

# Entwurf und Analyse von Crowd-Sensing Mechanismen für mobile Corona-Warn-Applikationen

\*Note: Sub-titles are not captured in Xplore and should not be used

Gabriel Bonnet  
*Bsc. Science Softwaretechnik*  
*Universität Stuttgart*  
Stuttgart, Germany

Kai Braun  
*Bsc. Science Softwaretechnik*  
*Universität Stuttgart*  
Stuttgart, Germany

Hannes Deichmann  
*Bsc. Science Softwaretechnik*  
*Universität Stuttgart*  
Stuttgart, Germany

Timm Marquardt  
*Bsc. Science Informatik*  
*Universität Stuttgart*  
Stuttgart, Germany

**Betreuer:** Dr. rer. Nat. Frank Dürr  
*Institut für Parallele und Verteilte Systeme*  
*Universität Stuttgart*  
Stuttgart, Germany  
frank.duerr@ipvs.uni-stuttgart.de

**Prüfer:** Prof. Dr. rer. Nat. Kurt Rothermel  
*Institut für Parallele und Verteilte Systeme*  
*Universität Stuttgart*  
Stuttgart, Germany  
kurt.rothermel@ipvs.uni-stuttgart.de

**Zusammenfassung—TODO**  
**Index Terms—Corona, Corona-Warn-App, Density Map, Risikoberechnung**

## I. EINLEITUNG

## II. BESTEHENDE ARBEITEN

### A. Corona Lage

### B. Corona Warn App

Die deutsche "Corona Warn App"(CWA) wurde im Auftrag des Robert Koch Institutes von SAP, der deutschen Telekom und weiteren Unternehmen entwickelt. Ihr Hauptziel ist die automatisierte und anonyme Kontaktverfolgung mittels Smartphones, um möglichst viele Infektionsketten zu unterbrechen und die Ausbreitung des Coronavirus weitmöglichst zu verlangsamen. Zum aktuellen Zeitpunkt ist die App seit etwa einem dreiviertel Jahr aktiv und wurde knapp 30millionen Mal heruntergeladen (Stand Mai 2021).

Die Funktion der Corona Warn App ist relativ simpel. Ein Nutzer lädt sich die App auf sein Smartphone und aktiviert seine Bluetooth Funktion. Falls er nun mit einer weiteren Person, die ebenfalls die CWA installiert hat in Kontakt kommt, werden anonym Daten ausgetauscht und die Begegnung auf den jeweiligen Endgeräten gespeichert. Sofern man sich nun mit dem Coronavirus infiziert hat, erhält man von einer Teststelle zusammen mit dem positiven Testergebnis einen Code, welchem man in seine App eintragen kann. Alle Kontaktpersonen werden dementsprechend informiert, ohne zu wissen von wem diese Mitteilung letztendlich stammt. Die Corona Warn App unterscheidet prinzipiell zwischen zwei Risikostufen, die mithilfe eines grünen bzw. roten Bildschirms

auf der Startseite der App angezeigt werden. Ein grüner Bildschirm bedeutet hierbei keine/kaum Risikobegrenzungen und damit ein vermindertes Risiko, während ein roter Bildschirm ein erhöhtes Risiko anzeigt.

Der Corona Warn App zugrunde liegendes Feature ist das "Exposure Notification Framework", welches in Kooperation von Google und Apple für Corona Apps entwickelt wurde und das Bluetooth Low Energy (BLE) Feature nutzt, um die Begegnungsdaten der Smartphones anonym untereinander auszutauschen.

### *Exposure Notification Framework:*

Jeden tag wird vom Framework ein sogenannter Temporary Exposure Key (TEK) generiert. Dieser hält 24 Stunden und ist ein zufällig ausgewählter Schlüssel. Aus diesem TEK werden Rolling Proximity Identifiers (RPI) abgeleitet. Diese RPI's haben eine Lebensdauer von 10-15 Minuten, bevor sie gelöscht und ein neuer RPI aus dem TEK erstellt wird. Wichtig ist hierbei, dass sich aus den RPI's nicht der TEK ableiten lässt. Die RPI's werden nun unter den Smartphones im Falle einer Begegnung mithilfe von BLE ausgetauscht. Neben den RPI werden zudem verschlüsselt Informationen über die Begegnungsdauer, sowie dem ungefähren Abstand ausgetauscht. Falls sich nun eine Person mit dem Coronavirus infiziert, werden dessen TEK (nun Diagnosis Keys genannt) der letzten 14 Tage auf einen zentralen Server hochgeladen. Jedes Smartphone hat Zugriff auf diesen zentralen Server und kann die Diagnosis Keys aller infizierten Personen herunterladen. Mithilfe der Diagnosis Keys können dann die gespeicherten RPI's aus den Begegnungen abgeglichen und im Falle einer Übereinstimmung die Begegnungsdauer und der Abstand entschlüsselt werden. Aus diesen Informationen

bestimmt die Corona Warn App das Infektionsrisiko der Person.

### C. Bluetooth Low Energy und Density Maps

## III. LÖSUNGSVORSCHLÄGE

### A. Übersicht aller Erweiterungen

*Impfpass:* - In der aktuellen Situation ist es wichtig, nachweisen zu können, dass man geimpft wurde. Dies könnte man entweder dadurch regeln, einen analogen Impfnachweis immer bei sich zu tragen, oder durch einen digitalen Impfpass in der Corona-Warn-App.

Größter Vorteil eines in der App integrierten Nachweises wäre die Digitalisierung des Verwaltungswesens und dadurch eine Vereinfachung des Organisationsaufwands.

