

Die offenen Themen finden Sie ab Seite 2. Hier einige einleitende Überlegungen.

Der Studienschwerpunkt mobile Anwendungen ist Teil einer Hochschule der **angewandten Wissenschaften**. Wir arbeiten an der Lösung von Problemen, die für uns (ja, ich meine Sie :)) sichtbar sind, oft aber nicht im Massenmarkt. Alle Innovationen fangen so an. Leider sind spektakuläre Ideen anfangs nicht von dummen zu unterscheiden. Das ist Teil des Spiels. Als Ingenieur und angewandter Wissenschaftler interessieren mich Probleme für die es nicht die offensichtliche Lösung gibt. Mich interessiert primär nicht, ob es einen Markt für eine Lösung gibt. Wir wissen nicht, ob wir gerade die nächste Smartphone-Revolution oder das nächsten Cargolifter-Projekt aufsetzen. Im Nachhinein ist man klug. Eins aber weiß ich genau: Der sichere Weg, keine Innovationen zu finden besteht darin, es so zu machen wie es alle machen. Deshalb machen wir, was wir spannend finden.

Ich betreue **Bachelor-, Masterarbeiten, Independent Coursework und Forschungsprojekte**. Die folgenden Themen lassen sich in allem Modulen einsetzen – man muss sie manchmal ein wenig adaptieren. Die folgenden Projekte beschäftigen mich derzeit sehr.

Projekt ASAP / Shark - SharedKnowledge

ASAP / Shark ist ein Framework zum Bau mobiler dezentraler Anwendungen. ASAP Apps sind mobile P2P System. Mehr Infos finden Sie in der Mediathek

(<https://mediathek.htw-berlin.de/album/view/aid/278>) und im Wiki auf Github

(<https://github.com/SharedKnowledge/ASAPJava/wiki>)

Projekt OHDM

Sie kennen Open Street Maps (<http://osm.org>). Es gibt einige Adaptionen, eine Karte spezialisiert für Fahrräder, eine für die Seefahrt. Wir bauen die historische Variante für OSM, siehe

<https://github.com/OpenHistoricalDataMap/OHDM-Documentation/wiki> und

<https://github.com/OpenHistoricalDataMap/OHDMConverter/wiki>

Projekt Hedwig

Drohnen kennen Sie. Wir wollen Drohnen programmierbar machen, so dass man mit Ihnen arbeiten kann wie mit einem Smartphone. Dazu arbeiten wir an eigener Drohnenhardware aber vor allem an einen Drohnenbetriebssystem. Wir unterstützen dabei Logistikprojekte und -unternehmen. Wir wollen es schaffen, dass sich eine unserer Drohne so einsetzbar ist wie die Eulen aus Harry Potter. Wir verstehen Drohneschwärme als mobile P2P Systeme. Damit hat das Projekt Schnittstellen zu ASAP/Shark.

Offene Themen

Inhaltsverzeichnis

Projekt ASAP / Shark - SharedKnowledge.....	1
Projekt OHDM.....	1
Projekt Hedwig.....	1
Offene Themen.....	1

Projekt Hedwig.....	2
App für Routing realer Pakete (ideal: zwei parallele Arbeiten).....	2
Bau einer flugfähigen Drohne.....	3
Projekt ASAP.....	3
Live-Tracker App als SharkComponent.....	4
Gruppenbasierte Verschlüsselung für ASAP.....	4
Service Discovery in Ad-hoc Netzwerken mit Android.....	5
Huge Bluetooth Scatternet.....	5
Bewegungsprofil für ortsbasiertes Routing.....	7
Corona - Warn-App.....	8
Programmieren mit Tor.....	8
GUI für SharkNet2.....	9
Dezentrale Personenprofile.....	9
Einfache Gestenerkennung für Nachrichtenaustausch.....	10
Projekt OHDM.....	12
Rendern historischer Kartendaten.....	12
Import historische Geometrien.....	12



Projekt Hedwig

App für Routing realer Pakete (ideal: zwei parallele Arbeiten)

