized, locally at the device	Run by either authorities or trustworthy institutions, broadcasting of notifications
CoroTrac [10]	GPS	Centralized, own database	Infrastructure maintained by the developer's institution
CovidWatch [11]	BLE	Decentralized, locally at the device	Public database maintained by the developer's institution, broadcasting of notifications
Pandoa [12]	BLE, GPS	Centralized, own database	Infrastructure maintained by the developers, send notifications for possible contacts
NextTrace [13]	BLE, GPS	N/A	Infrastructure maintained by the developers, send notifications for possible contacts
geoHealthApp [14]	GPS	Centralized, own database	Infrastructure maintained by the developers
CoronaTrace [15]	GPS	Centralized, own database	Infrastructure maintained by the developers, send notifications for possible contacts
TraceTogether [16]	BLE	Decentralized, locally at the device	Infrastructure maintained by the developers, send notifications for possible contacts
DP3T [9] part of PEPP-PT [17]	GPS, Cell phone triangulation , BLE	Decentralized, locally at the device	Run by either authorities or trustworthy institutions, broadcasting of notifications
NextStep [18] part of PEPP-PT [17]	BLE	Decentralized, locally at the device	Full details not disclosed, matching of IDs all done on device
NOVID20 [19]	BLE, GPS	Decentralized, locally at the device	Infrastructure maintained by the developers, send notifications for possible contacts
Luca-app [20]	GPS	Decentralized between host, guest, and health department	Encrypted data is stored in Germany by a provider certified according to ISO-27001
StopCorona [21]	Microphone, BLE	Centralized	Infrastructure maintained by the developers, notifications available publicly
Sismo [22]	GPS	Centralized, cloud-server	Infrastructure maintained by the developers institution

7. **User notification**: The key feedback channel back to the user needs to be identified, especially for any feedback the user may want to know about the data collected, e.g., graphs, statistics, summaries, or simple "encounter" information. Depending on the system and data, such data may be already privacy-protected, thus, encrypted.
8. **Storage**: The storage of data collected is partially important for comparisons, statistics, and trends. Thus, different approaches are deployed to store data in general, such as local storage, in the cloud, on private servers, or on individual devices only. Also, based on the storage approach, Backends (*cf*. next item listed below as "Backend") might be required to receive and actively forward information related to the medical status.
9. **Backend**: The Backend is important for *(a)* an exchange of data between devices, which are geographically not close (any more due to mobile users), *(b)* a possible "comparison" of data broadcast to the Backend, *(c)* a pure relaying of messages, or *(d)* a publication of static content. Depending on the specific role intended, one can derive how much power and information the provider of such a Backend holds. Example instances of these (not necessarily orthogonal) roles include servers with e.g., a central database of all medical states, a relay functionality for messages, a broadcasting function for messages, or a publishing activity of static content.

These current solutions available differ not only from technological perspectives but also in terms of privacy concerns. The solutions presented in [10], [12], [15], [20] rely on centralized

infrastructures that might result in privacy-concerns even with anonymization models in place. In a different direction, [9], [21] stand as pseudonymous solutions, since they collect information for research purposes. Others solutions as proposed in [18], [19] store personal information of users' contacts locally using symmetric encryption. This approach might result in an exposure of the user's identity and his/her contacts throughout the day.

Although many of the state-of-the-art solutions implement privacy-preserving mechanisms (e.g., data anonymization models and encryption), gaps still exist to achieve a fully anonymous model for contact tracing. WeTrace addresses these gaps by applying public-key cryptography mechanisms to achieve a decentralized and privacy-preserving model, thus, not storing any information about users and their contacts. Besides that, WeTrace is an open-source approach, which allows for continuous auditing and evolving according to the community demands.

III. TECHNICAL AND LEGAL REQUIREMENTS

Technical requirements of tracking and tracing application are explicitly extracted based on COVID-19 pandemic characteristics. They are outlined to determine *(a)* the minimal set of functionality needed and *(b)* major system boundaries and constraints of the WeTrace application, which are defined in terms of user privacy. The respective design following in later sections of the paper complies with such requirements. System boundaries and constraints are specially considered to comply with GDPR regulation [5] and the respective user data and user privacy.

Despite the main concern with the users' confidentiality and the legal aspects to which a mobile application is submitted, other requirements such as usability, scalability, and energy efficiency are also relevant to determine its success in terms of mass adoption. Thus, simplicity is the key to make intuitive user interfaces and to avoid unnecessary operations (e.g., intensive and explicit use of BLE or GPS). The user will have to receive the WeTrace application from a trusted platform, which means a minimal additional effort for users to launch and leave it running in the background. Ideally, the WeTrace protocol can be included into already existing and deployed apps so that the user does not need to install any additional application to ensure his/her privacy. However, if the WeTrace application is installed separately now, the installation experience can be straightforward for any user to get started quickly.

Besides essential requirements for such an application and its implemented system of being epidemiological sensible and useful, important soft requirements exist. Especially in cases where people utilize a tracing application voluntarily, the importance of those soft requirements becomes clear since volunteers installing such an application need to be ensured that such an application does comply with all of the following ones. Thus, the following list of requirements was identified as crucial:

- Privacy
- Scalability
- Energy consumption
- User overhead
- Legal compliance

A. Privacy Properties

Moreover, within the general context of privacy, the properties defined by [27] are taken into consideration for the design of the WeTrace application, which explicitly include the privacy from Snoopers, Contacts, and Authorities. These three properties determine three dedicated, and potential attack vectors since any of these three roles listed potentially could harm the system's coherent and trustworthy operation. Thus, they are evaluated in Section V.E.

B. Scalability

Within a pandemic setup, it is expected that data, i.e., of new positive cases, can grow exponentially in a short amount of time. WeTrace needs to be able to cope with this exponential growth to be useful when needed the most. Thus, *(a)* the number of infections, *(b)* the number of "close contacts", and *(c)* the number of keys determine relevant parameters impacting WeTrace's scalability. A fourth scalability dimension is determined in terms of regions covered. Without any doubt, achieving an application suitable also across multiple countries will be inherently more useful. A selected set of numerical examples related to the scalability of WeTrace is discussed in Section VII.

C. Energy Consumption

Even though the scanning and advertising of BLE packages have a minimal impact on a smartphone's battery life, compared to communication alternatives such as ZigBee/802.15.4 [28], it is evident that a user that allows the WeTrace application to run in the background will not perceive the application as acceptable, if it is draining the battery life of the device. In this sense, it is imperative to be compatible with battery optimization mechanisms of Android and iOS platforms. Hence, the WeTrace implementation has to consider the impact on battery life with the use of BLE for tracking close contact encounters [29], [30].

D. Legal Compliance

Legal compliance with data protection laws and regulations, e.g., the GDPR, is crucial for any technical solution that collects and analyzes user data [9]. Specifically, Article 25 of the EU GDPR states that *only personal data, which are necessary for each specific purpose of the processing, are processed. That obligation applies to the amount of personal data collected, the extent of their processing, the period of their storage, and their accessibility* [5]. Thus, any digital tracing solution must respect the legal requirements, minimizing the data collected and processed. This goal is reached by WeTrace since as, designed and documented below, the user decides on data to be shared besides the fact of a "close contact" in a fully anonymous manner.

According to Article 4(15) of the EU GDPR [5], *data concerning health* concerns any personal data related to the physical or mental health of a natural person, thus, applying to WeTrace. Since the WeTrace messages are encrypted with the public key of the Device B, WeTrace assures the confidentiality of all messages communicated (i.e., published) via the Backend (*cf*. Fig. 2). Hence, negligent parties (e.g., Device C), cannot decrypt data seen in this message received since these messages

are encrypted with the public key received via the "close contact" over a BLE advertisement packet.

