



GENo-P - Global Exposure Notification Protocol


Objectives

We propose a protocol that could be used to quickly analyse anonymized interactions and to notify people at risk. It must take into account global scenarios, thus it has been designed to support any device with Bluetooth LE technology and internet connection (Android, iPhones, custom hardware).

The protocol must deal with anonymized data.

The virus spreads faster than expected, so in order to notify anyone who is at risk, the analysis must explore the anonymized graph of interactions up to N degrees of connectivity.

The protocol natively invites whoever implements UTP to collaborate to better identify people at risk. We call roaming protocol the part of UTP designed for this objective.




Our solution (up to 98.5%)

Others (up to 71.7%)

Coverage

Our technology can anonymously monitor from 91.2% to 98.5% of all mobile interactions (iPhone and Android), compared to 71.7% of traditional technologies.


A linear reduction of Covid-19 R0, due to the greater number of anonymously intercepted interactions, has an exponential impact on the reduction of the spread of the virus.




Interoperability

We allow interoperability through the distribution of open-source SDKs that implement our privacy-preserving protocol.

Transparency is important to us! Everything we do is open source and available. Visit our organization on GitHub and GitLab:






Roaming

The open-source protocol supports the interoperability among the nations that will implement it, managing this way global scenarios.

We give the possibility directly to selected expert virologists to establish the rules and notifications to be sent to patients, country by country.

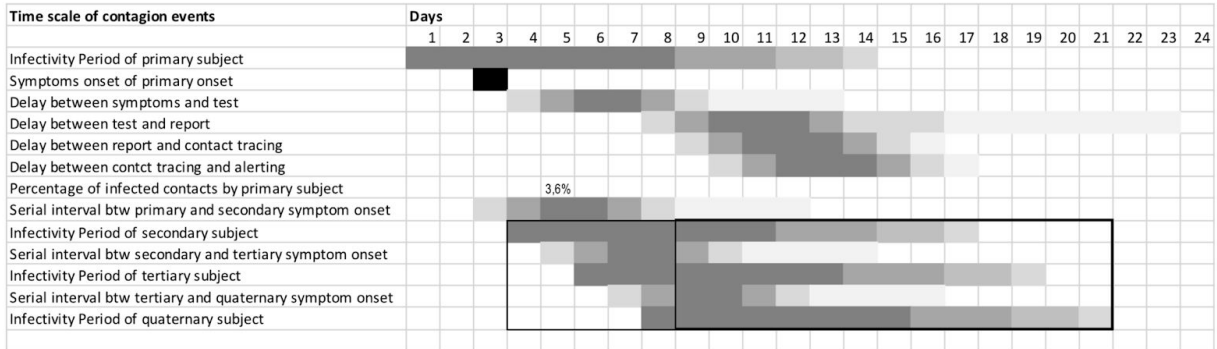


Speed of reaction

We can automatically contact up to N degrees of connection of a positive diagnosed user. In addition, the notification can be sent directly from the laboratories where the swab is analyzed.

Our platform also identifies patients who do not exhibit symptoms.

Transmission chains can be interrupted only if secondary and tertiary contacts accept to share their data for alerting



Time scale (days) of critical contagion events, based on delays as seen in Lombardy (Feb-Mar 2020). The thin-line boxed area covers the transmission to secondary, tertiary and quaternary contacts that - given the current delays in testing and reporting - cannot be prevented by contact tracing. The thick-line box covers the events that could be prevented by digital contact tracing, **provided that secondary and tertiary contacts accept to share their respective cronologies with public authorities for triggering the alert throughout the contact chains.**

22

Actors

There are 5 main actors:

- Users: people who use CovidApp
- Doctors: people who use CoviDoc
- Epidemiologists: who can access the Epidemiologist Internal Panel
- Local Authority: whoever implements the UTP Covid Community Alert protocol
- Central Authority: the authority that will keep the mapping (Installation_ID, Masked_ID) and the log of which Masked_ID has been seen by which Local Authority.

Technological limitations

We use Bluetooth LE to estimate the distance among devices, those being Androids or iPhones. We measure the RSSI of the Bluetooth LE signal transmitted by a device to estimate the distance from the receiving device/s.

In order to make this protocol universal, we needed it to be compatible also with iPhones. We faced some technology limitations due to iOS:

1. Apps can use region monitoring to be notified when a user crosses geographic boundaries or when a user enters or exits the vicinity of a beacon. A total of 20 beacons at maximum can be registered on an app.
2. An iPhone app can transmit Bluetooth LE signals if it runs in the foreground, but as soon as it starts running in the background, it can't transmit signals anymore.