Wir sind in Kontakt zu einem lokalen Logistik-Startup. Dort gibt es folgenden Plan: Pakete sollen in mehrere Etappen von einem Start- zu einem Zielort transportiert werden. Als ITler:innen kennen wir das Thema. Es nennt sich Routing. Es zunächst um die Lösung eines wesentlichen aber entscheidenden Details: Die Übergabe eines Paketes. Routing besteht immer darin, dass zwei Entität (im digitalen: z.B. IP-Router, in der Reallwelt: z.B. Bote/Botin, Drohne) ein Paket übergeben. Das passiert auf einer Route meistens mehrfach.

In der Realität muss diese Übergabe rechtssicher erfolgen. Übersetzt in IT heißt das: Wir müssen ein Verfahren entwickeln, mit dem die empfangende Einheit den Empfang nicht abstreiten kann und

die übergebende Einheit nachweisen kann, dass die Übergabe erfolgte.

Die kryptographischen Verfahren (siehe auch

<https://mediathek.htw-berlin.de/album/video/SEPrinzipienKryptografie/20ac30ae9d0301ebc1e5c66b9c33bf18/253>)

sind nicht sonderlich komplex. Die zweite Randbedingung ist diese: Das Verfahren soll auch ohne Internetverbindung möglich sein. Das ergibt sich einfach aus der Realität. Es wäre fatal, wenn eine Paketübergabe an einem Funkloch scheitert. Das darf nicht sein, weshalb Blockchain Lösungen ausfallen und insgesamt auch zu aufwendig wären. Glücklicherweise existiert bereits lauffähige Lösungen dezentrale für eine ähnliche Aufgabe, die ohne Internet-Access auskommt: digitales Geld (<https://github.com/SharedKnowledge/SharkCreditMoney/blob/master/src/net/sharksystem/creditmoney/SharkCreditMoneyComponent.java>)

Im Rahmen dieser Arbeit soll ein lauffähiger Prototyp entstehen. Er soll auf einem Handy laufen. Wir wollen also zunächst menschlichen Boten/Botinnen adressieren. Der Code soll aber so geschrieben werden, dass er ohne wesentliche Aufwände in einer Drohne laufen kann. Das klingt weniger dramatisch als es ist, siehe u.a.

<https://mediathek.htw-berlin.de/album/video/Mobile-P2P-Apps-entwickeln-mit-ASAP-Shark-Grundlagen-ASAPJava-SharkPeer/a0bd28f604bfd6cd0bb3d28329bdcc75/278>

Der Bau dieser Software bietet ausreichend Stoff für eine Abschlussarbeit.

Es besteht zusätzlich die Aufgabe, ein nutzbares GUI zu entwickeln. Das soll in enger Abstimmung mit dem Logistik-Unternehmen erfolgen. Hier wartet Entwicklungsarbeit in Android und – wenn Sie mögen - auch Webentwicklung auf Sie. Dieser zweite Teil ist ausreichend groß für eine zweite Arbeit. Diese beiden Arbeit können parallel ausgeführt werden. Sie beeinflussen sich nicht, d.h. wenn die eine schlecht voran kommt, hat das keinen negativen Einfluss auf die andere. Ideal wäre natürlich, dass beide erfolgreich arbeiten und am Ende ein lauffähiger Prototyp entsteht. Bevor ich den zweiten Teil vererbe, muss aber der erste vergeben sein. Alles andere ist wenig sinnvoll.

Bau einer flugfähigen Drohne

Es gibt eine Reihe von Anbietern von Drohnen. Das Problem mit allen: Sie sind nicht programmierbar. Drohnen lassen sich aber auch selber bauen. Die einzelnen Bestandteile findet man in jedem gut sortierten Elektroladen. Ein Kommilitone von Ihnen baute vor ca. 5 Jahren einmal vier flugfähige Drohnen. Es geht. Wir unterstützen Sie dabei natürlich stark.