IV. WETRACE STAKEHOLDERS AND ARCHITECTURE

Taking these requirements as of above into consideration, the WeTrace application addresses the problem of privacy in the tracking of COVID-19 cases. WeTrace, as a fully privacy-preserving solution, relies on reliable cryptographic mechanisms, such as public-key cryptography, to enable an application where the identities of users are only known by the user him/herself. Thus, details on stakeholders involved are presented in the context of the WeTrace architecture, which describes the flow of information between the major components and stakeholders.

A. Stakeholder Definitions

Even though WeTrace consists of a simple approach involving *Users*, *Devices*, and a *Backend*, there are other relevant and related stakeholders to be considered. These, including the main three ones, are described as follows:

- **Users** are individuals using the WeTrace application, in which any person can have at this state of the implementation three possible states: *(i)* "Not infected", *(ii)* "close contact with infected", or *(iii)* "infected". As discussed in Section II, actual states are to be determined by epidemiologists.
- **Medical doctor**: Currently, the WeTrace application relies on self-reporting to detect COVID-19 infections, which might lead to the spam of false positives. To address such a problem, a medical doctor does act as a testing person. However, this could potentially weaken the privacy of the approach if doctor-user relations may become public.
- **Governmental health agency**: Similarly to medical doctors, governmental health agencies could provide trusted data concerning the medical status of an individual. However, research on how to maintain privacy concerning eHealth data must be conducted, as pointed out by [31], [32]. Thus, the WeTrace application focuses at this stage of the implementation on those three medical states as determined above only.
- **Devices** determine the technical platform on which GPS and BLE-enabled communications happen and where WeTrace is installed on. A Device stores the following information: Master seed used to generate public-private key pairs, public keys of devices (which have WeTrace installed) encountered within 2 m of proximity and being in contact for longer than an epidemiological relevant time (e.g., 10 to 15 min), a timestamp, and the approximated geo-location of encounters.
- A **backend** broadcast messages of users who changed their status from "not infected" to "infected". The Backend does not store any data, only publishes them to other WeTrace-enabled devices, which perform the decryption of messages in case of needs.
- The **server administrator** of the Backend must be considered, since he/she will have access to messages originating from the WeTrace application. Although messages are encrypted with public keys and the server does not store private keys, the server administrator has knowledge of their existence and, consequently, access to related meta information, e.g., IP addresses and the device platform that could be used to infer with data analytics mechanisms further details in the context of the user's identity.
- The **identity provider** also has an important role in the privacy concerns. Even though the identity of the user is only known within the device (based on the storage of private keys generated via the master seed), such devices require outside the use of the WeTrace application the registration with *communication service providers*, e.g., telephony operators, 4G cells and their base stations, or *Internet service providers (ISP)* with their Wi-Fi networks. Thus, the identity of the device a user utilizes could be retrieved if such providers share this information for a meta-analysis.

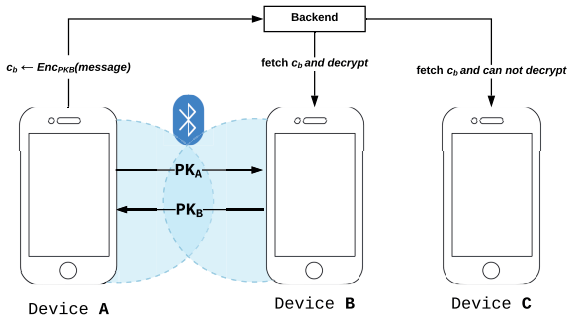


Fig. 1. WeTrace architecture.

B. Architecture

Fig. 1 depicts the architecture of WeTrace with three devices A to C. All three devices run the WeTrace application and are broadcasting their public key individually.

Whenever these devices see another device that is broadcasting its public key over BLE, they will store that public key received in their local storage. Furthermore, devices actively poll the latest encrypted messages from the Backend and try to decrypt them. This connection to the Backend happens through a general Internet connection since only the exchange of public keys happens through BLE. Finally, encrypted messages are published by the Devices via the Backend, which will then make these newly encrypted messages available for everyone else, thus, the other devices. The third-party hosting the Backend potentially will be able to collect IP addresses from those communications and devices, which are publishing encrypted messages. Even though the content of those messages cannot be read by a third party, the fact that they can be linked to IP addresses can potentially pose a threat to privacy. To avoid this from happening on that level, a possible mitigation routes the publishing of messages from a device through the Tor network.

C. Interaction and Sequence Diagram

The overall flow of information and interaction between the main stakeholders of WeTrace is shown in Fig. 2. On the one hand, Device A and Device B "see each other" for a long enough

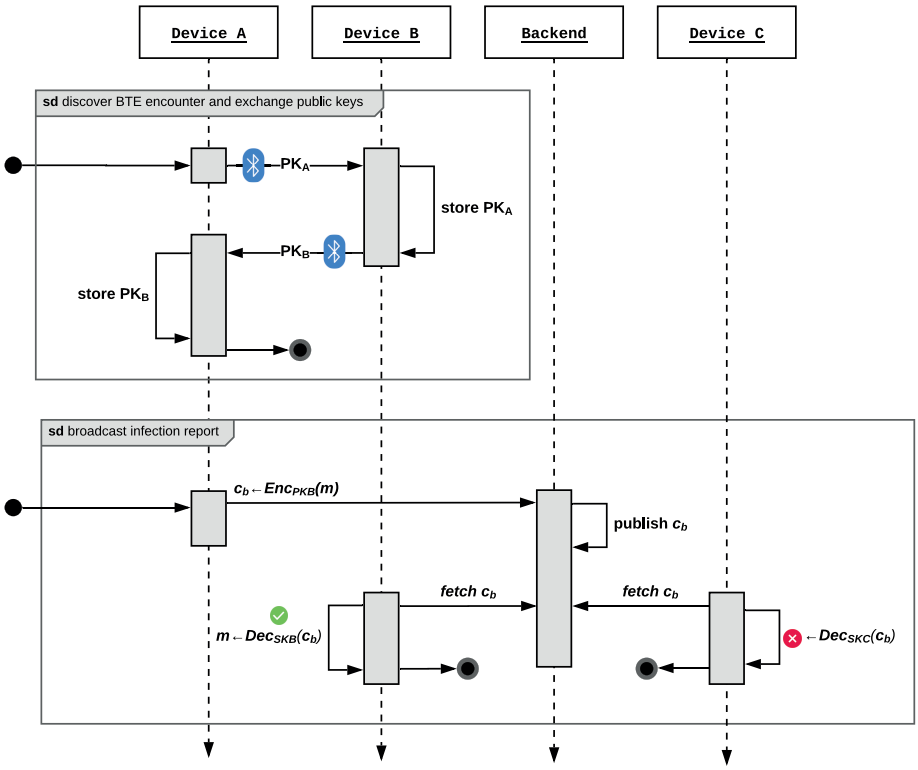


Fig. 2. WeTrace sequence diagrams.

time, such that they decide independently from each other to store the other party's public key in their local storage on the device. On the other hand, Device C is not in reach of those Devices A and B, hence Device C neither collects public keys nor advertises the own public key to anyone successfully. In case a fourth Device D may pop up in proximity to Device C, the public key advertised by Device C regularly will be stored by Device D and vice versa.

Under the assumption that Device A decides to report an infection, it will iterate locally through the public keys stored within the local storage and will encrypt the new message containing the information the user decides to broadcast with each public key individually, such as at least the infection status. Among those public keys locally stored, there will also be the public key advertised by Device B previously.

Since Device B will poll from the Backend regularly the latest messages, it will try to decrypt those ones received. Thus, Device B will then find out that one message can be decrypted with his/her own private key, meaning that the message was intended for this Device B only. At that point in time, the WeTrace application has to inform the user of Device B that the message was intended for this device and contains the decrypted information as communicated.

In the meantime, Device C will also poll messages from the Backend regularly. However, since his/her public key was never collected by Device A, none of these messages received can be decrypted. Hence, Device C will know that no message was intended for it.

