Digital Contact Tracing : Software Architecture

Dibyendu Baksi, PhD

Indepedent Researcher, Lawrenceville, GA 30043, USA

Abstract

The covid-19 crisis is providing a lot of impetus to the search for innovative technological solutions to solve major problems of tracking and containment of the pandemic. The major cornerstones of testing, isolation, contact tracing and quarantine are well understood and agreed upon at a general level. In this paper, the software architecture required for implementing successful digital contact tracing applications is elaborated. The goal of contact tracing is to proactively identify the infection chain of the population including asymptomatic people coming in contact with infected people who tested positive, i.e., to avoid asymptomatic people from spreading the disease without any intention. The entire ecosystem of contact tracing is explained so that the real challenges of integrating the key healthcare components are appreciated.

Index Terms : Contact tracing, public health, surveillance, software architecture and UML.

Impact Statement— Digital contact tracing is the key to get a good idea of how the current covid-19 pandemic is spreading so that appropriate control measures can be taken. A software architecture viewpoint is the place to start to understand the challenges for all potential implementors.

1 Introduction

As the SARS-CoV-2 pandemic unfolds and drastic control measures to limit the spread are taken, the two main goals are the same in front of all nations.

- maintain low case numbers of newly infected population so that healthcare systems are running just below capacity and
- relax restrictions in a phased manner to a return to normal life to minimize socioeconomic toll

The cost benefit and economical aspects of a complete lockdown versus aggressive reopening in terms of saving lives versus wrecking economies will vary

depending on the context [1]. On the other hand, a trade-off analysis of the long term dynamics of applying different policies in the framework of Control theory from engineering is available in [2].

In the context of control systems from an engineering standpoint, the major components involve measurement to get accurate data and control policies to keep the variables of interest to their desired levels. Measurement in this context is testing and control policies are actions such as isolation and quanrantine. Contact tracing is more like an observer that tries to predict the social network graph of probably infected people that need testing. A good definition of these measures is given in [11].

Asymptomatic transmission is quite common in Covid-19 and there are research papers estimating this to be around 40% []. This is a major containment problem with COVID-19 as infected cases can transmit the virus 1-3 days before they develop the symptoms before falling ill. *Contact tracing seeks to unconver the social or network graph of contacts of an infected person so that preemptive action can be taken.*

The following section introduces the problem of contact tracing via the user scenarios. It outlines the steps typcially involved in the process of contact tracing from the point of view of the user.

2 User scenarios

Availability of usable solutions whether done via the market economy or command economy requires precise specification of the problem leading to design options that can accommodate all the aspects. There are two ways to doing contact tracing.Traditional ideas and practices of contact tracing shown in Figure 1 involved a centralized public health tracking or surveillance system. Information of disease outbreaks with confirmed cases and possible cases are gathered using manual (i.e., interviews) or other means from test labs etc.

- Interviews : This is the standard method of doing it manually. Interviews have their limitations as they take time to conduct and are not scalable in terms of resources as human being are involved. The other aspect is the dependence on the ability of the person being interviewed to recall the details regarding the proximity of the contact.

- Digital proximity : Using cellphones and the bluetooth network, identifying people who come in close proximity is the main idea behing digital contact tracing. Bluetooth low energy network is used to record cellphones that were in proximity in terms of variables such as anonymized temporary identifiers, distances, timestamp etc. There are lots of options for implementation of such an idea along many angles such as use location tracking, privacy concerns related to data elements being tracked, how alerts are delivered etc.