Device OS	Market Share ¹
Android	72.26%
iPhone	27.03%
Others	0.71%

Device of infected user	Device of user at risk	% of total interactions	Can the infected user's device detect the device of the user at risk?	
			Other solutions (except Apple and Google APIs)	Covid Community Alert
Android	Android	52.2%	Yes	Yes
Android	iPhone	19.5%	No	Yes
iPhone	Android	19.5%	Yes	Yes
iPhone	iPhone	7.3%	No	<ul style="list-style-type: none"> • No, if completely isolated. • Yes, if one or more Android are close to them.
Theoretical limit of users that we can notify:		98.5%	71.7%	91.2% - 98.5%

¹ <https://gs.statcounter.com/os-market-share/mobile/worldwide>

Workarounds

Since iOS allows the monitoring of a total of 20 iBeacon, we decided to use a predefined iBeacon on which the Masked_ID (more on this later) of the devices will be transmitted.

By using a predefined iBeacon, even if iPhones are invisible to Androids once their app runs in the background, they have the possibility to receive the iBeacons transmitted by Android devices, and therefore they will be able to log their interactions.

The Bluetooth signals are known to be noisy, thus to clean the signals each device collects and aggregates the received Bluetooth LE signals if they belong to the same source. The aggregation could be extended up to 3 minutes of logging, and the aggregation operation will sort the values to get the 50th percentile of the sorted RSSI. This will remove spikes due to fluctuation of the Bluetooth LE signals and will return a better estimation of the distance between the two devices. We decided not to use a Kalman filter, more suited for real-time operations, since it's not critical for the application to be real-time. A statistical correction every 3 minutes is a good compromise on the responsiveness of the application and the quality of the approximated distance.

By sending the interactions to the Local Authority, we can duplicate the number of signals analysed since we will average the 50th percentile of the RSSI for the interaction logged by both the devices.

When the iOS applications send the interactions they logged to the Local Authority, we can associate the same interactions to the Android who transmitted the received Bluetooth LE signals in the first place. This allows iPhones to be seen by Androids even if they never transmit when the application runs in the background.

The 16-byte long iBeacon identifier that has been pre-defined to wake up iOS applications running in the background, cannot be used to encode the Masked_ID. We then need to use the minor and major fields in the Bluetooth LE protocol to encode the Masked_ID. These are two 2-byte long fields.

The major and the minor are used to code the Masked_ID of the device itself in the following way:

- $\text{minor} = \text{integer}(\text{Masked_ID} / 65536)$
- $\text{major} = \text{integer}(\text{Masked_ID} \% 65536)$

This means that given minor and major values we can retrieve the Masked_ID if the device with the following operation:

$\text{Masked_ID} = 65536 * \text{minor} + \text{major}$

The concatenation of 2-byte long fields gives us the possibility to encode 2^{32} different device Masked_IDs.

Given the relatively small number of available Masked_IDs because of the available payload in the minor and major fields, during the initial setup, the device will ask the Central Authority for an list of Masked_ID to be used in the Bluetooth LE iBeacon.

To protect users' privacy, the Central Authority will generate an Installation_ID and will return a list of Masked_IDs and a list of iBeacon identifiers that the devices will use within a span of 10 minutes each. This list will contain a list of Masked_IDs and iBeacon identifiers valid for at least one week, thus the message will not exceed $(16B+2B+2B)*7*24*6 \approx 20KB$.

This workaround gives us the ability to detect interactions between Android-Android, iPhone-Android and Android-iPhone bringing the theoretical limit of users at risk that we could notify from 71.7% to 91.2%.

It follows that to detect interactions between two iPhones, we need one or more Androids relatively close to them to triangulate these two devices relatively to the local group of devices whose interactions have been logged by both the two iPhones in analysis.

This would bring the theoretical limit of users at risk that we could notify from 91.2% up to 98.5%.

An interesting property of this triangularization is that it works best when the environment is crowded (bus, tube, offices, etc.) and therefore more risky in terms of contagiousness, since the probability of being close to an infected individual $F(g)$ can be modeled as a sigmoid-like function, where g is the number of people we are close to.