V. TECHNICAL IMPLEMENTATION

WeTrace is proposed to be operational as an application on mobile devices, typically smartphones running iOS or Android. WeTrace deploys mechanisms and technology, which exists today and has proven to work in many instances. All stakeholders, as defined in Section IV.A above, are considered to be operational and in existence. The open-source WeTrace implementation is available at [3] and serves as the basis for aspects described in the following.

A. *WeTrace Properties*

Before WeTrace's technical assumptions are made explicit in Section V.B below and the respective security and attack models are outlined in Section V.D below, the major WeTrace application properties are defined as follows:

- Data of "close contacts" stay only on the mobile device. These are never shared with a Backend.
- Locally collected "close contacts" ($\leq$ 2 m) are stored together with the timestamp of when the contact had happened and approximate geolocation of where the contact had happened.
- In case of an "infection report" being received from a third party, e.g., a doctor or a hospital with which the owner of the mobile devices had medical interrogations, only those users, who had been in close contact will be notified (*cf.* Fig. 2). The Backend sees those messages being encrypted, cannot read its content, and operates with the sole purpose of relaying the message.

- The user, who received a report on an infection or a non-infection, can decide whether he/she wants to: *(a)* Report only the infection to the close contacts of the last 14 days. This value of a constant of the protocol will be configured within the application upon deployment and defines a medical consensus defined by epidemiology, *(b)* report the above *and* the timestamp of when that close contact had taken place, *(c)* report the above *and* the geolocation of where the close contact had taken place; or *(d)* report the above *and* the timestamp of when that close contact and the geolocation of where the close contact had taken place.

By utilizing a suitable combination of BLE protocol features — available in many mobile devices — and security mechanisms — to be made available via the WeTrace application — in addition, these concepts in conjunction define the underlying system's architecture.

B. Underlying Technical Assumptions

The device in use for the WeTrace application is assumed to be trusted since the secret key (used as the master seed to derive multiple public keys) is managed by the operating system (OS) (e.g., Android or iOS) and stored at the secure enclaves provided by the mobile OS. Furthermore, the general assumptions applying to asymmetric cryptography also apply to WeTrace, since public-key cryptosystems rely on computational hardness assumptions (e.g., factoring discrete logarithms or the decisional Diffie-Hellman problem [33]). Additionally, trust in the integrity of the GPS location and timestamp is also assumed since it is generated by the underlying OS, using respective sensors of the smartphone. The underlying communication protocol BLE and especially the BLE advertising packets [34] are deployed for exchanging close encounters are applied. Considering the WeTrace architecture depicted in Fig. 1, the following assumptions are listed:

- The **Device** is assumed to be trusted since secret keys must remain private, and, thus, any data generated and broadcast is trustworthy.
- The **Backend** is assumed not to be trusted with respect to confidentiality, integrity, and availability. Therefore, devices are required to encrypt any message, including notification messages, with the public key of devices that had been received in close proximity over a defined period of time. Therefore, an interaction scheme is applied, where public keys are exchanged regularly, while devices are near and reachable via BLE.
- The **communication channel** is provided by BLE's advertisements, which are assumed not to be trusted. Thus, a service provider cannot assure the availability of the service, i.e., messages can fail to be delivered and their content is readable from anyone receiving the message. If not, an explicitly encrypted payload is maintained. While this reflects the communication channel among BLE devices, the Device-to-Backend communication channel uses the Internet and is assumed not to be trusted as well. Similarly, the message content is readable from anyone receiving messages. If not, an explicitly encrypted payload is maintained. Furthermore, revealing the identity of that device running these communications may be possible with external metadata from the service provider, such as IP addresses or phone numbers. However, this is mitigated by applying the sending of messages to the backend via the tor network.

C. Use Case — WeTrace Application Operation

The WeTrace application's operation is exemplified within a given use case. Thus, for this example, User A with Device A, User B with Device B, and User C with Device C are considered. WeTrace is agnostic with regards to the required a public-key cryptosystem (e.g., RSA, ElGamal), which for an efficient and scalable operation is based on Elliptic Curve Cryptography [33].

1. Every device that installs the WeTrace app first generates a secret key SK_A. Based on this SK_A, the public key PK_A is generated second. For this examples PK_A stands for the public key of Device A and SK_A stands for the secret key of Device A. Thus, this step generated PK_A, PK_B, PK_C, and SK_A, SK_B, SK_C, respectively.
2. In turn, every device starts broadcasting its PK_*, which is also considered to be its unique identifier to its surrounding devices.
3. Upon devices (i.e., A and B) now getting into "close contact" with User A, the Device A receives PK_B and the Device B receives PK_A via the exchange of these via the BLE protocol. Besides these pubic keys PK_*, both devices also store a timestamp and the approximate geolocation of where this encounter had happened. Generally, this contact information is collected for all devices in close proximity.
4. Upon User A's information received, e.g., from a doctor or a hospital with which the owner of the device had medical interrogations with, User A wants to report an "infection" or a "non-infected" status. Thus, Device A will go through the list of close contacts collected within step 3 and encrypts one message each for every public key of close contacts. In case of User A, a notification will be encrypted once with PK_B, because in this example, B was the only contact. The messages will be transmitted to the Backend over the Internet, from where the Backend relays the received messages to devices. The messages will contain the data that User A chose to share, so either only the fact that an infection happened or not, or additionally when or even where it happened. As noted above, only the reporting user must decide if he/she wants to share this additional information.
5. Device B and Device C are now notified by the Backend. The notification informs them that new reports have been seen. Thus, Device B will then attempt to decrypt every message already recorded at step 3, the close encounters, with SK_B and will eventually find out that a message was directed at him/her, thus, indicating to User B the status User A now knows and was reported about. Device C will do perform exactly the same steps. However, no message in his/her local storage can be decrypted since it was not encrypted using any known private key (in this case PK_C). Thus, no information is relayed to User C, which can now delete the message.

While this example indicates clearly how the use of public-key cryptography mechanisms ensures the generation of an un-

Table 3. CIA analysis of the backend and devices.

CIA	Backend	Devices
Confidentiality	Does not provide guarantees	Notifications are encrypted with PK of previously in range devices
Integrity	Does not provide guarantees on the delivery of notifications	Does not provide integrity guarantees of notifications
Availability	Backend can fail or be a target of DDoS attacks	Not ensured that neither the sending nor receiving devices are available.

recorded — thus, no individual identity being assigned to a public-private key pair — identity, which is temporary for the lifetime of the application. A new installation will generate a new public-private key pair, while additionally, the device generates a new public-private key pair every 15 min. The example shows as well that the Backend does not maintain the content nor any data, even if it would do so, run as a possible attack, the content will remain encrypted and unusable, since no proximity information would be available to that server.

D. Security and Attack Model

The WeTrace design does consider a broader spectrum of potential vulnerabilities and includes respective countermeasures. Therefore, to provide the technical means necessary to ensure user privacy and user data privacy, it is mandatory *(a)* to ensure that at least one Backend is available to broadcast notifications to others and *(b)* to ensure the integrity of these notifications sent by Devices with a minimal Backend interaction.

Furthermore, the WeTrace Backend (*cf.* Fig. 1) is assumed not to be trusted as such. Therefore, with respect to confidentiality, integrity, and availability the Table 3 discusses those characteristics for WeTrace Devices and the Backend.

D.1 Adversary Model

WeTrace assumes an adversary model with malicious users, which can potentially cause a denial-of-service (DoS) or, in the case of multiple malicious users, a distributed DoS (DDoS) attack. Also, no trust guarantees on the Backend concerning confidentiality, integrity, and availability (CIA) (*cf.* Table 3) exist.