There is a big trade-off between getting more data for the cases or possible cases and privacy concerns in democratic nations without support from the majority

of people. The general idea of tracing based on digital proximity is shown in Figure 1. Day 1 activities show how an asymptomatic subject carrying the virus comes in contact with others at home, work and public transporation (e.g., train) and then upon reaching home starts showing sympotoms of the diease. On Day 2, after getting tested, he/she alerts all the contacts from Day 1 via the mechanisms described here in this paper who then take appropriate measures. In this paper, the focus is on contact tracing using current digital technology and

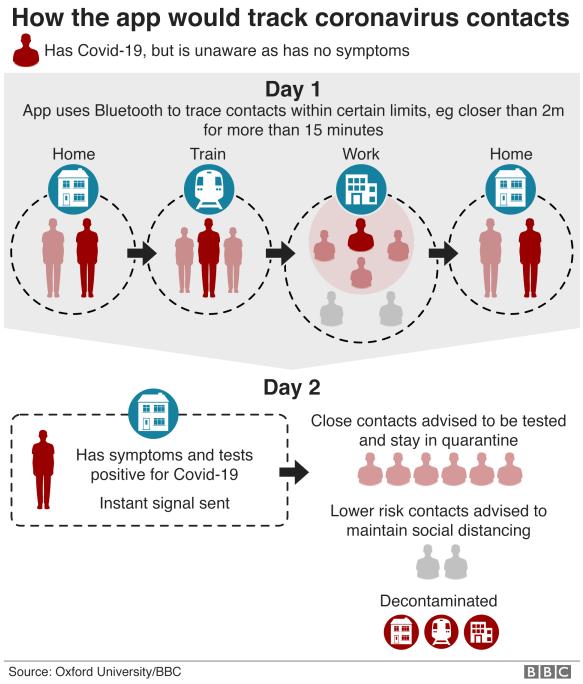

Figure 1: Contact tracing basics

smart design practices to maximize the exploration of the underlying exposure network graph with minimal invasion of privacy.

3 Architecture of Digital Contact Tracing :

The goal of contact tracing is to identify the entire set of asymptomatic people who could carry the virus and would test positive without waitng for them to develop symptoms and then arrive at testing locations. One of the goals of this communication is to demystify the 'behind-the-scene' details of contact tracing applications that are touted as solutions so that the responsible party (e.g., local, state or federal public health agencies) can make a judicious choice regarding which option to choose. In other words, what one sees in an app is only part of the puzzle behind a successful contact tracing solution.

It is critical to define the overall architecture of something important like digital contact tracing before jumping to a solution. There is a lot of confusion in terms of what exactly is offered by a vendor and if it provides the end to end solution of the problem or problems being tackled. It is also important to understand the trade-offs, overlapping areas and capabilities of solutions in the healthcare space to make full use of any product. Since business process lies at the heart of any transformation or modernisation, it is an appropriate place to start. Business process activities will result in the overall architecture whihc is a 3-tuple, i.e, {Components, Connections, Configuration}. The following sections provide the two major confugurations (centralized and decentralized) with fairly detailed view of the components and their connections or protocols.

The main question around the minimum amount of data needed to alert someone who came to close proximity of an infected person sparks heated debates around what information to collect and who triggers the event. The answer to the question also provides the extent to which epidemiological research can be conducted to decipher the evolution of the disease spreading process in terms of the underlying social network graph.

For a centralized approach, there must be way to identify the smartphones that came to proximity and a way to link them to actual tests being done in testing facilities and laboratories. Since the core task of matching the infected cases to asymptomatic ones takes place centrally as shown in Figure 2 in the 'Centralized Contact Tracing Subsystem' (CCTS) swimlane, it also needs to be the source of the temporary identifiers that the smartphones use in their proximity protocol using bluetooth technology. The backend public health system will only store the anonymous temporary IDs from the phones in this approach. Once someone tests positive after a test is conducted and a matching algorithm is run centrally to identify the susceptible contact, that specific app is alerted. No information is added or removed in CCTS.

A confirmation of the proximity proof of being close to an infected person and the duration/timestamp of that event are all that a digital contact tracing application needs to alert the asymptomatic person. Other public health surveillance or EMR (electronic medical record) information systems with data about the infected person, their contacts, location of that contact, context for that contact etc are unnecessary to be part of the contact tracing application on the phone. This results in the possibility of decentralized implementation to keep minimal amount of data on the device (i.e., phone) as part of the core contact tracing app.