Registration

User app protocol

1. As soon as the setup starts, we run a Google invisible recaptcha to avoid malicious attackers to generate Installation_IDs through a flood attack and saturate the protocol, whose limits are imposed by the Bluetooth LE standards.
2. Recaptcha challenge will only be shown when a suspected activity is detected.
3. If the challenge is passed or the invisible recaptcha has not been shown, the app sends to the Central Authority a request to generate a new anonymous internal Installation_ID.

Covid Community Alert

Mobile: +44 (0) 7858 61 3934

Email: coronavirus.outbreak.control@gmail.com

Website: <https://coronavirus-outbreak-control.github.io/web/>

4. The Central Authority replies with a list of Masked_IDs and a list of iBeacon identifiers that the devices will use within a span of 10 minutes each. This list is saved locally on the device itself.
5. Subsequently, the device sends to the Central Authority the ID token created from iOS or Android that allows the reception of push notifications. The association between the internal Installation_ID, generated by the backend of the Central Authority, and the token relative to the push notification is saved on the Central Authority database. The token generated by iOS or Android is unique for each application installed on the device and cannot be linked to the user device. Only Apple or Google knows the association between the tokens and the devices.
6. The devices continuously publish via Bluetooth LE technology the Masked_ID_i on the iBeacon_i associated with the current 10 minute slot-time_i, before changing it with the next ones.
7. The Android devices continuously advertise on a predefined iBeacon to awaken nearby iPhones with a predefined frequency.
8. The devices continuously scan the surrounding environment and detect the Masked_IDs of close devices, if they find any. Such information is stored locally together with other metadata - such as timestamp and the Received Signal Strength Indication (RSSI) of the Bluetooth LE signal detected.
9. The scan lasts for 10 seconds or more to capture as many Bluetooth LE signals as possible every time the devices get awakened.
10. RSSI signals are aggregated within a 3-minutes time frame to remove spikes due to fluctuation of the Bluetooth LE signals.
11. The devices send to the Local Authority the logged interactions with all the Masked_IDs, and subsequently deletes the logs.
12. Periodically the application deletes old interactions to keep avoid consuming storage on the device.
13. The backend of a Local Authority stores the interactions of the received Masked_IDs on a secure cloud database. Our open source implementation uses Amazon AWS - encrypted S3 buckets, which are not publicly accessible.
14. The Local Authority periodically sends to the Central Authority the list of Masked_IDs received to support the roaming protocol.

15. The app can display on the mobile screen its current Masked_ID as a QRCode.

Emergency/Medical app protocol:

1. When an infection case is diagnosed, the medical staff scan the patient's QRCode and send to the Central Authority the scanned Masked_ID together with the patient's health status - "Confirmed Case of Coronavirus", "Suspected Case of Coronavirus".
2. Given a patient's Masked_ID, thanks to the step #14 of the User app protocol, the Central Authority sends to all the Local Authorities that saw the patient during the previous 14 days all the Masked_IDs they have seen associated with the same patient.
3. The Local Authorities will retrieve the Masked_IDs the infected patient has interacted with by searching for them in their secure database, then they will send these Masked_IDs back to the Central Authority. The Local Authority will use the predefined set of rules created through the Internal Epidemiologist Panel to find the Masked_IDs of possible people at risk by analysing their distance and time of exposure from the infected person.

The screenshot displays the 'Coronavirus Outbreak Control' web application. In the background, the 'Virologist Interface' shows a list of notifications with columns for 'Exposure Distance' and 'Exposure Time'. Three notifications are visible: 'Immediate (< 40cm)', 'Near (40cm - 2m)', and 'Far (2m - 20m)'. Each notification has 'EDIT' and 'DELETE' buttons. A modal window titled 'Edit Notification' is open in the foreground. It contains the following fields: 'Exposure Distance' (a dropdown menu set to 'Immediate (< 40cm)'), 'Exposure Time' (a slider set to '30 - 60 minutes'), 'Exposure Type' (radio buttons for 'Suspected' and 'Confirmed', with 'Confirmed' selected), 'Message' (a dropdown menu set to 'High'), 'Title' (a text field containing 'High - Immediate - Confirmed'), and 'Description' (a text area containing 'A high severity notification that is triggered when the user is immediately close to someone with a confirmed coronavirus case for 30 - 60 minutes'). The modal also includes 'CANCEL' and 'SAVE' buttons at the bottom right.

4. The Central Authority, that has the mapping (ID, Masked_ID) and the corresponding Google or Apple notification token, will send a silent push notification to the user at risk using the message specified by the Local Authority.