Anstatt mehrere Unterlagen mit sich herumzutragen, hat man diese stattdessen gesammelt in einer App.

Digitalisierung ist zudem ein positiv konnotierter Begriff, welcher einen Anreiz für die Bevölkerung schafft, die App zu nutzen. Dies ist immer ein relevanter Nutzen, da es das Ziel ist, die Corona-Warn-App von 100% der Bevölkerung nutzen zu lassen.

Implementiert man zusätzlich dazu eine Art Krankenakte, in der man Merkmale wie Gewicht und Vorerkrankungen einträgt, lässt sich dadurch die Impfpriorisierung und die Reihenfolge in der Bevölkerung besser organisieren.

Benötigte Daten für einen generellen Impfpass wären:

- Name
- Krankenkasse
- Durchgeführte Impfungen

Für die Einschätzung der Impfpriorisierung wäre zusätzlich erforderlich:

- Alter
- Vorerkrankungen
- Gewicht
- Beruf/Arbeitsumfeld

Bei einem unverbindlichen Impfpass wären diese Angaben nicht wirklich nutzbar außer für den Endnutzer, da man auf die Eigenverantwortung des einzelnen Nutzers angewiesen wäre, die Daten also einfach gefälscht werden können.

Stattdessen müsste man, um die Daten als sicher zu garantieren, die Bearbeitung des Impfpass beschränken, sodass nur offizielle Stellen wie Krankenhäuser oder Arztpraxen diesen bearbeiten können.

Somit wird Richtigkeit der Daten gewährt, welche dann für das Krankwesen genutzt werden können.

### *Gamification:*

Eine Möglichkeit, das Problem einer zu niedrigen Nutzung der App zu adressieren, ist Gamification. Dadurch versuchen wir, den Anreiz zu steigern, die App herunterzuladen und auch aktiv zu benutzen.

Gamification der App kann es in verschiedenen Varianten geben. Zum Beispiel lassen sich simple Spielmechaniken wie persönliche Highscores einbauen, welche man mit anderen Nutzern der App vergleichen kann. Dafür bieten

sich Statistiken an, die sich bereits aus der Nutzung der App erstellen lassen. Dazu zählen die getroffenen Personen an einem Tag, die durchschnittliche Dauer einer Begegnung oder der ermittelte Risikofaktor. Ziel der Highscores wäre es dann, diese Werte so möglich wie gering zu halten oder im Vergleich zu anderen Nutzern bessere Werte zu haben.

Vorstellbar sind auch daraus resultierende optische Veränderungen der App durch neue Farben oder ein anderes Design.

Dieses Feature richtet sich vermutlich an eine jüngere Zielgruppe, die mit solchen Mechaniken meist mehr vertraut ist als ältere Generationen.

Es könnte sogar sein, dass dieses Feature als Verharmlosung der ernstesten Thematik interpretiert wird, was in einem gegensätzlichen Effekt zur eigentlichen Intention resultieren würde.

Eine Alternative zu Highscores und anderen virtuellen Errungenschaften wäre eine Art Belohnung in der Realität, zum Beispiel durch eine in der App durchgeführte Lotterie.

Dies ist vermutlich attraktiver für alle Generationen, jedoch schwieriger umzusetzen, da hier Betrug oder mehrmalige Teilnahme verhindert werden muss. Vorstellbar wäre auch eine Verbindung mit den ermittelten Statistiken, sodass Leute mit guten Werten eine höhere Gewinnchance haben.

### B. Bewertungskriterien

Um einen möglichst einheitlichen Vergleich der Features zu garantieren, haben wir sechs Bereiche festgelegt, nach denen die Features beurteilt werden.

Diese Bereiche sind wie folgt aufgeteilt:

*Privatsphäre:* Wie stark greift das Feature in die Privatsphäre ein?

*Nutzen:* Welchen effektiven Nutzen bringt das Feature und lohnt sich der Aufwand für den effektiven Nutzen?

*Umsetzbarkeit:* Wie realistisch ist es, dieses Feature zu implementieren in Anbetracht von Datenschutz, Programmierung oder anderen Hindernissen?

*Akzeptanz:* Da wir eine möglichst hohe Nutzung der App in der Bevölkerung erzielen wollen, ist es wichtig, dass das Feature entweder einen Anreiz zur Nutzung schafft oder zumindest diesen nicht schwächt. Die Frage ist also, ob die neue Funktion Skepsis oder Missbilligung beim Nutzer auslöst.

*Sicherheit:* Wie gut lässt sich das implementierte Feature vor Missbrauch und Angriffen schützen?

*Daten:* Welche Daten werden mindestens benötigt, dass das Feature funktioniert?

### C. Bewertung Nebenfeatures

## IV. HAUPTFEATURES

### A. Erweiterte Risikoberechnung

1) *Familienfeature:*

2) *TRL update:*

3) *Roter Summand:*

#### 4) Grüner Summand:

##### B. Density-Map

Eine Density-Map zeigt an, wo sich bestimmte Datenpunkte auf einer Karte häufen. Häufig findet sie Anwendung bei Bevölkerungsaufzeichnungen. Im Sinne der erweiterten Corona Warn App möchten wir dieses Feature benutzen, um in nahezu Echtzeit die Besucherzahl an öffentlichen Orten oder Geschäften aufzuzeichnen und grafisch auf einer Karte darstellen. Damit kann eine Nutzer stark besuchte Gegenden identifizieren und meiden. Ein ähnliches Feature wird bereits vom Unternehmen Google angeboten. Wer in dessen Suchmaschine nach Geschäften in der Umgebung sucht kann neben den Öffnungszeiten meist auch die live Stoßzeiten und durchschnittliche Aufenthaltsdauer ansehen. Google verwendet hierfür jedoch personalisierte Daten seiner Nutzer, wie z.B. den Standort um diese Aufzeichnung vorzunehmen. Um die Sicherheit und Privatsphäre der Nutzer zu schützen schlagen wir deshalb eine alternative Lösung mithilfe des BLE-Features der Corona Warn App vor.