The main business process of a decentralized digital contact tracing is shown Figure 3. The two bottom swimlanes of 'Contract Tracing Backend' and 'Laboratory or Healthcare Providers' are something not visible to the end users of the digital contact tracing applications although they play crucial roles. The approach to identifying the underlying contact netowrk graph in the decentralized involves minimal data about the infected person and its contacts. Only the list of 'infected', i.e., people who tested positive is either downloaded by the app at a certain interval or pushed via some altering mechanism. The identifiers are good enough to run the matching algorithm locally on the smartphones of the

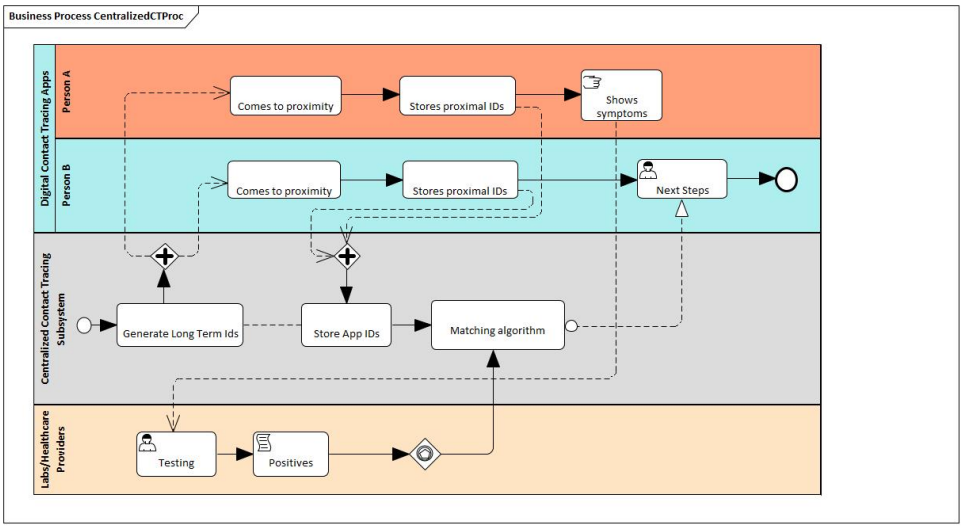

Figure 2: General process : centralized

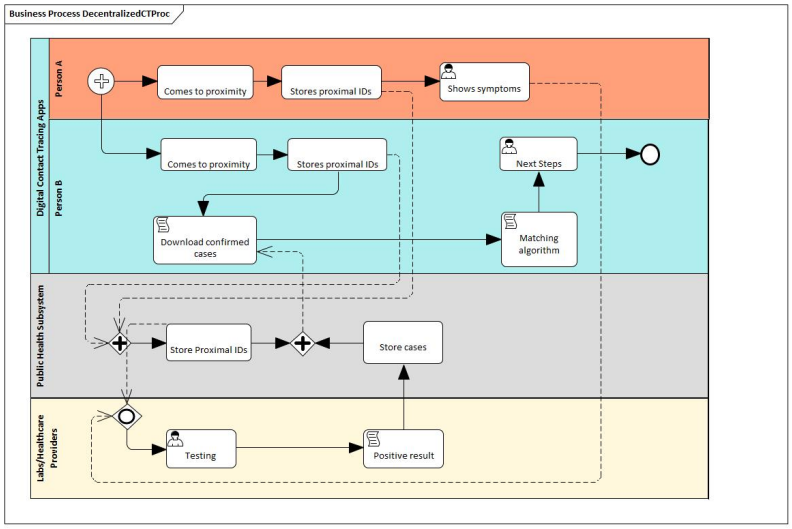

Figure 3: General process : decentralized

asymptomatic people who came in contact with the infected person.

The proximity tracing process is supported by a backend server that shares anonymous contact information with the app running on each phone. This backend server Then, they will instruct their phones to upload to the backend a compact representation of their EphIDs for the infectious period. The backend stores these compact representations.