Es soll eine flugfähige Drohne entstehen. In Ihrer Arbeit beschreiben Sie möglichst ausführlich welche Alternativen es bei den einzelnen Komponenten gab.

Projekt ASAP

Wir sind – Stand September 2022 – in einer interessanten¹ Situation. Ein Blick in die [Technology Readiness Level](#) ist hilfreich. An einer Hochschule kann man gut TRL 4 erreichen. Dort sind wir mit dem Kern des Frameworks [ASAPJava](#). Das funktioniert – mehrfach gezeigt im Labor stabil. Ja, alles geht natürlich besser, die Doku usw. usf. Aus Software-Produktsicht: Wir haben dort eine Alpha-Version erreicht – kann man etwa so übersetzen: Wir wissen, dass die Funktionen und

1 das ist *akademisch* für: enorm frustrierend

kontrollierten Szenarien funktionieren. Gutwillige Entwickler:innen können damit arbeiten. Kleine (nicht funktionale) Fehler erkennt man an vielen Stellen. Wir wissen auch nicht wie sich das System funktional in extremeren Umgebungen verhält. Das ist der Weg zur Beta. Man braucht in der Phase gutwillige Tester:innen, die ihr Ziel im freundlichen Entdecken und Dokumentieren von Bugs sehen, die es eindeutig gibt. Wir brauchen Stabilität.

Eine Beta-Version entspräche TRL 5-6 – Übersetzung etwa: Wir sagen einfach mal, dass das alles bestens läuft – beweist mal das Gegenteil. Man braucht in der Phase Tester:innen, die das System wirklich runter reißen wollen und dafür viel Engagement aufbringen. Beta-Tester:innen müssen weniger gutmütig (ich meine Handeln, nicht den persönlichen Charakter!) vorgehen.

[ASAPAndroid](#) ist die Basis für ASAPApps auf Android. Bluetooth ist implementiert. Wir wissen, dass auch Wifi-Direct sehr gut passen würde und ebenfalls LoraWan.

Live-Tracker App als SharkComponent

Sie implementieren eine SharkComponent (<https://mediathek.htw-berlin.de/album/video/Mobile-P2P-Apps-entwickeln-mit-ASAP-Shark-Grundlagen-ASAPJava-SharkPeer/a0bd28f604bfd6cd0bb3d28329bdcc75/278>, <https://github.com/SharedKnowledge/ASAPJava/wiki/SharkComponent>)

Diese SharkComponent realisiert ein Tracking-System. Zu dem Thema gibt es bereits Vorarbeiten, siehe **Lukas Ludwig** *Entwicklung eines offline GPS-Trackers für Android in dezentralen Netzen* (http://www.sharksystem.net/htw/FP_ICW_BA_MA/index.html#ludwig2021)

In dieser Arbeit wurde das prinzipielle Vorgehen des Trackings aber vor allem die Arbeit mit osmdroid beschrieben. Der realisierte Code läuft, lässt sich aber nur schwer in Shark-Apps integrieren. Dabei ist kann das Live-Tracking die Basis vieler interessanter Anwendungen sein:

In geschlossenen Chats könnten sich Teilnehmer:innen ihren Standort mitteilen. Wir könnten ortsbasierte Spiele realisieren, die in Ad-hoc Netzwerken arbeiten. Die zu entwickelnde Komponente wird die Basis vieler anderer sein und soll unter GPL publiziert werden, wenn Ihnen das recht ist. Sie soll die Tracks aufzeichnen, die in einer Nutzergruppe geteilt werden können.

Sie zeigen durch einen Prototypen, dass diese SharkComponent lauffähig ist.

Gruppenbasierte Verschlüsselung für ASAP

ASAP ist die Basis für ein mobiles P2P System. ASAP unterstützt End-to-End und Punkt-zu-Punkt-Verschlüsselung basierend auf asymmetrischen Schlüsselverfahren (siehe <https://mediathek.htw-berlin.de/album/video/ASAP-Routing/7236f992117295667ad666c9cd1fc4cc/278>).

