# PROPOSED SOLUTION

Figure 1 gives an overview of the different roles and their interactions. Six roles can be distinguished in the system. A catering facility can be a pub, a restaurant or a hotel. The catering facility relies on a QR code to support user registration. A user is each person that can visit catering facilities. A user needs to register when entering a catering facility and must be informed if an infected person was at the catering facility at the same time interval. The general practitioner examines people with symptoms and initiates the contact tracing procedure when a patient is diagnosed as ‘covid infected’. The other roles are explicitly defined to support contact tracing. The registrar has three major tasks. First, it enrolls new catering facilities and provides them with a tool to generate QR codes on a daily basis. Second, it enrolls new users and provides them with tokens that can be used when visiting a catering facility. Third, it reveals contact information of possibly infected people. The matching service keeps information about visits and supports contact tracing. Note that uniquely identifying user and catering facility data are not revealed to the matching service.

Besides those key roles, a mix proxy shuffles incoming messages (i.e. capsules - see further) and flushes them at regular time intervals to the matching service. Further, a central health authority (not included in figure 1) mediates interactions between the general practitioner and the matching service.

Five phases can be distinguished. During enrollment, users and catering units register to the system. The second phase includes the steps that are taken when a user visits a catering facility. Phase three focuses on the flow that is initiated when a patient is diagnosed as covid infected by a general practitioner. Next, we show how visitors are informed about their risk status (phase four). Finally, random spotchecks are discussed.

A picture containing text, map

Description automatically generated

Figure 1. Interaction flow between stakeholders.

## Enrollment

Both catering facilities and users need to enroll in the system.

**Catering Facility Enrollment.** The former interacts with a web service provided by the registrar. The latter generates a master secret key . Each catering facility exposes uniquely identifying information (e.g., unique business number, name and address of the facility, and/or phone number) and the authenticity thereof. The registrar then derives each a new secret key , based on a unique identifier of the catering facility that is calculated as follows (with KDF being a secure key derivation function):

.

Next, it calculates a day-specific pseudonym for the catering facility as follows (with H being a cryptographic hash function):

The pseudonym is sent to the catering facility ultimately at the start of that particular day. Note that for efficiency reasons, a set of pseudonyms can be sent in batch (e.g., at the beginning of each month). The application running at the catering facility CF then generates a random number and a new QR code on a daily basis that contains three values:

**User Enrollment.** An individual first downloads a covid-catering app from a trustworthy app store. After installation, the user needs to go through an enrollment phase which is bound to the phone number. That number can act as unique identifier and prevents users from enrolling many times. After successful enrollment, the registrar issues a set of signed tokens to each user. Each token can be spent ***exactly once*,** at a catering facility. Although other time windows can be used, our prototype issues ***48 tokens a day to each user***. Note that the registrar keeps the mapping between the phone number and the tokens that were issued. None of the other parties in the system is able to derive the identify of the user from a token.

## Visiting a Catering Facility

We assume that each pub, hotel or restaurant shows her QR code at a clearly visible location (e.g., at the entrance of the facility or attached to each table). The code is valid during one day. When a visitor enters the facility, she scans and logs the three values , and exposed by the QR code together with the entry time locally on the phone, for a specific number of days. The duration depends on the governmental directives and is related to the incubation time of the virus. The app then registers the visit by sending a *capsule* to the mixing server via a secure connection with server authentication activated. This capsule contains the current time interval, a valid user token and the 3rd value in the QR code (i.e., ). Note that and remain locally on the phone of the visitor. Upon receiving a capsule, the mixing server first checks (a) the validity of the user token, and then verifies that (b) it is a token for that particular and (c) it has not been spent before. If all three checks are successful, the mixing server signs and sends it back as confirmation to the app of the user. The app then transforms the bits of the signature to a visual representation. This can be a symbol with a specific color -- or a variant -- and the current time interval. As all customers receive the same signature, the facility manager can easily verify this visual representation of this signed acknowledgement (as the facility manager has also received the symbol when he registered that day as the first customer). As such, the facility manager is convinced that the app has really sent the third value and a user token to the mixing server.

The user can show the visual representation to the catering personnel when doing a first order (and optionally at subsequent orders). Depending on local directives either one individual per bubble or all individuals scan the code.

Every time the app sends a capsule to the mixing server (i.e. two times an hour in our prototype), the mixing server adds the time to this entry. This way, users cannot lie about the time he has visited that location. The user is prevented from delaying the transmission of an entry to the mixing server, as the catering facility demands a proof (i.e. the visual code) that the capsule was sent correctly to the mixing server.

The mixing server holds the capsules of all users during some time interval. Shortly after the time interval has finished, the capsules are flushed to the matching service in a random order. The data is removed from the database of the matching service after a predefined time interval, which can be imposed by the government (with a minimum of one day to prevent multiple spending of the same user token).

When a user leaves the catering facility, the smartphone logs the exit time. The user can push a button in the app when exiting, or the waitress forces the users to take an action. Alternatively, an app notification can be sent to the user when he leaves the facility. Geo-fencing can be used to detect users leaving a certain location.

## Registering an Infected User

When feeling sick, a user should visit a practitioner who can diagnose that a user is covid infected. If so, the practitioner reads out part of the users’ logs (i.e. and ) and the time intervals that were stored on the smartphone. He signs and forwards these via the central health authority to the matching service, possibly after shuffling them with tuples of other users. The matching service downloads all for a particular day from the registrar. Once the matching service knows all day-specific pseudonyms , it has all information required to control the validity of the data provided by the user by taking hashes of the registrar’s identifier . Hence, the latter cannot lie about his locations, as he cannot provide a valid for places he did not visit. As long as a user does not report about his locations, someone that did not scan the QR code cannot retrieve the location (as the location cannot be revealed without knowledge of ).

The matching server marks all entries that contain capsule data for that facility as ‘critical’, and marks the token of the infected user (that visited the general practitioner) as ‘informed’. All other tokens in the same entry are ‘uninformed’.

## Informing Possibly Infected Users

Once a day, each user (i.e. the smartphone app) fetches a list of fresh critical tuples . The app checks whether the tuples also appear in its local storage. This would mean that the user was at a particular location within the critical time interval. If so, the user picks the corresponding and sends these to the mixing proxy which further forwards it to the matching service. The matching service marks all confirmed tokens as ‘informed’. After a certain time interval (e.g., one day), the remaining tokens are forwarded to the *registrar*. The latter contacts all ‘uninformed’ people that were at the facility at the critical time interval.

## Random Spotchecks

To prevent that a catering facility cheats and does not show the right QR code to its customers, the catering facility registrar performs random spotchecks. Therefore, an agent of the registrar visits a catering facility and scans the QR code. As the registrar knows the pseudonym of each catering facility beforehand, it can immediately check the validity of the QR code when scanning it.