5. The Central Authority will find all the previous Masked_IDs of the new set of people at risk and will iterate this process one more time to notify people at risk up to 3 degrees of connections from the original infected person.
6. As soon a person has his/her QRCode scanned by a doctor, all the network will be notified using the rules for a suspected case specified by the epidemiologists through the internal panel. This will save days waiting for the response of the swab.

All communications use HTTP over Transport Layer Security (HTTPS), an encrypted connection via asymmetric cryptography. The certificates are handled by Amazon AWS.

ID Randomisation

Fully decentralised protocols have to find good encryption functions to mask the randomised IDs that each device transmits. The encryption should be reliable because people could listen to a very high number of samples and could, therefore, infer the original user ID.

In our case, the backend sends randomly chosen IDs to each device. We can protect our Installation_IDs by running two operations on our backend:

1. we hash the Installation_IDs with random seeds and very secure cryptographic hash functions;
2. we sort these hashes to map them to a 2^{32} bits space.

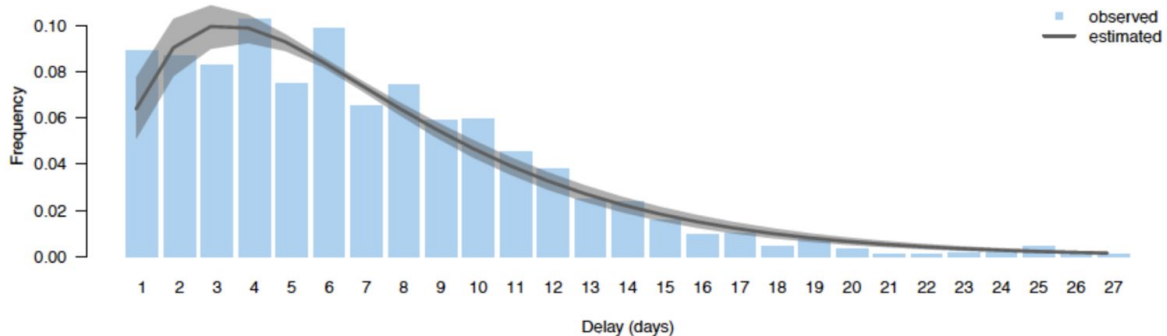
By hashing and then sorting all hashes of all our IDs we make every Masked_ID dependant from all the others, thus making the transmission of such Masked_IDs even more secure.

Data on Local Authority and Data on Central Authority

Each Local Authority only stores interactions among Masked_IDs. By saving them on the Local Authority datacenter, we can easily store all the interactions of the last 30 days to even cover the entire distribution of symptoms delays without incurring any limitations due to the device's

limited storage capability.

Reporting delays for the symptom onset dates



Distribution of the test reporting delays for the symptom onset dates.
The bars represent the data and solid line represents the mean of the best fitting gamma distribution (the shaded area represents 95%CI).

Source: Cereda D. et al, The early phase of the COVID-19 outbreak in Lombardy, Italy, <https://arxiv.org/ftp/arxiv/papers/2003/2003.09320.pdf>

The Central Authority stores the mapping between Installation_ID and its Masked_IDs and all the Masked_IDs seen by all the Local Authorities.

The Central Authority will not store information about the interactions.

The Google and Apple notification IDs saved in the Central Authority cannot identify a person, nor their device, but only the current installation of the application.