3.1 Centralized design

The sequence diagram in Figure 4 shows two people, Ryan with an iPhone and Eric with an Android based phone come in to contact and later on Ryan tests positive for the virus after having symptoms. Later on, Ryan provides consent to the local centralized contact tracing systems via a public health app on iPhone and upon receiving a code from the authorities, uploads his results. Eric's Android phone eventaully gets an alert from the authorities as he was identified as someone who came in close contact with Ryan. The matching algorithm determines a score which dictates the next steps to be taken by Eric. A 'Centralized Contact Tracing' system is the key component in this design. It generates and holds a long-term pseudo-identifier and generates the ephemeral pseudo-identities (EphID s) for the smartphones. These EphIDs are then received by smartphones that come to close proximity thats locally stores them with the corresponding proximity and duration. The centralized systems also updates the central public health federal system so that epidemiological research can be undertaken by scientists at the federal level with this data.

The detailed design of a contact tracing application requires a few components beyond what one can view in the app itself. The success of contact tracing is critically dependent on all these components to work seamlessly with agreed upon protocols, messaing and data standards.

3.2 Decentralized design

The decentralized approach shown in Figure 5 preserves privacy better as it stores less information to the backend public health system. Smartphones locally generate and broadcast via Blutooth LE technology frequently changing ephemeral identifiers (EphIDs). Neighboring smartphones listen to these EphIDs and store them together with the duration and timestamp. At a certain frequency, all smartphones download the list of patients who got infected and reconstruct the corresponding EphIDs of infected patients locally. If the smartphone has stored a record of any of these infected EphIDs, then the smartphone's user has been in contact with an infected person and the smartphone computes the owner's risk score. If this score is above the threshold the smartphone initiates a notification process. The sequence diagram clearly shows the simpler trace of how after being tested positive and providing consent, Ryan's phone uploads test results to the local public health system. Eric's Android phone periodically downlods the list of infected people with a time window and can detect locally the fact that he did come in contact with Ryan.

3.3 Implementation and Deployment Aspects

The implementation of the design outlined in the previous sections can make use of any of the modern software development stacks. The main components as shown in Figure 6 are (i) front-end app platform, (ii) contact tracing API, (iii) a public health contracting service for the centralized option, (iv) a local or