##### 1) Lösungsansatz:

Das zentrale Problem, mit dem wir uns beim Erstellen einer Density Map beschäftigen ist, wie die anonymen Daten gebündelt und auf einen zentralen Server geladen werden können, ohne die Sicherheits-/Privacy Aspekte signifikant einzuschränken/verletzen. Ein naiver Ansatz wäre beispielsweise, dass jedes Smartphone seine gesammelte Begegnungszahl auf einen zentralen Server hochlädt und dieser diese Daten auswertet. Jedoch müssten dafür neben der Standorte der einzelnen Smartphones auch die RPI's aller aufgezeichneten Geräte mit hochgeladen werden, um etwa eine doppelte Zählung bestimmter Endgeräte zu verhindern. Eine derartige Sammlung aller Kontaktdaten (auch wenn es sich bei diesen Daten um anonyme RPI's handelt) auf einem zentralen Server würde jedoch der grundsätzlichen Philosophie der Corona Warn App widersprechen. Diese verfolgte in ihrer Kontaktverfolgung einen möglichst dezentralen Ansatz, um die Privatsphäre der Nutzer zu garantieren. Dementsprechend suchen wir ebenfalls gezielt nach Verfahren, welche weitmöglichst eine dezentrale Lösung verfolgen. Hierfür haben wir uns für die Umsetzung 2 Verfahren überlegt, mit welchen wir das Problem angehen: Eine infrastrukturbasierte Lösung mittels **zentraler Masten**, welche in Echtzeit die BLE-Geräte in ihrem festen Empfangsgebiet aufzeichnen, sowie eine infrastrukturlose Lösung **ohne Masten**.

##### Infrastrukturbasierte Lösung mit Masten:

Die Idee zentraler Masten besteht darin, dass diese an stark besuchten Orten/Plätzen, wie z.B. Supermärkten oder Haltestellen, ihre Umgebung scannen und für ein gewisses Zeitintervall alle erfassten Smartphones aufzeichnen. Nach Ablauf des Zeitintervalls wird die Anzahl der Begegnungen auf einen zentralen Server hochgeladen. Anders als beim naiven Ansatz, bei dem der zentrale Server die Daten aller Smartphones erfassen würde, ist hier die Aufzeichnung auf

tausende bis zehntausende einzelne Masten verteilt. Dadurch gäbe es keinen *single point of failure*", bei dem beispielsweise die Standorte aller Smartphones preisgegeben werden würde. Da außerdem die Daten dezentral bei den Masten ausgewertet werden und dann erst auf einen Server hochgeladen, reduziert dies den Datenstrom und die Last des zentralen Servers erheblich.

Die Begegnungsaufzeichnung erfolgt analog zur Aufzeichnung mobiler Endgeräte untereinander per BLE. Somit sind in diesem Aspekt Sicherheit und Privatsphäre gleichermaßen gesichert, wie bei der bisherigen Corona Warn App. Außerdem sollen erst ab einem gewissen Threshold von Aufzeichnungen (z.B. 5 Personen) die Daten auf den Server hochgeladen werden, sowie lediglich die Anzahl der Begegnungen und nicht die einzelnen RPI's der Geräte. Diese Aggregation der Daten stellt sicher, dass keine einzelnen RPI's oder Personen über die Masten hinweg nachverfolgt und damit identifiziert werden können. Abb. 1 zeigt den groben Ablauf eines zentralen Masten.

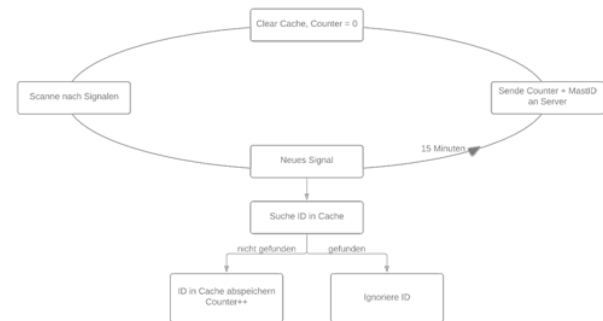


Abbildung 1. Funktionsweise eines zentralen Masten

Wie bereits erwähnt scannt ein Mast seine Umgebung nach Endgeräten und zeichnet alle Begegnungen innerhalb eines Zeitintervalls von etwa 15 Minuten (Lebensdauer eines RPI's) auf. Die Aufzeichnung erfolgt analog zur bisherigen Begegnungsaufzeichnung der Corona Warn App über BLE. Es werden demnach nur die RPI's ausgetauscht und keinerlei sensible Nutzerdaten. Nach der Aufzeichnung überprüft der Mast, ob es sich bei der Begegnung nicht um ein bereits bekanntes Gerät handelt und erhöht dementsprechend den Counter für die Begegnungszahl um eins. Nach Ablauf des 15 Minuten Intervalls lädt der Mast die gesammelte Anzahl an Begegnungen auf einen zentralen Server hoch, setzt seine Aufzeichnungsdaten zurück und fängt wieder von vorne mit seiner Begegnungsaufzeichnung an.

Da ein Mast unter Umständen einen größeren Bereich abdecken muss als ein normaler Nutzer, sollte das Signal (z.B. mit Repeatern) unter Umständen verstärkt werden, falls dies mit BLE möglich. Andernfalls müsste ein Ort in mehrere Bereiche unterteilt werden, die dann von einzelnen Masten aufgezeichnet werden können.