Es gibt die Möglichkeit der Erzeugung von Gruppenschlüsseln. Dazu entstand eine Bachelorarbeit, siehe **Vladimir Vilenchik** *Entwicklung eines Protokolls zur Erzeugung von Gruppenschlüsseln in Ad-Hoc-Netzwerken* (http://www.sharksystem.net/htw/FP_ICW_BA_MA/index.html#vilenchik2020). Die Verfahren sind dort ausgezeichnet beschrieben. Der Prototyp basierte allerdings auf einer sehr frühen Form von ASAP, die nunmehr verworfen wurde.

Primäre Aufgabe ist die Implementierung eines Packages, das diese Gruppenschlüssel für ASAP Apps bereit stellt. Sie beweisen die Funktionsfähigkeit durch eine beliebig einfache App. Sie prüfen außerdem, ob es alternative Möglichkeiten zur Erzeugung von Gruppenschlüsseln gibt.

Wozu kann das dienen? Für die Verschlüsselung von Chats wäre das ganz ausgezeichnet. Eine Chatgruppe könnte einen Gruppenkey erzeugen. Der Key muss von allen teilnehmenden Geräten erzeugt werden und kann auch nicht verändert werden. Das ist mehr als End-to-End-Verschlüsselung. Es gibt Key für eine konkrete Konversation, die nicht dadurch gebrochen werden kann, wenn ein private Key eines teilnehmenden Gerätes gestohlen wird. Es wäre auch eine enorm spannende Basis für ein ASAP-E-Payment-System an dem gerade gearbeitet wird.

Service Discovery in Ad-hoc Netzwerken mit Android

Die meisten Android-Phone unterstützen zwei Protokolle zum Aufbau von Ad-hoc Netzwerken: Bluetooth und Wifi-Direct. Im Projekt Shark unterstützen wir derzeit Bluetooth in der Bibliothek ASAPAndroid. Beide Protokolle unterstützen aber auch Service Discovery. Geräte können so Dienste anbieten und andere können diese Angebote erkennen ohne eine Verbindung aufzubauen. Ermöglicht wird das durch z.B. durch das Service Discovery Protocol von Bluetooth. ASAP Peers können u.a. auf Smartphones laufen. Nahezu alle Projekte konzentrieren sich derzeit darauf. Derzeit aber erzeugen wir ein Bluetoothverbindung zu allen Geräten in der Umgebung. Die meisten werden aber kein ASAP unterstützen. Das soll sich ändern.

Sie sollen eine Implementierung realisieren, in der für Bluetooth und Wifi-Direct die Geräte ermittelt werden, die ASAP unterstützen. Zu denen soll dann eine Verbindung aufgebaut werden. Sie prüfen die Möglichkeit von Multicast-Kommunikation in den Protokollen. Sie schaffen die Möglichkeit, Messungen durchzuführen wie lang eine Verbindungsaufnahme dauert. (Das wird ein kleines Androidprogramm sein, das eine Zeitmessung erlaubt. Sie müssen Ihre Code sowieso testen, insofern ist das kein zusätzlicher Aufwand)

Sie diskutieren außerdem wie groß die maximale Anzahl von Bytes ist, die mittels einer Servicebeschreibung übermittelt werden kann. Der Hintergrund: ASAP Peers können Interessenprofile haben. Man könnte den Verbindungsaufbau perspektivisch erweitern, in dem nicht nur die Tatsache verbreitet, dass das Gerät ASAP-fähig ist, sondern auch welche Themen relevant sind. Das ist aber eine nachfolgende Arbeit. Hier interessiert die mögliche Datenmenge.

Huge Bluetooth Scatternet

Bluetooth ermöglicht mobile Ad-hoc Netzwerke (MANETs). Bluetooth ermöglicht die Etablierung einer Picozelle in der bis zu acht Geräte Daten austauschen können. Oft beinhalten Picozellen lediglich zwei Geräte, wie z.B. ein Headset und einen Rechner. Die Reichweite von BT liegt bei ca. 10 Metern.