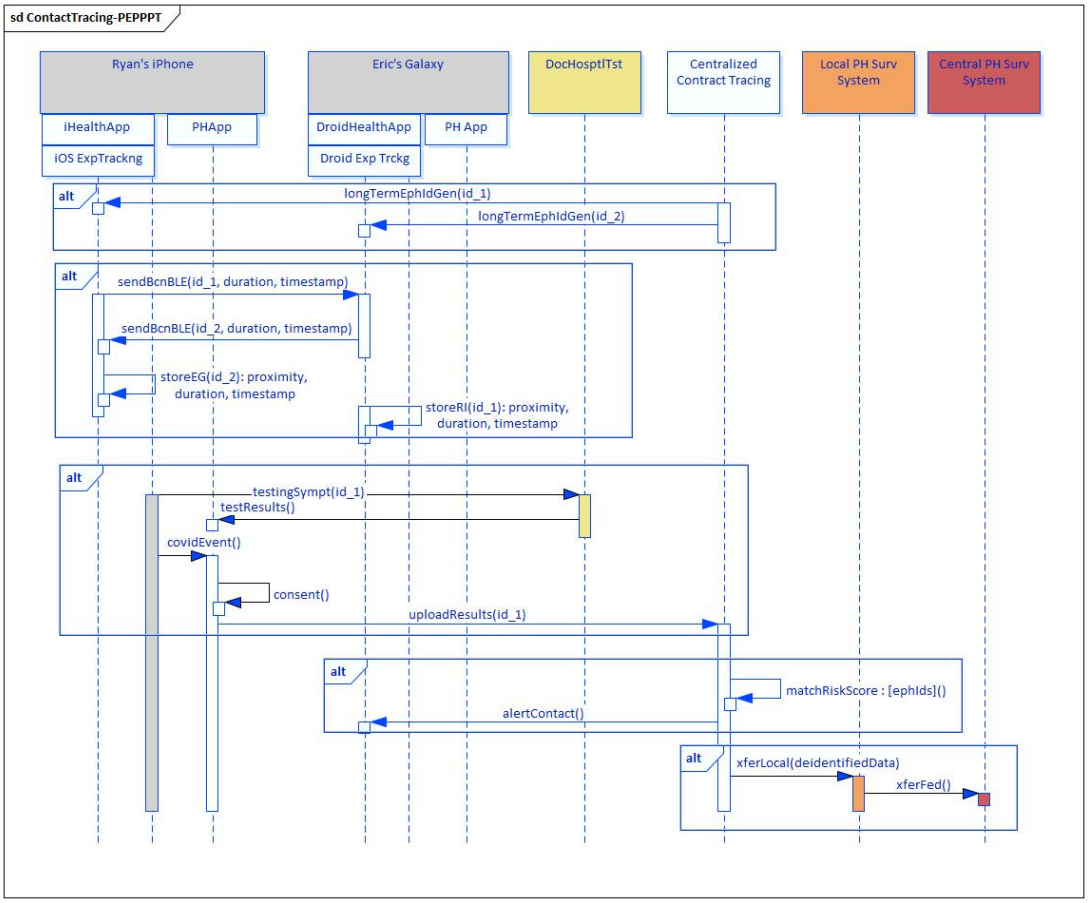

Figure 4: Sequence Diagram of Centralized Contact Tracing

state public health surveillance system and (v) testing facilities or laboratories reporting system.

The front-end can be a single integrated contact tracing app with a built in public health component or can be two separate apps that where the contact tracing app is dependent on the public health app. The contact tracing APIs are the interfaces being worked on by the two leading providers Apple [8] and Google [9]. It is important to understand that the providers are providing the APIs that can be used by anyone willing to create such apps. The backend system for the centralized option is the most debatable component as this evokes images of the 'Big Brother' although the amount of data collected is the key parameter here. The last two components are typical of public health infrastructure in many developed nations and most of the actual epidemiological analysis and research happen in those systems be it at regional, state or federal levels.