D.2 DDoS on the Backend

A device may intentionally or unintentionally cause a DoS on the server by sending multiple requests to notify a list of devices previously in range. Additionally, a DDoS can happen with multiple devices performing the same operation. Furthermore, the same concern should be considered to prevent one or multiple servers from flooding one or multiple devices. Therefore, the notification schema should be carefully designed to prevent the flooding of messages on both sides (backend/application).

Furthermore, such an attack scenario is mitigated by a combination of two mechanisms: *(i)* By adding an anonymous authentication when publishing data to the Backend, such as via dedicated token only hospitals may use or by applying a proof-of-work activity and *(ii)* by limiting the size of the message a single device can report to a sensible maximum value, such as that a device can only contact 1,000 users at most.

D.3 Impersonification of the Backend

Impersonification attacks might take advantage of a trusted relationship between different systems, in which an attacker can send packets, messages, and information pretending to be from a trusted source. Thus, this kind of attack is hazardous since it can result in access to users' data or even control entire systems.

In the case of WeTrace, malicious users may try to impersonate the server in order to intercept the connection of multiple devices to the Backend. Thus, it is essential to use a secure communication channel (e.g., TLS/HTTPS or end-to-end encryption) with the Backend to prevent malicious devices from acting as a man-in-the-middle.

D.4 False Notification Reporting

It is also important to prevent users from issuing false notifications, either maliciously or unintentionally. Thus, it is necessary to ensure that the application issues a "confirmation page/button" before changing status and also (and not less critical) prevent multiple changes of status over a period of time. Although it had not yet been confirmed so far, there are only few cases of multiple COVID-19 infections on the same individual.

Since there does not seem to exist a reason to allow a device to change its status to "infected" or "not healthy" twice or often, it is highly relevant to foresee such a warning/confirmation page/button. Furthermore, the Backend should prevent the situation where multiple individual notifications are sent to a recipient device in multiple copies. Therefore, the mitigation measure will follow the basics in terms of DDoS mitigation as described above: anonymous authentication when publishing data to the Backend prevents such falsified notification reports.

E. Privacy Enforcement

The enforcement of those privacy properties as set-up in Section III.A is key for approaches that involve the tracing of people. Therefore, the WeTrace approach discusses and addresses that as follows.

E.1 Privacy from Snoopers

Since WeTrace will broadcast a signal on "close contacts" based on public keys generated (*cf.* Fig. 2) so that others in close proximity can detect and possibly register such a contact anonymously, snoopers will also be able to see public keys being advertised. However, since those public key-based identities will be valid for a limited time only (which can be further shortened if needed), the user will not be more exposed to snooping than he/she already is with a Wi-Fi-enabled device that is broadcasting its medium access control (MAC) address without any consent or knowledge of the user. This is still happening in case MAC randomization is enabled.

E.2 Privacy from Contacts

Privacy from contacts is addressed in a similar argument as in the case of "Privacy from Snoopers". Close contacts will receive a notification if a user chooses to broadcast his/her infection status. However, close contacts without exceptions will *not* know from whom this message originated. A single case where this could be inferred would be when a user *only* is in range

with another *single* contact during the last 14 days, and that device would broadcast this information. While this is clearly not impossible to happen, its likelihood is small. Thus, WeTrace covers in the large majority of realistic cases this property.

E.3 Privacy from Authorities

Due to the fact that only encrypted messages are sent to the Backend (*cf.* Section IV.A), it does not have access to any personal user data from any of these messages relayed. Thus, communications from or to the Backend do not reveal, e.g., the number of infections or the identifiers of any recipient. The Backend only knows that someone wants to inform about medical status, hence the existence of related communications only reveals that "at least one medical status update had happened".

Additionally, if WeTrace would introduce random messages, any third party, such as an attacker or any authority, will not even know about such a medical status change. Therefore, this property is achieved by WeTrace as well.

F. WeTrace Attack Vectors and Other Mitigations

The basic WeTrace Architecture and its stakeholders' involvement potentially give rise to concerns, which had been identified as weaknesses or attack vectors, but are also addressed directly with suitable mitigation means. While two security-related aspects include possible attacks against the privacy requirement (a malicious scanning of advertisements sent of arbitrary devices and active message injections in combination with eavesdropping), the respective countermeasures are introduced below. A BLE-related protocol concern exists with respect to its limitations of broadcast messages, such that at worst public keys could not be broadcast, thus, a proper operation is defined to circumvent this problem. Moreover, finally, performance concerns may raise with respect to the scalability of message decryptions, especially in the case of many close encounters.

F.1 Malicious Scanning of Advertisements

The remaining privacy concern is due to the fact that a malicious user or attacker could start tracking a users' location by scanning his/her advertising packets. This can potentially happen if the attacker is in the proximity of the user under attack. However, this case does not occur in reality, since within step 1 as above, the device's generation of a key pair is, in fact, the generation of a so-called "master seed". This master seed is used to derive, in turn, an unlimited number of key pairs deterministically. This is designed so that the user will be changing the key in a specified period of time (e.g., every 30 min), making him/her traceable with that public key for that time frame only. Thus, the local knowledge of this device's validity period and the respective applied key pair's storage remain at the discretion of the user's device only and is fully decentralized. This leads to the situation that even a maliciously collecting Backend would possibly collect "different" public keys, which cannot be mapped onto a single device by any means.

The major advantage of this approach is — besides its elegance of hiding temporarily identities even further efficiently — that the user still only stores one master seed and derives upon the reception of a notification from the Backend all key pairs used during the past 14 days. Using those derived key pairs, the device tries to decrypt the message by iterating over those keys. This does only require the storage of master seed only since the dynamic derivation of all keys generated is time-wise not costly but saves valuable storage. However, the implementation of the alternative, storing all keys generated over a 14 days period, can improve the decryption time at the cost of a higher local key storage size.

F.2 Message Injections and Eavesdropping

In the approach developed, the risk of an attacker injecting packages and messages instead of eavesdropping exists. Thus the risk evaluation deals with the question, whether that is better or worse for a snooper.

On the one hand, selected tracing applications and proposals are today still prone to eavesdropping i.e., pan-European privacy-preserving proximity tracing (PEPP-PT). Eavesdropping refers to the fact that an attacker passively listens to all communications going on, which requires in case of BLE communications to be in close proximity or to install maliciously a BLE-based Backend, which collects all local communications to forward it to the attacker's infrastructure for analysis — none can be prevented from happening. On the other hand, tracing applications and proposals are today still prone to message injections i.e., WeTrace here. The major commonalities and differences are summarized as follows:

- All unauthorized listening and eavesdropping are undetectable in the general case.
- Message injection is, in general, difficult or even impractical if these messages are "directed at someone" explicitly.

Thus, only an approach, which can prevent message injection while being prone against eavesdropping, can survive a security analysis. For the WeTrace approach, that means: In order to know, if User A were infected, the attacker would need to ensure that User A *only* receives the message injected since if multiple other users would also receive that message, the attacker would not be able to distinguish anymore from whom the message was received.

Furthermore, if the attacker relies on the fact that the "reporting" party needs to record his details, that party has control over "when" to record the attacker as a "close contact" — that party can define, e.g., how high the signal level needs to be or how long the contact needs to last.

Finally, for the eavesdropper's scenario, the attacker would collect as much as possible, and the other party has no control over what is being recorded or not. However, these data collected are of no use for the attacker since the temporary identities frequently change over time and cannot be associated under any measure reliably with a device, thus, a user.

F.3 Limitations of BLE Broadcast

The BLE protocol is limited with respect to how many bytes can be broadcast while in background operation. Thus, the major concern is, does public key fit into a BLE broadcast?