Hier wird auch deutlich, dass sich die Aufzeichnungsradien der einzelnen Masten möglichst nicht überschneiden sollte,

da sonst ein Smartphone von mehreren Masten gleichzeitig aufgenommen wird. Eine mögliche Lösung für dieses Problem wäre, die BLE Signale der Masten mit bestimmten Tags zu versehen, sodass ein Smartphone erkennt, falls es sich im Radius von mehreren Masten aufhält und dementsprechend nur einem Masten seine Daten sendet.

Da die Installation zentraler Masten an jedem belebteren Platz ein aufwändiges und kostspieliges Unterfangen sein kann schlagen wir vor, dass registrierte/zertifizierte mobile Endgeräte ebenfalls als zentraler Mast fungieren können. Hierfür müssen jedoch Einschränkungen gelten, sodass die Funktion als zentraler Mast auf beispielsweise einem Smartphone nicht missbraucht wird. Folgende Regeln und Restriktionen sollten für alle Masten gelten:

- Masten können selbständig von Dritten erworben oder im Falle eines Smartphones, welches als zentraler Mast fungieren soll, die App installiert werden. Jedoch muss vor der Nutzung des Features das Gerät von einer offiziellen Behörde autorisiert werden. Damit darf dann etwa ein Restaurantbetreiber, über eine zentrale Behörde registriert, seine Einrichtung mit seinem Smartphone aufzeichnen.
- Zentrale Masten müssen während ihrer Nutzung dauerhaft ihren Standort aktivieren und können nur im Umfeld ihres registrierten Bereiches (z.B. Ort des Restaurants) als Mast fungieren. Verlässt ein Mast den für ihn zugewiesenen Aufzeichnungsbereich sollt dieser seine Aufzeichnung automatisch stoppen. Dies verhindert, dass das Feature für unerlaubte Zwecke benutzt wird.
- Verletzung der Eigenschaften sollte von anderen Nutzern gemeldet werden können. Beispiele für Verletzung der Eigenschaften wäre etwa eine Person, die in einem unerlaubten Gebiet (z.B. Wohngebiet) einen Masten betreibt oder ein Mast, der viele Begegnungen aufzeichnet, obwohl sich keine Person im Umkreis befindet.

Masten, die ihre Begegnungen aufgezeichnet haben, laden diese auf einen zentralen Server hoch. Dieser verarbeitet die gesammelten Daten aller Masten und erstellt die Density Map, welche dann vom Nutzer auf seinem Endgerät abgerufen werden kann. Abb. 2 zeigt einen grafischen Ablauf zu Erstellung der Density Map mittels zentraler Masten.

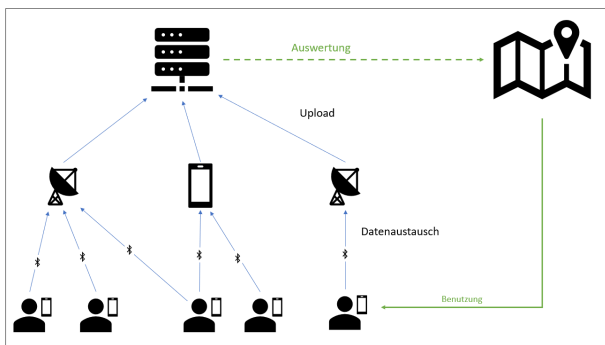


Abbildung 2. Übersicht: Erstellen der Density Map

Wie in Kapitel 2 beschrieben besitzen die RPI's eine Lebensdauer von etwa 15 Minuten. Dementsprechend muss beachtet werden, dass Smartphones auch während des Aufenthaltes im Umkreis eines Masten ihren RPI regelmäßig ändern, wodurch ein Endgerät vom Mast als zwei unterschiedliche Begegnungen erkannt werden kann. Der Mast zeichnet dementsprechend zu viele Begegnungen auf und muss seine gemessene Anzahl an Begegnungen nach unten korrigieren, um die tatsächliche Begegnungsanzahl herauszufinden. Wie stark der Wert nach unten korrigiert werden muss hängt hierbei von der durchschnittlichen Verweildauer eines Smartphones am Masten ab. Beträgt die durchschnittliche Verweildauer beispielsweise 15 Minuten (was dem Maximalwert entspricht, da sich nach 15min. der Mast zurücksetzt) wird innerhalb der Aufenthaltsdauer am Masten jedes Endgerät auf jeden fall einmal seinen RPI ändern und als neue Begegnung vom Masten aufgezeichnet werden. In diesem Fall würde der Mast doppelt so viele Begegnungen aufzeichnen als er in Realität. Ist die durchschnittliche Verweildauer hingegen kurz im Verhältnis zum Änderungsintervall der RPI's (z.B. 3 Minuten) ist die Wahrscheinlichkeit, dass sich in diesem Zeitintervall der RPI eines Smartphones ändert sehr klein, was weniger mehrfache Aufzeichnungen zur Folge hat. Wie stark ein aufgezeichneter Wert nach unten korrigiert werden sollte wird mittels folgender Formel errechnet:

$$IDs = \frac{RPIs}{(1+p)}$$

Eine Übersicht infwiefern der gemessene Wert mithilfe der Formel korrigiert wird in Abb. 3 aufgezeichnet. Auf der x-Achse befindet sich die durchschnittliche Verweildauer der Smartphones in Mastreichweite in Minuten, während die y-Achse die Anzahl an Smartphones (ID's) angibt. Wir gehen davon aus, dass der Mast in jedem Fall genau 100 ID's aufgezeichnet hat (blauer Graph). Je nach Verweildauer muss der gemessene Wert nun mittels Formel nach unten korrigiert werden (orangener Graph).