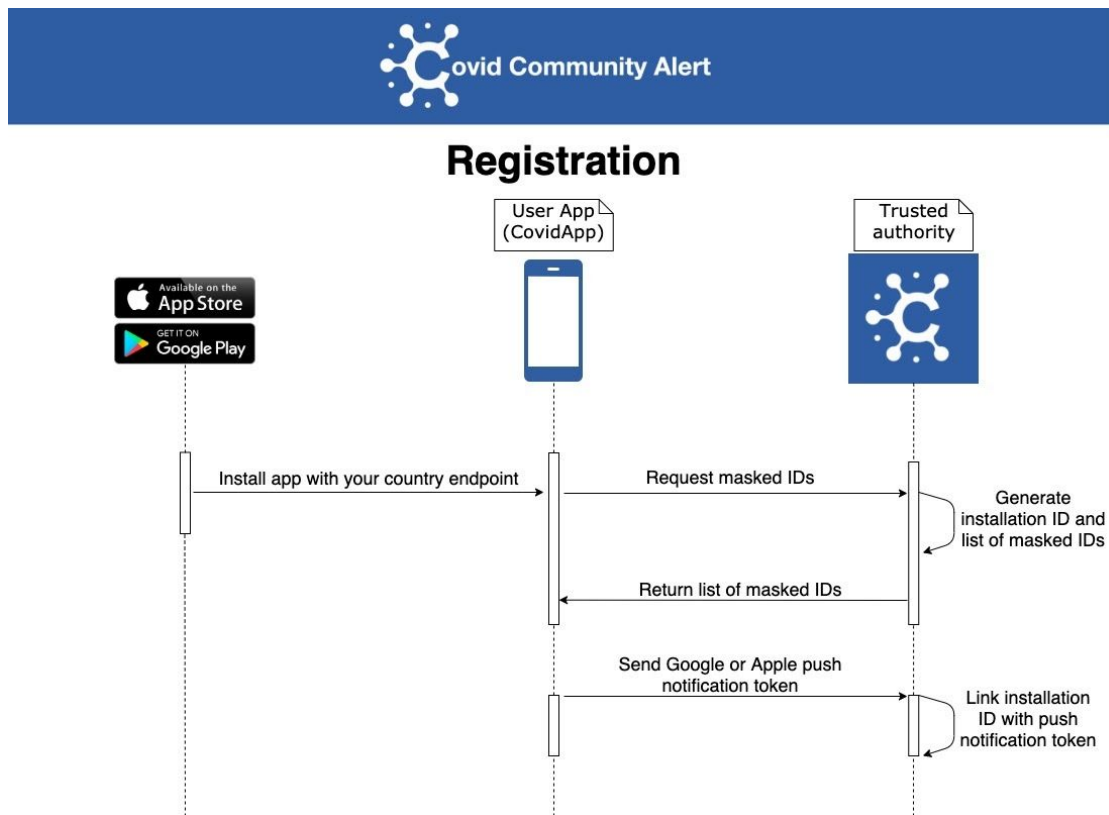
Covid Community Alert

Mobile: +44 (0) 7858 61 3934

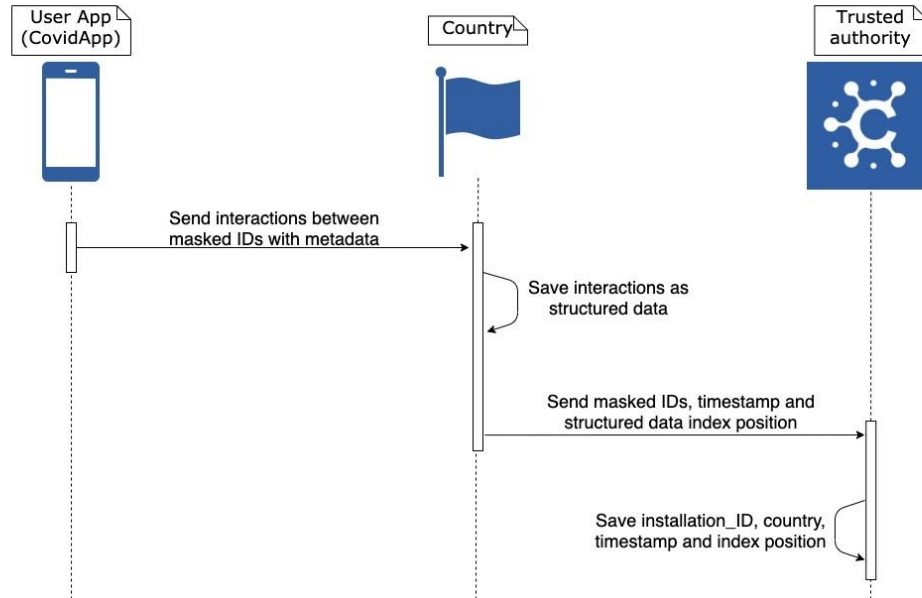
Email: coronavirus.outbreak.control@gmail.com