WeTrace requires, for secure encryption, at least 24 Byte for the public key. Ideally, it would above 32 Byte. While the BLE advertisement message is limited with respect to the number of bytes being included as a payload, solutions exist to work

around this limitation, even if the payload would be limited to only 16 Byte. The following options exist:

- Do not broadcast the entire public key. In this option, only the first *n* bytes (*n* being the number of bytes to be included into the payload) are broadcast. Afterward, the remaining 32-*n* bytes are published to a server, for instance, as a map that uses as the lookup key the hash of the first *n* bytes. This will allow for the operation with an infinite size of keys since only those devices that had been able to collect these *n* bytes will be able to request the remaining bytes. Thus, the authority running the server will not know by any chance the full public key, but only 32-*n* bytes.
- Use multiple advertisement packets with an *n* bytes payload. Since a possible contact counts as a "close contact" only after a certain amount of time, it is clear that a device has to collect at least two separate advertisement packets to be able to define that duration. Thus, it is viable for WeTrace to split up the key into multiple advertisement packets.
- Advertise only a fixed service universally unique identifier (UUID). A UUID allows others, users and devices, to request the characteristics of that service. In this case, the characteristics are determined by that 32 Byte.

F.4 Scalability of Message Decryption

The WeTrace approach and protocol outlined require a user to decrypt all messages to understand (i.e., interpreting the content of an initially encrypted message correctly) if a message was directed at him/her. While this scales reasonably well as long as the user has to decrypt up to 1 million messages, it might become a problem with larger numbers, mainly because the drainage of the users' device battery with decryption tasks needs to be avoided. However, under the assumption of COVID-19 curfews as well as lockdowns, the likelihood that mobile devices are recharged more often is large due to having fixed power supplies.

A suitable path to mitigate this performance aspect without any considerations of more frequent recharging options is by prefixing the message with the first *n* bytes of the hash of the public key. This allows the user to select and reduce drastically the number of messages he/she needs to decrypt. Such an approach will basically allow a device to cope with almost any reasonable number of encrypted messages in those cases of hundreds, even thousands of close encounters.

VI. OBSERVATIONS AND DISCUSSION

The WeTrace approach and application — as it had been designed — suits its needs. Key details of these are discussed and evaluated from the perspective of advantages over other related work and drawbacks compared to related work — including their mitigation means. While major conclusions are drawn, it is essential to observe which trade-offs have been taken into account and how the next steps for tracking and tracing applications in future pandemics are foreseen.

A. Trade-offs

There are many trade-offs to consider. In the theoretical view of the design that WeTrace follows, it trades off *(a)* a central analysis versus *(b)* the privacy of users and user data. While it is evident that a central analysis of data can be advantageous, once authorities, for instance, want *(i)* to detect "hot spots" of infections or *(ii)* to perform page ranks on possible subsequent infections, such data being processed will have to be stored either centrally or locally, while for the latter access to authorities have to be guaranteed. Thus, authorities in the case of *(a)* will know more about the participating users' behavior than necessary.

In essence, this additional information is for the successful analysis, prediction, and action plan development of COVID-19 cases not needed, since the measure of "proximity" is based on the evaluation of epidemiological requirements fully sufficient. As noted above, on an individual basis and designed as an opt-in approach, individuals may add location information and time. In the case of large cities, this is unlikely to impact the user's privacy; in rural locations, where only a few dozen inhabitants reside, such decisions may be considered to be more critical with respect to the privacy aspect. However, it was clearly stated to be an option; thus, freely available data does not violate privacy regulations.

B. Proximity Discussion

Furthermore, a key requirement in WeTrace is that the application has to see both devices needing to record one another, the proximity. This is only and solely based on the use of public-private cryptography, for which such key pairs may be generated on the spot since there is no need to register these key pairs at a Certification Authority due to the fact that not the individual's identity is the key, but the fact that two individuals exchange the proximity information at first and may exchange later infection status without revealing any identity for that second step, only the public key once collected at the close encounter. Thus, this approach enables WeTrace to encrypt the message for a possible receiver of the encrypted report and does fulfill the trade-offs as of *(b)*. A symmetric encryption scheme will not work since the full independence of any centralized authorities establishes an exchange model of status information on an ad-hoc basis without any centralized control.

The proximity requirement is met because WeTrace defines a close contact as an individual being in a distance of 2 m of proximity to a COVID-19 infected person and for an epistemological relevant period of 10 to 15 min. These parameters are in accordance with several standards defined by major health organizations worldwide. For example, the US centers for disease control and prevention (CDC) defined "close contact" as "*...being within approximately 6 feet (2 meters) of a COVID-19 case for a prolonged period of time*" [35], the European CDC defines as "*...having had face-to-face contact with a COVID-19 case within 2 meters and more than 15 minutes*" [36], the New South Wales Ministry of Health defines as "*...greater than 15 minutes face-to-face contact in any setting with a confirmed case in the period extending from 24 hours before the onset of symptoms in the confirmed case...*" [37], and the Brazilian Ministry

of Health defines as "*A person who has had face-to-face contact for 15 minutes or more and at a distance of fewer than 2 meters (m)*" [38]. Thus, WeTrace's major parameters are aligned with major guidelines of the "close contact" definition worldwide.

C. Privacy Discussion

Concerning the privacy requirements listed in Section III.A, the WeTrace approach and application addresses the imposed privacy requirements and challenges highlighted in [27]. Hence, the approach tackles the following privacy aspects:

- **Privacy from snoopers**: WeTrace addresses this challenge by limiting the time-wise validity of public and private key pairs, generating a new one every X min. A *snooper* is not aware of the device's exact location nor its identification. The snooper is only aware of the notification that close contact with an unknown individual took place.
- **Privacy from contacts**: This is tackled by encrypting the message with the public key from "close contact" devices. Thus, an individual will know that "infection" messages were sent to him/her, but not who sent them, since the Backend and the application do not store any private information.
- **Privacy from authorities**: Similarly, with the employment of public-key encryption, messages exchanged between the Backend and devices are encrypted. They are only decrypted with the knowledge of the private key, which remains solely in the user's device. Thus, authorities cannot have access to the messages' content. Therefore, although access to encrypted data as such for server administrators is possible, the "lacking knowledge" of private keying material will not reveal without significant efforts (e.g., data mining and data analytics mechanisms on metadata surrounding an encrypted message's reception and storage) any critical user-related details, thus, leading to a low-risk situation.
- **Infrastructure requirements**: WeTrace requires a single Backend with a simple message broadcasting application. Even though logically a single server is required only, multiple instances of such a Backend can exist to increase their availability and performance without negative implications.

D. Usability and Integration Discussion

The WeTrace application resides as of today in a separate prototypical implementation. Thus, the question of how to integrate this important privacy-preserving functionality into tracking and tracing apps, which focus on those layers, needs to be answered.

For example, the world health organization's (WHO) app as an e-Library of evidence for nutrition actions (eLENA) [39] could place one example for such an integration. The WHO Zika App [40] as of Google Play can serve as a second one once it is turned into a COVID-19 app. On the one hand, any integration would need to consider technical constraints carefully. On the other hand, only applications that do not require user credentials are suitable since otherwise privacy may suffer and be at risk. Thus, the WeTrace application will have to be offered as an software development kit (SDK)), which makes it easier for any other application to integrate WeTrace.

E. Major Observations

Overall, the trade-offs and discussions highlighted indicate the key aspects of a system in which many individual participants act as in one role (inhabitants) and only few acting in the second role (authority). Thus, the WeTrace design decision taken does enable the two roles to act as they are required to act independently. However, based on each other, the method implemented shows properties in which the privacy requirement of the proposed solution is integrated elegantly and easy to deploy.

Of course, the important next step will be that the "community" of major players and stakeholders can agree on a "standard" on how to trace infections in case of COVID-19 and then ensure that developers will use the same standard or protocol such that the system can profit from a network effect across different applications, regions, and even countries. It is imperative that this initiative, to which WeTrace, as well as many other applications, need to be counted, is following an open-source philosophy so that *(i)* security-related measures can be verified openly, *(ii)* functionality verification can be performed at no risk, *(iii)* various application developers can cooperate, and *(iv)* stakeholders involved can collect those data, which are essential and securely collectible.