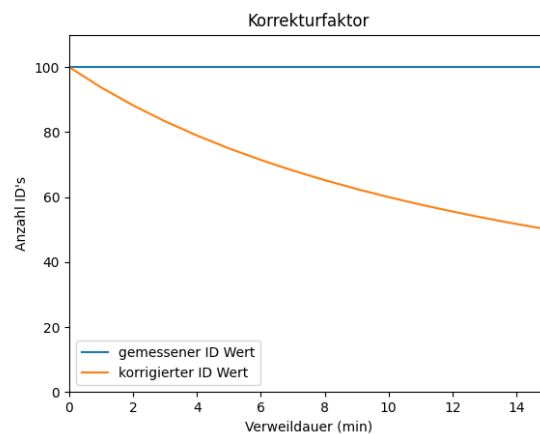


Abbildung 3. Korrekturfaktor in Abhängigkeit der durchschnittlichen Verweildauer

### Ohne Masten:

Beim Ansatz ohne zentrale Masten besteht die Idee darin, dass jedes Smartphone als „mobiler Mast“ arbeitet und die Funktion eines Masten übernehmen kann. Auch bei diesem Ansatz möchten wir zunächst die Daten dezentral bündeln und aggregieren, um den bereits erwähnten Privatsphäre- und Sicherheitsrisiken des naiven Ansatzes entgegenzuwirken.

Begegnen sich zwei Smartphones tauschen diese, wie bisher in der aktuellen Corona Warn App, ihre RPI's und Daten untereinander aus. Zusätzlich wird nun eines der beiden Geräte als Mast ausgewählt (Abb. 4). Dieses zeichnet die Begegnung der beiden untereinander auf und übernimmt alle bisherigen Aufzeichnungen des anderen Gerätes in diesem Zeitintervall. Die Aufzeichnungsdaten werden also transitiv bis Ende des Intervalls weitergegeben. Nach Ablauf des Zeitintervalls lädt das Smartphone, welches als mobiler Mast ausgewählt wurde, die aggregierte Begegnungsanzahl mit seinem aktuellen Standort auf den zentralen Server.

Bei dieser Methode können natürlich nur Smartphones, welche ihren Standort aktiv haben, als mobiler Mast arbeiten. Im besten Fall hat jedes Handy diese Funktion aktiv. Smartphones, die ihren Standort nicht aktiv haben, werden können von anderen Smartphones mit aktiven Standort aufgezeichnet, jedoch selbst nie als mobiler Mast ausgewählt werden.

Anders als bei zentralen Masten kann nun an jedem Ort, an dem sich ein Smartphone mit der Corona Warn App befindet eine Aufzeichnung für die Density-Map erfolgen. Die Datenerfassung der Density-Map ist somit nicht mehr von vorinstallierten zentralen Masten abhängig.

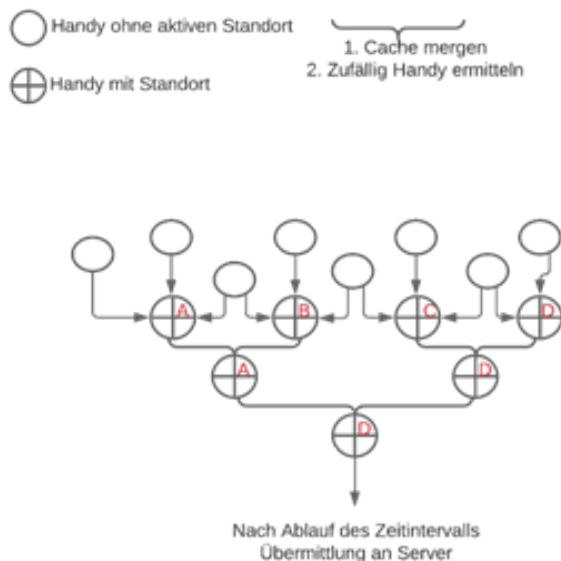


Abbildung 4. Smartphones als Mobile Masten

Beim mobilen Ansatz können diverse Probleme auftreten, welche gelöst werden müssen: Zum Beispiel können zwei

Smartphones A und B ihre gesamten Daten immer an das jeweilige andere Handy überreichen, bevor ihr 15 Minuten Intervall abgelaufen ist. Somit bleiben die gesammelten Daten für unendlich lange Zeit lokal auf A oder B gespeichert, ohne dass sie jemals hochgeladen werden. Eine Lösung für dieses Problem wäre immer das Smartphone als mobilen Mast auszuwählen, das näher am „Reset“, also am Ablauf des 15-Minuten Intervalls ist.

Ein weiteres Problem bei mobilen Masten besteht darin, dass sie sich innerhalb des Zeitintervalls über längere Strecken bewegen können während sie ihre Begegnungen aufzeichnen. In diesem Fall entsprechen Ort der Aufzeichnung eines anderen Gerätes und Ort des Hochladens der Daten nach Ablauf des Zeitintervalls nicht überein und das Bild der Density-Map wird verzerrt. Dieses Problem wird teilweise bereits durch die Corona App selbst gelöst, welche Begegnungen erst dann aufzeichnet, sobald sich Geräte über etwas längere Zeit in Kontakt stehen. Bei sich schnell bewegenden Smartphones müssten sich demnach anderer Smartphones mit gleicher Geschwindigkeit in dieselbe Richtung bewegen, damit eine Begegnung aufgezeichnet wird. Das Verkürzen des Zeitintervalls für Uploads auf beispielsweise 5 Minuten würde diesem Problem ebenfalls entgegenwirken, da in kürzeren Intervallen die Smartphones sich im Schnitt weniger weit bewegen können.