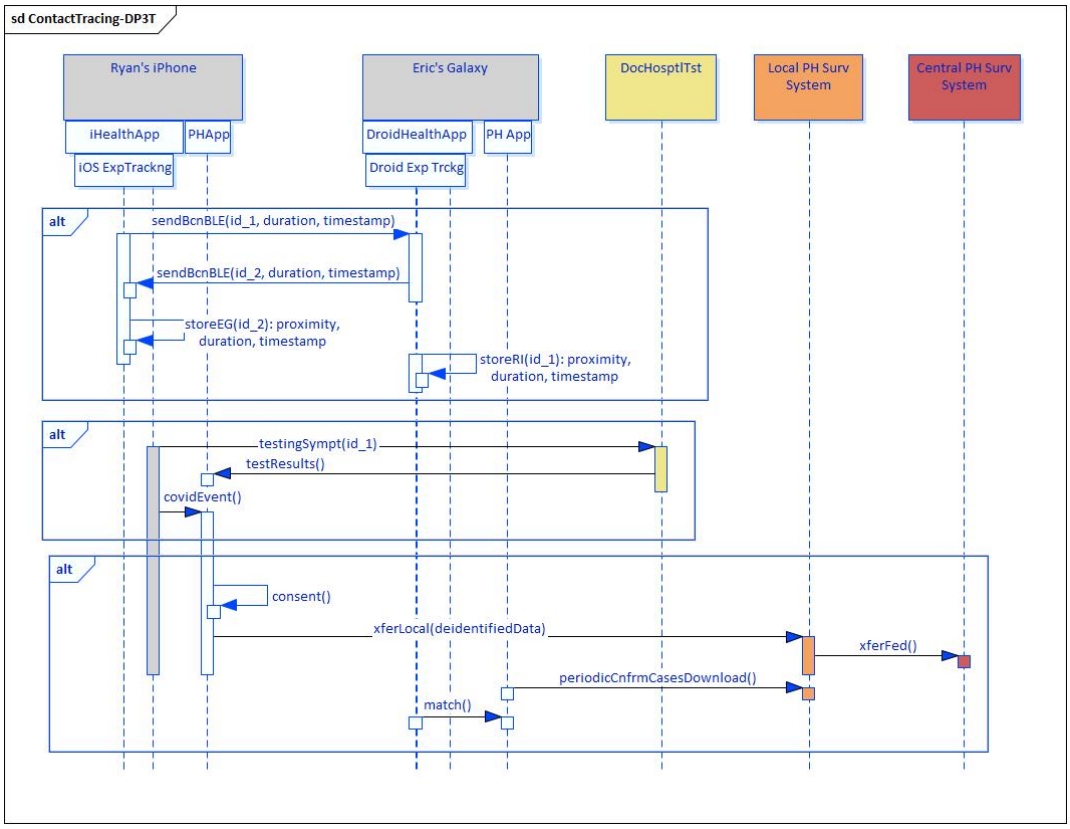

Figure 5: Sequence diagram of Decentralized Contact Tracing

As is shown, the deployment aspects of the different components can have many variations. As an example, the centralized component for that type of approach can be hosted in an on-prem data center as would be many healthcare providers and the federal public health or disease reporting systems. On the other hand, laboratories and regional or state health systems can be on a public or private cloud. Any implementaiton needs to work out the network and security issues before declaring readiness to roll out apps. The red arrows show the information flow between components in the centralized design while the others are common to both.

4 Security and Privacy Issues

Automated digital contact tracing as an aid to support of public health response is a top priority now. A lot of countries are trying to accelerate the use of these proposed apps and their associated backend components which are pre-

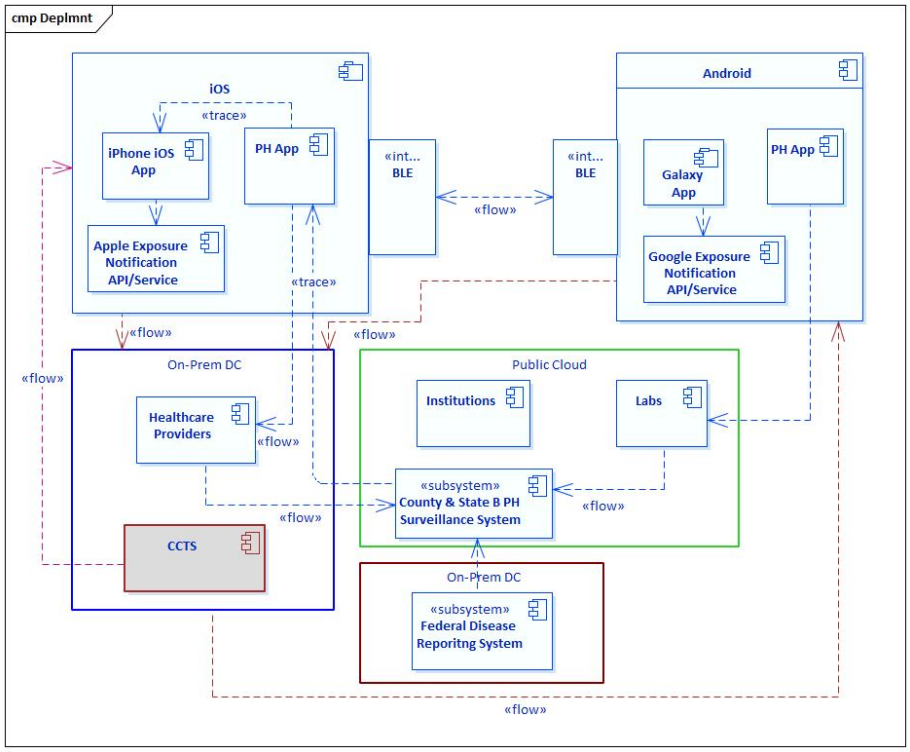

Figure 6: Deployment diagram of DP3T

domainatly centralized by the Federal goverments resulting in debates around privacy concerns. At one end of the spectrum are countries like China, South Korea, Singapore with strong centralized goverments and advanced technology and surveillance capabilities. At the other end of the spectrum are European countries like Germany or the United States who have the technological capability but there is an intense debate and politics involving federal and state governments on policies and privacy concerns. A good example is Germany where the decision to build a centalized architecture for contact tracing suffered a setback and the final decision converged on a decentralized architecture [6]. Concerns about allowing the Goverment access to pseudonymized proximity data and social graph of individuals in the society are raising alerts of state surveillance by privacy experts.