Der Bluetooth-Standard definiert ebenfalls Scatternets. BT Geräte können innerhalb von zwei Picozellen gleichzeitig sein. Ein solches Gerät kann daher als Router dienen, der Daten einer Picozelle in einer andere überträgt. Das ist Routing.

Leider aber bieten BT-APIs diese Fähigkeit selten an, vermutlich unterstützt die Hardware diese Fähigkeit nicht immer. Das ist schade. Aber das kann auch programmiertechnisch gelöst werden und darum geht es in dieser Arbeit. Hier ist die Idee:

Ein BT-Gerät kann in seiner Umgebung nach Geräten suchen und eine maximale Anzahl von Verbindungen aufbauen. Diese maximale Zahl gilt es auf unterschiedlicher Hardware einmal zu testen (in Dokumentationen steht vieles, was nicht immer stimmt). Im realen Einsatz sollte auch noch die Chance bestehen z.B. ein BT-Headset zu betreiben. Ein Gerät kann als eine variable Anzahl von Verbindungen zu anderen Geräten aufbauen – nennen wir diese Zahl n .

Der Aufbau solcher Verbindungen wird von keiner zentralen Stelle koordiniert. Angenommen die Geräte A und B kommen in Empfangsreichweite und haben noch keine Verbindung aufgebaut. Hier nur eine Skizze eines Algorithmus für ein Gerät z.B. A:

1. Sind bereits n Verbindungen aufgebaut?

Nein: weiter mit 2.

Ja: Ist B bereits von A aus über einen anderen Pfad erreichbar, d.h. gibt es ein Gerät Y für den diesen Pfad existiert: A-Y-...-B ? Nein: Abbruch.

Ja: Trenne die Verbindung zu Y und verbinde direkt mit B. (Ziel: Geräte mögen sich mit Geräten verbinden, die in physischer Nähe (Sendereichweite) befinden. Verbindungen über mehrere Hops sollen gelöscht werden. Das führt in der Tendenz zu einer geographische Ausbreitung des Netzes und einer Reduktion der Vernetzung innerhalb eines Gebietes. Wir decken eine wachsende Fläche ab.)

2. Gibt es einen Pfad A-...Z-B?

Nein: Verbinde A mit B.

Ja: Fordere Z auf, die Verbindung mit B zu trennen. Baue eine Verbindung zu B auf. (Ziel: Sie Punkt 1.)

In jedem Fall: Teile neue Verbindung alle erreichbaren Geräten mit.

Weiter mit 3:

3. Finde Geräte X, die vor dem Aufbau einer Verbindung sowohl von B als auch A erreichbar waren. Welche Weg ist länger? A-X oder B-X. Lösche die Verbindung des längsten Weges. Wenn beide gleich lang sind entscheidet der Zufall, z.B. mittels der MAC-Adresse.

Dieser Algorithmus dient dazu, einen zyklensfreien Graphen aufzubauen. Prinzipiell ist die Größe eines solchen BT-basierten Netzwerkes unbegrenzt. Das ist auch nur eine Skizze eines Algorithmus – einer erste Idee, die auch verworfen werden kann.

Als Anwendungsszenario wird die Verknüpfung aller BT-Geräte von Studierenden in einem Seminarraum oder von allen Menschen innerhalb eines S-Bahnwagens gesehen.

Der Algorithmus ist unabhängig von Bluetooth, ließe sich also auch mit Wifi-Direct implementieren. In jedem Fall soll eine prototypische Implementierung erfolgen, die unabhängig vom genutzten Layer2 Protokoll ist. (Hier schlägt die Stunde vom Interfaces und ein paar kleinen Pattern. Software-Engineering ist gut!)

Diese Arbeit arbeitet dem Projekt ASAP/Shark lediglich zu, ist aber inhaltlich völlig unabhängig davon. Ein Einarbeiten in das Thema ist nicht notwendig. Alle ASAP-Apps würde aber sofort mit diesem BT-Ring arbeiten können.