Eine weitere Lösung, die wir diesbezüglich vorschlagen ist ein sogenannter „Discount Faktor“. Dieser gewichtet indirekte Verbindungen eines mobilen Masten schwächer in die Auswertung der Density Map.

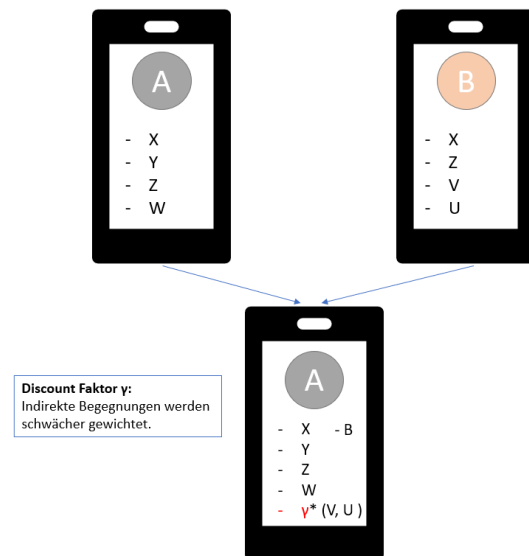


Abbildung 5. Discount Faktor für Mobile Masten

In Abbildung 5 begegnen sich die beiden Smartphones A und B. A hat bereits die RPI's X, Y, Z und W aufgezeichnet und B die RPI's X, Z, V, U. Nun tauschen die beiden Smartphones ihre RPI's gemäß funktionsweise der Corona Warn App aus. Außerdem wird A als mobiler Mast ausgewählt und übernimmt die Aufzeichnungen von B für die Density-

Map. Die RPI's  $V$  und  $U$  wurden nur von  $B$  aufgezeichnet. Dies bedeutet, dass sie sich nicht im direkten Umkreis von  $A$  befinden können.  $B$  muss sich vor der Begegnung mit  $A$  also an einem anderen Ort befunden haben und sich zu  $A$  bewegt haben (aus Sicht von  $A$ ). Dementsprechend werden die Begegnungen RPI's  $U$  und  $V$  mit dem Discount Faktor  $\gamma$  versehen. Der Discount Faktor kann auf verschiedene Arten realisiert werden. Er könnte beispielsweise einen Zahlenwert im Intervall  $(0, 1)$  annehmen. Somit würden indirekte Begegnungen mit dem Faktor multipliziert und nicht voll gewertet werden. Im Beispiel wären die zwei indirekten Begegnungen  $U$  und  $V$  mit gewähltem  $\gamma = 0.6$  versehen und nur als  $2 \cdot 0.6 = 1.2$  Begegnungen aufgezeichnet werden. Eine andere Lösung wäre beim Auswerten der Density-Map indirekten Begegnungen mit anderen Standorten zu versehen. Während alle direkten Begegnungen von  $A$  im unmittelbaren Umkreis seines Standortes auf der Map angezeigt werden  $U$  und  $V$  (beim hochladen aggregiert in 2 Begegnungen) in einem größeren Radius um  $A$  angezeigt. In diesem Fall wäre der Discount Faktor nur eine Markierung und müsste keinen numerischen Wert erhalten.

Der Discount Faktor kann ebenfalls transitiv weitergegeben werden. Sollten etwa die beiden indirekten Begegnungen  $U$  und  $V$  an ein weiteres Smartphone  $C$  übergeben werden, so würden sie bei  $C$  doppelt mit dem Discount Faktor versehen werden.

#### Hybridlösung:

Um die Effekte beider Ansätze nutzen zu können und die Effizienz weitmöglichst zu erhöhen schlagen wir eine Hybridlösung aus zentralen und mobilen Masten vor. Die Umsetzung hierfür ist relativ simpel. Wie oben beschrieben werden an häufig besuchten Orten und Plätzen zentrale Masten aufgestellt, um Daten für die Density-Map zu sammeln. Überall dort, wo jedoch keine solchen zentralen Masten existieren, greift der dezentrale Ansatz und die Mobiltelefone fungieren als mobile Masten. Der im infrastrukturbasierenden Ansatz beschriebene spezielle Tag für RPI's der Masten kann hier verwendet werden, damit ein Smartphone erkennt, ob er sich zurzeit im Umkreis eines zentralen Masten befindet oder nicht. Mittels Hybridlösung könnte man die Vorteile beider Ansätze vereinen und die Nachteile ausgleichen. Zentrale Masten können zuverlässiger einen festen Ort aufzeichnen, da für diese Funktion optimiert sind und sich dauerhaft an diesem Ort befinden. Mobile Masten hingegen können an jedem beliebigen Ort eine Ansammlung vieler Smartphones und damit vieler Menschen erkennen und auch eine Bewegung großer Menschenmassen aufzeichnen. Sie sind demnach flexibler, was den Standort angeht.

Das letzte zentrale Problem, was nicht nur unsere Lösungsansätze sondern auch die Corona Warn App im allgemeinen stark betrifft ist, dass nicht jede Person ein Smartphone besitzt und die Corona Warn App auf seinem Smartphone installiert hat. Dementsprechend liegen

die aufgezeichneten Werte durch die Masten stets unter der tatsächlichen Anzahl der Personen, die sich am entsprechenden Ort befinden. Laut aktuellem Stand (Mai 2021) benutzen etwa 27,8mio (34%). Unter der Annahme, dass die Nutzer der App auf die Standorte bezogen gleichverteilt sind bedeutet dies, dass man die gemessene Anzahl etwa verdreifachen muss, um die tatsächliche Anzahl an Personen zu erhalten.