VII. EVALUATIONS AND FINDINGS

The investigation of related work in tracking and tracing applications on the case of COVID-19 did reveal that the problem is not the collection of data as such, typically provided by accessing mobile devices such as smartphones, which are in possession of an individual, but the guarantee that those data collected are fully maintaining the basis and the relevant details of a privacy-protected approach. Thus, the human individual and his/her privacy, his/her private data, and fully anonymous processing of related data is the key to meet European and many other countries' demands, while at the same time being compliant especially with the European regulation, especially the GDPR.

A. Communication Channel Evaluation

The WeTrace approach utilizes BLE communications, which many modern mobile devices provide today. This coincides with low range requirements of the medical dimension since infections potentially can only happen in case of close proximity, where humans need to stay below a 2 m distance for approximately 10 to 15 min.

The pure knowledge of such proximity determines the essential information for epidemiologists since based on density-related information, not requiring the exact geographical location, but a region only, prediction models of spreading rates or relaxing measures can be derived. However, since proximity and location determine a highly valuable good for every single person and individual on earth, it needs to be fully protected from possible misuse or unintended use. Just imagine the value of a human's geographical position for marketing, commercial services, or monitoring? This threat for an open society has to be balanced with the medical and health threats COVID-19 imposes on society. WeTrace allows for both to be reached and maintained at a highly secured level of operation.

Furthermore, the WeTrace application requires between 24 and 32 Byte to be transmitted via the BLE communication channel. Unfortunately, that is technically limited in iOS-based devices since two advertisement packets are required here. However, firstly, this "loss" of a single packet approach is not crucial to the game since WeTrace requires the reception of two advertisement packages always to measure the time a possible "close contact" had taken. Thus, the potential limitation to technically one message only does not harm at all. Secondly, the robust privacy-preserving approach adds dedicated time to the approach processing since a human associated with a smartphone can consider himself/herself "infected" or "uninfected" only once relevant data had been decrypted, which might causes in the general case a higher compute burden. However, this drawback can be mitigated already by adding the first few bits of the relevant public key into the message being communicated, such that only those messages need to be decrypted, which provides a partial match to the owner's public key.

B. Privacy and Attack Evaluation

The application of the well-known asymmetric cryptography only allows for deciphering a message at that destination. It had been intended for since that human operating that mobile device may remember his/her private part of the keys. Moreover, proximity-related messages are sent in an encrypted manner over BT in a low range setting. Since additionally, literally every other potential participant only listens to random data, even a brute force attack will not succeed to decrypt messages reliably on the fly. Therefore, WeTrace is the only known approach so far, which ensures that any receiver of a message knows that this is for him/her but does not know who the original sender was.

Furthermore, the users deploying WeTrace are offered an opt-in path to decide whether they want to add to the proximity message additional information, such as the exact location (not only a region) and the time. Thus, the application operates in an open-source manner only on the fundamental and privacy-protected data needed to crowd-source data to help COVID-19 countermeasures based on currently measured details.

While this is considered to be a clear advantage, even further relevant attacks are mitigated. A passive collection of communications in such a certain physical near range will not provide any reasonable amount of information, which could be used to reveal the sender's identity. Although, as outlined above, potentially the injection of public keys is possible in any setting, it can only happen if and only if the attacker is "local" for a certain amount of time. Thus, the WeTrace approach developed does not suffer from this attack since the application does configure and decide on the received signal strength indicator (RSSI) and the time. Therefore, eavesdropping does not show any negative impacts.

The WeTrace application addresses challenges that have been highlighted in [27]. This means that the WeTrace application complies with the stated privacy requirements, especially with the respective general demand to its key detailed requirements.

C. Device Storage Evaluation

Since the Backend just stores encrypted messages of the last 14 days, it offers these to whoever requests them. This is feasible and can scale quickly with commercially available off-the-shelf storage products. However, space can be considered to be a constraint for the personal devices of selected users. Therefore, it is important to shed light on the storage requirements when using the WeTrace approach in mobile devices.

Concerning especially the data stored on the client-side (i.e., the smartphone), WeTrace safely assumes individual encounters with 5,000 other smartphones in 14 days as close contacts (i.e., $\leq$ 2 m every 4 min) and rotating to a new public key every 15 min. Furthermore, assuming a 4 Byte size for longitude, 4 Byte for latitude, and a 4 Byte length for the timestamp, an encounter message will have the size of 12 Byte. Considering that 4 new keys are generated every hour for 24 hours and 14 days, an individual will own 1,344 private keys generated and 5,000 encounter messages stored. Thus, the device will need to decrypt 6,720,000 messages (i.e., approximately equaling 80 MB of data considering the 12 Byte-sized message). For today's devices, this is not considered to be a large number, due to an average of 80 GB of storage capacity per smartphone [41]. Also, since WeTrace uses a sliding window of 14 days (i.e., it deletes those keys collected after 14 days of the encounter; so it does not store keys forever), there exist no scalability concerns regarding the amount of encounters and private keys to be handled by one device.

It should be noted that this scenario presented is already considered to be an "extreme example" and it is highly unlikely to happen in daily operations. Therefore, it is relevant to evaluate such simulated extreme conditions of use of the application and edge cases in order to explore common breaches. Henceforth, neither the server nor the client's side storage is a critical aspect of the WeTrace approach.

D. Energy Consumption Evaluation

Energy consumption is a critical aspect of mobile devices. Therefore, it is one of the key aspects of COVID-19 tracing apps. As the energy consumption depends on the device and users' usage characteristics [42], it is not a trivial task to precisely determine always energy consumption. However, BLE was specifically designed for lower power consumption, allowing *beacon* devices advertising their presence every 100 ms using a 1,000 mAh battery to last for up to 4.5 months [43]. Thus, it can be seen that BLE does present a viable and energy-efficient solution to be used in contact tracing apps.

In the context of data encryption, the energy consumption of a mobile device depends on the battery's capacity, cryptography algorithms, and data size. In this sense, the energy consumption is massive to encrypt large files, e.g., files larger than 1 GB, might require more than 15% of battery available in a popular smartphone with 1,305 mAh of battery [44]. It should also be noted that public-key algorithms are more energy-intensive to perform the encryption than running the decryption [45]. Thus, in the case of WeTrace, for the "extreme scenario" (*cf.* Section VII.C), where the user device has to decrypt 6,720,000 messages (80 MB of data in total), it would require less than 2% of

battery on the same device. However, since WeTrace only sends one message to the server per private key owned, this is also not critical in terms of the potential energy consumption for the approach prototyped.

Thus, although experiments with real-users were not conducted explicitly, based on WeTrace's energy consumption assumptions and the literature available, this work here is based on acceptable evidence that WeTrace can run in popular mobile devices together with other applications without a significant impact on the energy consumption of this device.

E. Scalability Evaluation

The overall scalability of WeTrace depends on set of factors: *(a)* The number of infections, *(b)* the number of close contacts, and *(c)* the number of keys. Thus, if these numbers grow also the product grows. Currently, a smartphone is able to decrypt approximately 1 million messages within seconds, which is acceptable. However, if these numbers grow on the scale of billions, the scalability has to be mitigated: Every message is prefixed with n bits of the public key. By doing this, the device will only try to decrypt those messages, which match with the first n bits of their public key. This straightforward and easy to implement scaling strategy allows for an exponentially cut down of the number of messages to be decrypted. i.e., if a 1-bit prefix is assumed, the reduction of decryption sets is at 50%, with a 2-bit prefix 75% are achieved. However, the prefix should remain at an overall size, where the number of bits being disclosed does not reveal about the actual public key.