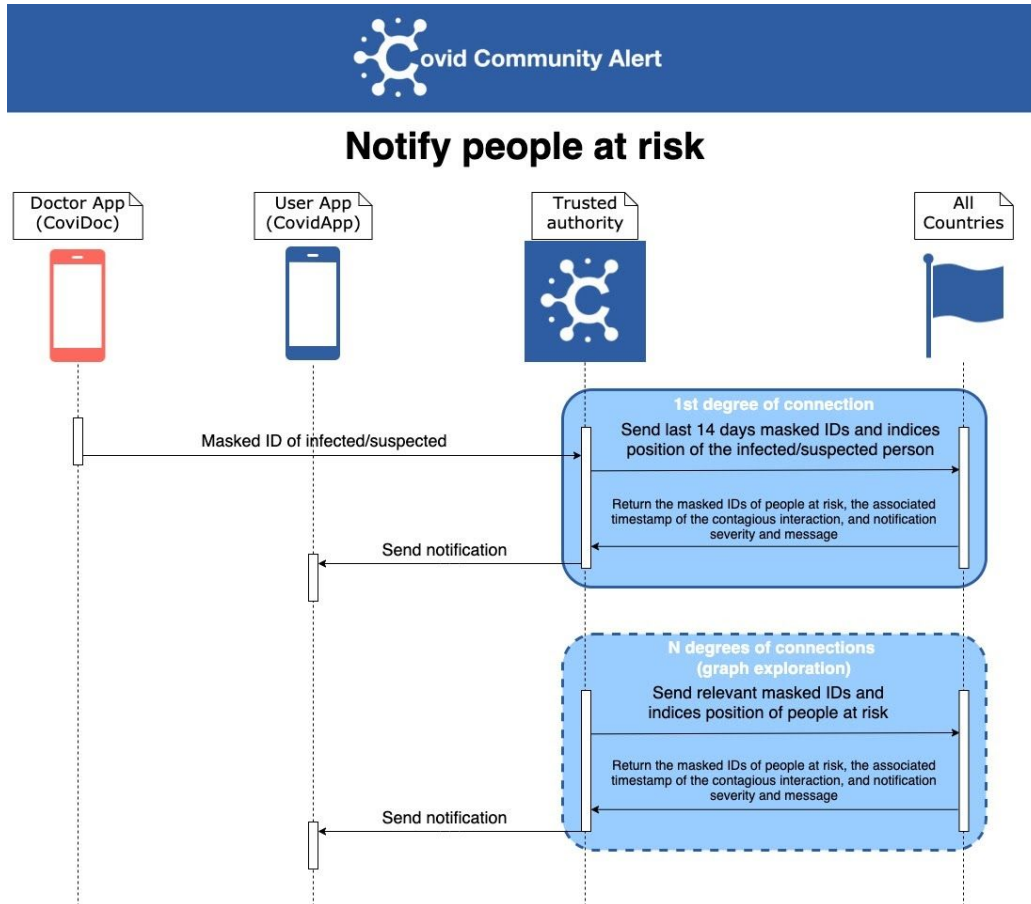
Website: <https://coronavirus-outbreak-control.github.io/web/>

Roaming Protocol



Monitor anonymous interactions





Collaboration & Support

We are working with recognised Health Institutions and Research Centers

We collaborate with:

- Stefano Quintarelli: Mentor, Associazione Copernicani, Member of the AI High Level Expert Group at European Commission and Chairman of the steering Committee of Agenzia per l'Italia Digitale (AGID)
- Bhaskar Krishnamachari, director of USC Viterbi Center for Cyber-Physical Systems and the Internet of Things and professor of Electrical Engineering and Computer Science Professor in the University of Southern California, LA
- Raffaele Perego, Director of Research, ISTI-CNR
- Vania Bogorny, Head of the Computer Science Graduate Program at Universidade Federal de Santa Catarina (UFSC) - Brazil

Covid Community Alert

Mobile: +44 (0) 7858 61 3934

Email: coronavirus.outbreak.control@gmail.com

Website: <https://coronavirus-outbreak-control.github.io/web/>

We are supported by:

- Stefano Fratepietro, Group Head of Cybersecurity - Chief Security Officer at Be Think, Solve, Execute S.p.A.
- Fabio Cassanelli & Emanuele Bartoli - Cyber Security specialists at Be Shaping the Future
- Marco Trombetti, founder at Pi-Campus
- Oreste Pollicino, Director of Bocconi LL.M. in Law of Internet Technology and full professor of Constitutional Law at Università Bocconi
- Stefano Leonardi, Full Professor at Sapienza University of Rome
- Sébastien Bratières, Director of AI, Translated
- Sebastian Filetti, STITCH (Sapienza information-based Technology InnovaTion Center for Health)
- Many more...

Covid Community Alert

Mobile: +44 (0) 7858 61 3934

Email: coronavirus.outbreak.control@gmail.com

Website: <https://coronavirus-outbreak-control.github.io/web/>