#### 2) Simulation:

Um die im Lösungsansatz erarbeiteten theoretischen Überlegungen zu verifizieren und deren Umsetzbarkeit nachzuweisen haben wir eine Simulation der Masten programmiert. Eine reale Umsetzung des Features mit echten Masten und Smartphones erwies sich als zu umfangreich und im Rahmen unseres Forschungsprojektes zeitlich nicht umsetzbar. Die Simulation beschränkt sich auf die Umsetzung mittels zentraler Masten. Als Grundlage diente uns hierfür die Simulation "COVID-19 spread simulator" von Miguel Ángel Durán [1].

#### Motivation:

Unser Ansatz der festen Masten basiert darauf, Handys beim Eintritt in einen Mastbereich zu registrieren und ihren RPI zu speichern.

Dadurch, dass der RPI periodisch wechselt, werden zwangsläufig Handys doppelt registriert, sollten sie sich zum Zeitpunkt des Wechsels in der Reichweite des Mastes befinden.

Die gemessene Anzahl an Handys weicht somit immer zu einem bestimmten Teil von der realen Menge ab. Diesen Wert korrigieren wir nach unten, abhängig von der durchschnittlichen Verweildauer.

Die Wahrscheinlichkeit  $p$  einer doppelten Registrierung ist definiert durch

$$p = t_V / t_Z$$

wobei  $t_Z$  die Dauer eines Zyklus ist und  $t_V$  die durchschnittliche Verweildauer während dieses Zyklus. Die Berechnung der Verweildauer wird in Abschnitt "Aufbau" beschrieben. Da doppelt-registrierte Handys einfach gezählt werden sollen, muss die Hälfte des ermittelten Anteils trotzdem gezählt werden, womit sich die folgende Formel ergibt:

$$\text{Korrigierte Anzahl} = \text{Gemessene Anzahl} * (1 - p/2)$$

Abb. 6 zeigt den durch diese Formel berechneten Unterschied zwischen dem realen und gemessenen Wert, abhängig von der durchschnittlichen Verweildauer.

Das Ziel der Simulation ist es, festzustellen, wie nah der auf diese Weise berichtete Wert an der tatsächlichen Zahl von Handys liegt.

#### Aufbau:

Um die Formel zu verifizieren, modifizieren wir die Simulation "COVID-19 spread simulator".

Es werden 1000 Handys simuliert, die als Kreise mit zufälliger Bewegung und Position über ein festgelegtes Gebiet wandern. Auf diesem Gebiet sind 4 Masten, dargestellt durch größere Kreise, welche permanent speichern, welche Kreise in ihrer

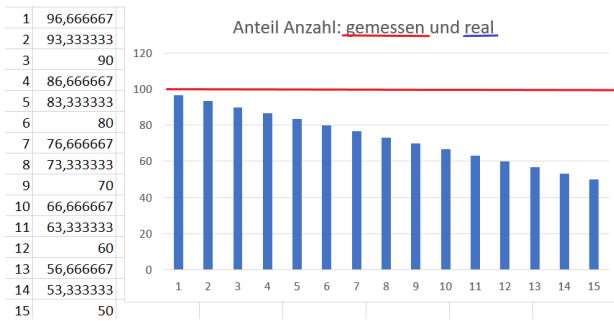


Abbildung 6. Anteil realer Wert am gemessen, abhängig von der durchschnittlichen Verweildauer

Reichweite liegen und wie lange dies der Fall ist.

Jeder Kreis hat eine sich periodisch verändernde, temporäre ID (RPI) und eine individuelle, permanente ID. Dies ermöglicht es uns, das Zählen von RPI's zu simulieren und gleichzeitig die korrekte Anzahl zu wissen.

Die durchschnittliche Verweildauer wird immer aktuell berechnet nach Ablauf des Zyklus eines Mastes. Dazu wird während des Zyklus mit jeder ID eine Zeitdauer abgespeichert und aus diesem der Durchschnitt ermittelt. Zur Laufzeit der Simulation speichern die Masten eintretende Kreise neu ab und zählen ab diesem Zeitpunkt einen Counter hoch. Am Ende eines Zyklus werden diese Counter addiert und durch die gemessene Anzahl der Kreise geteilt.

Je höher die durchschnittliche Verweildauer  $t_V$  des Zyklus ist, desto höher ist die Wahrscheinlichkeit  $p$ , dass ein einziges Handy doppelt registriert wird.

Zu Verifizierungszwecken haben wir die durchschnittliche Verweildauer mit Hilfe der permanenten IDs berechnet, was einen genaueren Wert liefert. Benutzt man stattdessen die RPIs, verringert sich die durchschnittliche Verweildauer und der korrigierte Wert wird ungenauer. Dies entspricht jedoch der Anwendung in der Praxis, da aus Datenschutzgründen nicht die IDs und ihre Verweildauer gemessen werden.

Die Simulation bietet zudem die Möglichkeit, Social Distancing durchzuführen. Dies führt dazu, dass ein Großteil der Kreise keine Geschwindigkeit mehr haben, und nur einige wenige sich überhaupt bewegen.

**Ergebnis:** Die gesammelten Daten belegen, dass die durch die Formel berechnete Anzahl (orangener Graph) wesentlich näher an der tatsächlichen Zahl (blauer Graph) liegt als die gemessene Menge (grüner Graph). Es fällt jedoch auf, dass in fast allen Fällen die berechnete Zahl immer noch größer ist als die tatsächliche. Nur in einzelnen Fällen wird die Zahl zu niedrig abgeschätzt.