VIII. SUMMARY AND CONCLUSIONS

The protection of people and society against harm and health threats involves a variety of different disciplines. While in case of the COVID-19 pandemic, the virus and its medical treatment – from currently affected patients to the vaccination of future people – do see a major focus of research and work, the data collection of basic and health-related data of individuals in today's highly mobile society does help to plan, protect, and identify next steps health authorities and governments can, shall, or need to plan for or even implement. Thus, every individual, every human, and every inhabitant of the world is the key player – different to many past crises'.

Although the involvement of all humans cannot be considered to be negative as such, the individual's *(a)* health and *(b)* privacy shall be considered in a carefully crafted balance, not overruling one with another or prioritizing one aspect. If the solution of the current pandemic's data collection can be based on a fully privacy-preserving application, which can be used by individuals on their mobile devices (e.g., smartphones) while maintaining at the same time their privacy and while respective data collected in such a fully distributed setting does help to confine the pandemic, an important step forward can be achieved in a democratic and open, but still and especially privacy-protecting world.

Thus, the WeTrace approach utilizes the BLE communication channel, which many modern mobile devices provide in a way where asymmetric cryptography being applied only allows for the deciphering of a message for that destination it had been intended for. Since literally every other potential participant only listens to random data, even a brute force attack will not succeed. WeTrace is the only known approach so far, which ensures that any receiver of a message knows that this is for him/her but does not know who the original sender was.

Besides this clear advantage, even a passive collection of communications in a certain physical range will not provide any reasonable amount of information, which could be used to reveal the sender's identity. Although potentially, the injection of public keys may be possible, if and only if the attacker is "local" for a certain amount of time, the approach developed does not suffer from this attack, since it alone does configure and decide on the RSSI and the time. Therefore, eavesdropping does not show any negative impacts.

Finally, a slight drawback of this strong privacy-preserving approach is only the overhead to determine if a human associated with a smartphone can consider himself/herself "infected", since all relevant data needs to be decrypted. However, this can be mitigated already by adding the first few bits of the relevant public key into a message communicated, such that only those messages need to be decrypted, which provides a match to the owner's public key.

In conclusion, the WeTrace application provided in close relation to those requirements being defined and evaluated a highly suitable system based on the BLE communication channel in support of crowd-sourcing for COVID-19-relevant data in a privacy-protecting setting. This approach is scalable as well since close proximity of humans can be considered in the range of a few hundreds of people, not thousands anymore, since these are legally forbidden. Therefore, in case a mobile device would see way too many messages, a possible alarm can be raised, which by itself already identifies that a violation of meeting regulations had occurred.

In the same line of arguments, the resource consumption of mobile devices is not at stake, since especially data to be stored is limited to the public keys of those messages received. While the overhead on the compute side had already been mentioned, and it is considered to be at the lower end of the spectrum, the legal compliance with especially privacy considerations of users and humans have been met in full.

Note that full-fledged performance evaluation of this approach and the WeTrace application has not been performed with real-users but the privacy-preserving relevant aspects are discussed in detail and qualitative evaluations conducted to provide clear evidence of the WeTrace approach. The evaluations also considers hypothetical information and scenarios all based on real applications of WeTrace. Besides, the open-source implementation is available at [3] and related soft-requirements' suitability of thresholds not reached in practice have been discussed.

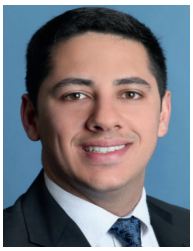
REFERENCES

[1] J. Rocklöv and H. Sjödin, "High population densities catalyse the spread of COVID-19," *J. Travel Medicine*, vol. 27, no. 3, p. taaa038, April 2020.

[2] A. D. Carli, M. Franco, A. Gassmann, C. Killer, B. Rodrigues, E. Scheid, D. Schoenbaechler, and B. Stiller, "WeTrace – A privacy-preserving mobile COVID-19 tracing approach and application," April 2020. [Online]. Available: https://arxiv.org/abs/2004.08812

[3] WeTrace Team, "WeTrace github page," 2020. [Online]. Available: https://github.com/WeTrace-ch/WeTrace

[4] Y. Sweeney, "Tracking the debate on COVID-19 surveillance tools," *Nature machine intelligence*, vol. 2, no. 6, pp. 301–304, June 2020.
[5] Regulation (EU) 2016/679 of the European parliament and of the council of 27 February 2016, "General Data Protection Regulation," 2018. [Online]. Available: https://gdpr-info.eu/
[6] GDPRhub.eu, "Projects using personal data to combat SARS-CoV-2," 2020. [Online]. Available: https://bit.ly/37ntO38
[7] S. Munir, D. H. Kim, A. K. Bairagi, and C. S. Hong, "When CVaR meets with Bluetooth PAN: A physical distancing system for COVID-19 proactive safety," *IEEE Sensors J.*, vol. 21, no. 12, pp. 1–12, Mar. 2021.
[8] Federal office for information technology, systems and telecommunication FOITT, federal institutes of technology in Zurich (ETH) and Lausanne (EPFL) and the Swiss company ubique, "SwissCovid app: Protect yourself and others," 2020. [Online]. Available: https://foph-coronavirus.ch/swisscovid-app/
[9] DP3T, "Decentralized privacy-preserving proximity tracing," 2020. [Online]. Available: https://github.com/DP-3T/
[10] CoroTrac, "CoroTrac," 2020. [Online]. Available: https://corotrac.com/
[11] I. Becker-Mayer, D. Blank, J. Colligan, R. Fenwick, M. Hittle, M. Ingle, O. Nash, V. Nguyen, J. Petrie, J. Schwaber, Z. Szabo, A. Veeraghanta, M. Voloshin, S. V. Arx, T. White, and H. Xue, "Covid watch," 2020. [Online]. Available: https://covid-watch.org/
[12] Pandoa Team, "Pandoa - Corona virus tracker," 2020. [Online]. Available: https://pandoa.org/pandoa
[13] T. Bedford, A. Black, J. Freeman, C. Megill, D. George, L. Meyers, A. Phelan, C. Rivers, and C. Viboud "NextTrace - Controlling COVID-19," 2020. [Online]. Available: https://nexttrace.org/
[14] M. Gleser, I. Bölükbas, and R. Sachartschenko, "geoHealthApp - Spread the app not the virus," 2020. [Online]. Available: https://en.geohealthapp.de/
[15] Cision, "CoronaTrace - Amber Alerts for Coronavirus," 2020. [Online]. Available: https://www.coronatrace.org/
[16] TraceTogether Team, "TraceTogether, Safer Together," 2020. [Online]. Available:k https://www.tracetogether.gov.sg
[17] PEPP-PT, "Pan-European Privacy-Preserving Proximity Tracing," 2020. [Online]. Available: https://www.pepp-pt.org/
[18] Ubique Innovation AG, "Next Step - Get Back Together," 2020. [Online]. Available: https://next-step.io/de/
[19] C. Tockner, H. Trautsch, M. Kowatschew, and A. Petersson, "NOVID20 - The Private Way of Tracing Contacts," 2020. [Online]. Available: https://www.novid20.org/en
[20] neXenio GmbH, "Luca-app," 2021. [Online]. Available: https://www.luca-app.de/
[21] Austrian Red Cross, "STOP CORONA - MY CONTACT DIARY," 2020. [Online]. Available: https://participate.roteskreuz.at/stopp-corona/
[22] F. Finazzi, "Earthquake network - Pilot investigation COVID-19 in Val Seriana," 2020. [Online]. Available: https://sismo.app/covid/
[23] Many Contributors, "Unified research on privacy-preserving contact tracing and exposure notification for COVID-19," 2020. [Online]. Available: https://docs.google.com/document/d/16Kh4_Q_tmyRh0-v452wiul9oQAiTRj8AdZ5vcOJum9Yedit?ts=5e801c37#
[24] John Hopkins University, USA, "COVID-19 dashboard," 2020. [Online]. Available: https://bit.ly/3pugsIG
[25] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of COVID-19 contact tracing apps," *IEEE Access*, vol. 8, 2020.
[26] T. Martin, G. Karopoulos, J. L. Hernández-Ramos, G. Kambourakis, and I. N. Fovino, "Demystifying COVID-19 digital contact tracing: A survey on frameworks and mobile apps," *Wireless Commun. Mobile Comput.*, vol. 2020, July 2020.
[27] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs," 2020. [Online]. Available: https://arxiv.org/abs/2003.11511
[28] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is Bluetooth low energy? comparative measurements with ZigBee/802.15.4," in *Proc. IEEE WCNC Workshops*, 2012.
[29] J. Araujo, R. Matos, V. Conceição, G. Alves, and P. Maciel, "Impact of capacity and discharging rate on battery life time: A stochastic model to support mobile device autonomy planning," *Pervasive Mobile Comput.*, vol. 39, pp. 180–194, Aug. 2017.
[30] R. Schrader, T. Ax, C. Röhrig, and C. Fühner, "Advertising power consumption of Bluetooth low energy systems," in *Proc. IDAACS-SWS 2016*, Sept. 2016.
[31] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE J. Biomedical Health Informatics*, vol. 18, no. 4, pp. 1431–1441, 2014.
[32] R. Mahesh and T. Meyyappan, "Anonymization technique through record elimination to preserve privacy of published data," in *Proc. IEEE ICPRIME*, Feb. 2013.
[33] N. P. Smart, *Cryptography made simple*. Springer Int. Publishing, 2016.
[34] Aegenox, "BLE Advertizing Primer," 2020. [Online]. Available: https://www.argenox.com/library/bluetooth-low-energy/ble-advertising-primer/
[35] United States Centers for Disease Control and Prevention, "Evaluating and testing persons for Coronavirus disease 2019 (COVID-19)," 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-criteria.html
[36] European Centre for Disease Prevention and Control, "Case definition and European surveillance for COVID-19," 2020. [Online]. Available: https://bit.ly/34LQTK9
[37] New South Wales Ministry of Health, "COVID-19 case definition and testing advice," 2020. [Online]. Available: https://www.health.nsw.gov.au/Infectious/diseases/Pages/2019-ncov-case-definition.aspx
[38] Brazilian Ministry of Health, "About Coronavírus (COVID-19))," 2020. [Online]. Available: https://coronavirus.saude.gov.br/sobre-a-doenca#transmissao
[39] World Health Organization (WHO), "e-Library of Evidence for Nutrition Actions (eLENA) ," 2020. [Online]. Available: https://www.who.int/elena/eLENAmobile/en/
[40] World Health Organization (WHO), "WHO ZIKA App," 2020. [Online]. Available: https://play.google.com/store/apps/details?id=com.universaldoctor.zika
[41] T. Coughlin, "How much memory do cell phones need?," *IEEE Consumer Electronics Mag.*, vol. 9, no. 3, pp. 32–33, Apr. 2020.
[42] A. S. B. Neto, F. Farias, M. A. T. Mialaret, B. Cartaxo, P. A. Lima, and P. R. M. Maciel, "Building energy consumption models based on smartphone user's usage patterns," *Knowledge-Based Systems*, vol. 213, Feb. 2021.
[43] Aislelabs, "The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs," May 2015. [Online]. Available: https://www.aislelabs.com/reports/beacon-guide/
[44] M. Masoud, I. Jannoud, A. Ahmad, and H. Al-Shobaky, "The power consumption cost of data encryption in smartphones," in *Proc. IEEE OSS-COM*, Sept. 2015.
[45] J. C. Pry and R. K. Lomotey, "Energy consumption cost analysis of mobile data encryption and decryption," in *Proc. IEEE MS*, June/July 2016.