The Decentralised Privacy-Preserving Proximity Tracing (DP-3T) project is an open protocol for COVID-19 proximity tracing produced by a core team of over 25 scientists and academic researchers from across Europe. It has also been scrutinized and improved by the wider community. DP-3T members have been participating in the loose umbrella of the 'Pan-European Privacy-Preserving Proximity Tracing' (PEPP-PT) project. DP-3T is not the only protocol un-

der this umbrella. PEPP-PT also endorses centralized approaches with very different privacy properties. Pandemics do not respect borders, so there is substantial value in PEPP-PT's role of encouraging dialogue, knowledge-sharing, and interoperability.

A naive example of centralized architecture is India's ArogyyaSetu [12] application that not only uses location services, stores PII such as Profession and other other sensitive personal data but also is mandatory for citizens to download ! The benefits of such solutions created most likely as a quick response by technology teams without much architectural thought or domain expertise are questionable especially in a country with a poor or almost non-existing public health surveillance infrastrucuture or leadership ! Singapore, on the contrary, has created TraceTogether app using the framework suggested by MIT that uses the architecture described here in a centralized manner witout using location services of the smartphones.

5 Conclusions

Before contact tracing applications are rolled out, there needs to be a cohesive policy and agreements between Central, Local governments, Legal systems and Technology privacy groups. Informed leadership with collaboration between public health experts, epidemiologists and technology experts with solid architectural background can create all encompassing solutions in a crisis like what we are experiencing today. It is a great opportunity to engage and leverage the expertise of true Enterprise Architects instead of depending on pure technology hypes or medical professionals as this is a medical informatics challenge. Unless a cohesive picture of the entire ecosystem is defined and appreciated, proper investment decisions cannnot be made in a phased manner to identify and tackle the priorities. An environment to start the dialog and collaboration involving the proper stakeholders is the need of the hour. The objective of this paper is to show a comprehensive view from the standpoint of Enterprise Architecture of all the multitude of aspects involved in digital contact tracing and illustrate the point that a fancy app in a phone, as described in popular media, is just the tip of the iceberg. The discourse needs to happen at the level descrbed here to make serious progress.

References

[1] Neil Bailey, The calculus of death shows the COVID lock-down is clearly worth the cost -https://bit.ly/35X6HKL, The Conversation, 2020.

[2] Spectrum latest issue, https://spectrum.ieee.org/biomedical/diagnostics/how-control-theory-can-help-control-covid19

[3] Mary Shaw and David Garlan,PEPP-PT Pan-European Privacy-Preserving Proximity Tracing, March 2020.

[4] Carmela Troncoso et al, Decentralized Privacy-Preserving Proximity Tracing, April 2020.

[5] Marcel Salathe and Ciro Cattuto, COVID-19 Response: What Data Is Necessary For Digital Proximity Tracing , April 2020.

[6] Natasha Lomas, Germany ditches centralized approach to aoo for COVID-19 contacts tracing - https://tcrn.ch/3ct1QU5, April, 2020.

[7] Juli Clover, Apple's Exposure Notification System: Everything You Need to Know - https://www.macrumors.com/guide/exposure-notification, MacRumors, May 4, 2020.

[8] Dev Building an App to Notify Users of COVID-19 Exposure - https://apple.co/3dCVL7J, April, 2020.

[9] Dev TeamExposure Notification : Android API Documentation, v1.2, April, 2020

[10] Zak DoffmanForget Apple And Google—Contact-Tracing Apps Just Dealt Serious New Blow - https://bit.ly/2X3e7s7, May, 2020.

[11] Neil M. Ferguson, Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand, March 16, 2020.

[12] Anand Venkatanarayanan, Covid-19 : How The Aarogya Setu App Handles Your Data- https://bit.ly/35TSE8W, BloombergQuintOpinion, April, 2020.