Abb. 8 stellt die gesammelten Daten in einem Schaubild dar, ohne das Social Distancing benutzt wird.

Hierbei ist der rote Graph die durchschnittliche Verweildauer, der grüne Graph der gemessener Wert, der orangene Graph der korrigierte Wert und der blaue Graph der reale Wert.

Mit Social Distancing (Abb. 9) ist die durchschnittliche Verweildauer wesentlich höher und der angegliche Wert näher

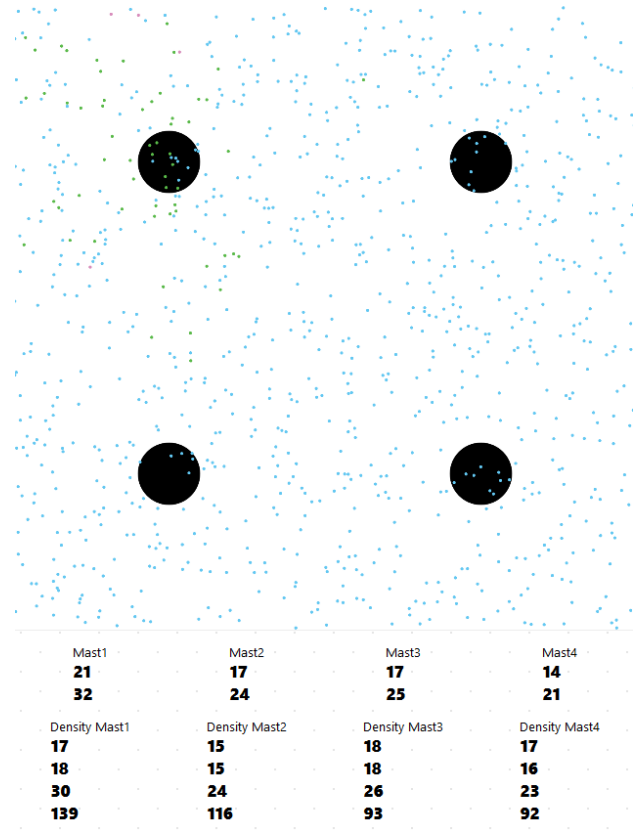


Abbildung 7. Darstellung der durchgeführten Simulation

an dem gemessen Wert liegt als in Abb. 8.

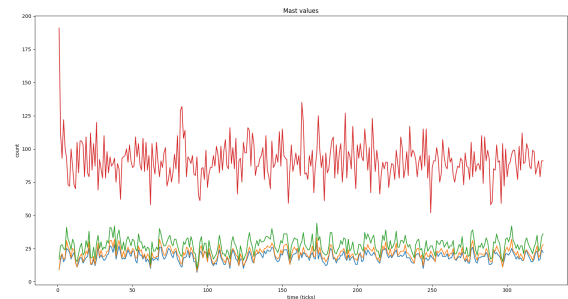


Abbildung 8. Ergebnisse ohne Social Distancing

Hinzu kommt die Beobachtung, dass je höher die durchschnittliche Verweildauer, desto geringer die Differenz zwischen korrigierten und realen Wert. Die Genauigkeit unserer Abschätzung wird demnach besser, je höher die Verweildauer ist.

**Schlussfolgerung:** Aus dem Ergebnis lässt sich die Erkenntnis gewinnen, dass die bisherige Formel nicht optimal ist. Daher haben wir einen neuen Ansatz zur Berechnung gewählt.

Die Wahrscheinlichkeit  $p$  einer doppelten Registrierung bleibt





Abbildung 9. Ergebnisse mit Social Distancing

gleich. Wir gehen davon aus, dass sich mit dieser und der korrekten Anzahl der IDs eines Zyklus die gemessene Anzahl berechnen lässt:

$$\text{RPIs} = \text{IDs} * (1 + p)$$

Nach Umformung ergibt sich somit eine neue Formel zur Angleichung unseres gemessenen Wertes:

$$\text{IDs} = \text{RPIs} / (1+p)$$

*Simulation neuer Formel:* Um die neue Formel zu testen, benutzen wir dieselbe Simulation wie zuvor, mit der Berechnung des korrigierten Wertes als einzigen Unterschied. Mit Hilfe der neuen Formel ergeben sich folgende Werte:

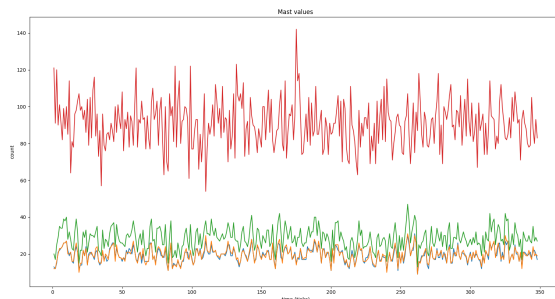


Abbildung 10. Neue Ergebnisse ohne Social Distancing



Abbildung 11. Neue Ergebnisse mit Social Distancing

### C. Bewertung der Hauptfeatures

## V. ZUSAMMENFASSUNG UND AUSBLICK

### DANKSAGUNG

Dank schön an meine Mama und mein Papa und mein Hund Bello, der immer so lieb mit mir gekuschelt hat! Danke an unseren Sponsor RAID: Shadow Legends und NordVPN! [?]

### LITERATUR

- [1] Miguel Ángel Durán, COVID-19 spread simulator, <https://github.com/midudev/covid-19-spread-simulator>