Muriel Franco is a Junior Researcher and Ph.D. student in Informatics under the supervision of Prof. Dr. Burkhard Stiller at University of Zürich UZH, Switzerland, within the Communication Systems Group CSG of the Department of Informatics IfI. Since September 2018 he is working in Zürich on cybersecurity, economics, blockchains, software-defined networking (SDN), and network function virtualization (NFV), participating and driving the work of the CONCORDIA project within a team of networking, security, and economic researchers. Besides that, from 2017 to 2020, Muriel developed jointly a federated ecosystem for offering, distributing, and execution of virtual network functions (FENDE project). Muriel holds an MSc from 2017 in Computer Science from the Federal University of the Rio Grande do Sul UFRGS, Brazil, and obtained a BSc from 2014 in Computer Science from the Federal University of Pelotas UFPEL, Brazil.

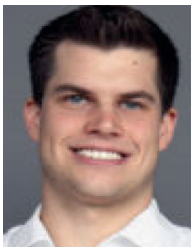
Bruno Rodrigues is a Postdoctoral Researcher at University of Zürich UZH, Switzerland, within the Communication Systems Group CSG of the Department of Informatics IfI. He received his Ph.D. in 2020 at UZH, focusing on blockchain-based collaborative network defenses, and his MSc from the Polytechnic School of University of São Paulo, Brazil, in 2016, where he worked on research projects in partnership with Ericsson Research focused on network management based on SDN and energy efficiency. his expertise and research is on collaborative network defenses based on Blockchain and he works on research projects, like CONCORDIA in the scope of cybersecurity and PasWITS in the area of wireless tracking.

Christian Killer joined the Communication Systems Group CSG, Department of Informatics IfI at University of Zürich UZH, Switzerland, in February 2019 as a Junior Researcher and Ph.D. student in Informatics to obtain his Ph.D. degree under the supervision of Prof. Dr. Burkhard Stiller. He finished his MSc Degree in Informatics at University of Zürich UZH, focusing on Security Management and Visualization in a Blockchain-based Collaborative Defense. His research is on the security of electoral processes and blockchain-based remote electronic voting systems.

Eder John Scheid is a Junior Researcher and Ph.D. student since December 2017 under the supervision of Prof. Dr. Burkhard Stiller at University of Zürich UZH, Switzerland, within the Communication Systems Group CSG of the Department of Informatics IfI. He holds an MSc degree in Computer Science from the Federal University of the Rio Grande do Sul (UFRGS), Brazil, which he obtained in 2017. He focuses his research on blockchains, smart contracts, policy-based network management, and network virtualization.

Alessandro De Carli is the founder and engineer of Papers.ch a mobile and crypto software company. He holds a BSc (2013) and MSc (2016) degree both from University of Zürich UZH, Switzerland, with a focus on Management Information Systems and Services. He wrote his Master Thesis with the Communication Systems Group CSG on blockchains in 2015.

Andreas Gassman is a Software Engineer at Papers.ch with experience in Web-related technologies. He holds a BSc (2017) from the Fachhochschule Nordwestschweiz FHNW, Switzerland.



David Schönbächler is with Papers.ch and holds a BSc in Finance (2015) from Fachhochschule Nordwestschweiz FHNW, Switzerland, and a MSc (2018) in Economics from University of Basel, Switzerland.

Burkhard Stiller received the Informatik-Diplom (MSc) in Computer Science and the Dr. rer.-nat. (Ph.D.) degree from the University of Karlsruhe, Germany, in 1990 and 1994, respectively. In his research career he was with the Computer Lab, University of Cambridge, U.K. (1994-1995), ETH Zürich, Switzerland (1995-2004), and the University of Federal Armed Forces Munich, Germany (2002-2004). Since 2004 he chairs the Communication Systems Group CSG, Department of Informatics IfI, at University of Zürich UZH, Switzerland. Besides being a member of the editorial board of the IEEE Transactions on Network and Service Management, Springer's Journal of Network and Systems Management, and the KICS' Journal of Communications and Networks, he is the past Editor-in-Chief of Elsevier's Computer Networks journal. His main research interests are published in well over 300 research papers and include systems with a fully decentralized control (Blockchains, clouds, peer-to-peer), network and service management (economic management), Internet-of-things (security of constrained devices, LoRa), and telecommunication economics (charging and accounting).