Bewegungsprofil für ortsbasiertes Routing

ASAP Apps basieren ausschließlich auf dem Austausch von Nachrichten zwischen eigenständigen Peers (ASAP Peers). Wie sehen die Nachrichten aus? Wann werden sie ausgetauscht? Das ist anwendungsspezifisch. ASAP ist aber für mobile Geräte (und hier Android) optimiert. Ein Android Phone kann Host eines ASAP Peers sein. Dieses ASAP Peer kann ein dezentraler Messenger, ein Augmented Reality Game sein oder was immer man mag.

In dieser Arbeit soll eine Komponente entstehen, die für eine ganze Klasse von Anwendungen nutzbar sein kann. Nehmen wir einmal an, dass eine verteilte ASAP App ausschließlich auf Android Phones laufen sollen, die Menschen gehören, die regelmäßig unterwegs sind.

Nun kann es eine Klasse von Apps geben, die folgenden Wunsch haben: Eine Nachricht soll von einer *Gegend* auf der Erde in eine andere *Gegend* transportiert gebracht werden und dort allen interessierten Peers übermittelt werden.

Da gibt es also bspw. Peer Alice das sich in an einem Ort befindet, der schnell via GPS ermittelt werden könnte. Das Peer hat eine Nachricht, die in ein Gebiet gebracht werden soll und zwar ausschließlich über Ad-hoc Netzwerke. (Warum? Diese Frage steht hier nicht. Wir nehmen das als Voraussetzung an. Vielleicht ist es ein Spiel...). Dieses Gebiet lässt sich einfach als Polygon beschreiben oder noch einfacher als Rechteck.

Und hier ist die Frage, die sich stellt: Treffen sich Peer Alice und Peer Bob an einer bestimmten Stelle auf der Erde. Soll Alice die Nachricht an Bob senden? Die Antwort ist ja wenn zutrifft:

- Alice befindet sich bereits in der *Gegend* – alle Peers sollen die Nachricht erhalten, so auch Bob. Oder
- Bob ist regelmäßig in der *Gegend*, in die die Nachricht transportiert werden soll. Das Peer kann also als direkter Messenger agieren. Oder
- Bob ist regelmäßig in Gegenden, die näher an der Zielgend ist als Alice. In dem Fall ist Bob ein geeigneter indirekter Messenger.

Die Frage ist schnell herunter gebrochen auf eine Implementierungsaufgabe und zwei beliebig komplexe Algorithmen. Offenbar muss eine Bewegungsprofil erstellt werden. Das ist die Implementierungsaufgabe, die sich mit dem Location Service von Android unproblematisch lösen lässt. Man muss wissen wo sich die Peers bewegen, sonst lassen sich die Bedingungen nicht prüfen. Dieses Profil soll lokal gespeichert werden und das Peer nicht verlassen.

Der erste zu realisierende Algorithmus besteht in der Erstellung einer Geometrie, die das Bewegungsprofil repräsentiert. Am einfachsten wäre ein Clipping-Rechteck. Das ist enorm simple. Alles andere ist komplexer und besser.

Der zweite Algorithmus kommt zum Zuge, wenn Alice und Bob Nachrichten austauschen könnten, allerdings außerhalb der *Gegend*. Hier muss die Frage beantwortet werden: Soll Alice die Nachricht Bob übergeben? Zur Beantwortung dieser Fragen werden sichtlich einige Parameter benötigt: a) beide Bewegungsprofile und b) eine Entscheidung ob die Peers sich wechselseitig oder nur in eine Richtung diese Profile offen legen wollen. Außerdem steckt eine kleine geometrische Teilaufgabe darin: Ist Bob regelmäßig *näher* an der *Gegend* als Alice?

Die offensichtlich einfachste Implementierung dieses zweiten Algorithmus ist: Die Nachricht wird unabhängig von beiden Bewegungsprofilen immer übertragen. Alle anderen Varianten sind komplexer und besser.

Diese Softwarekomponente könnte die Basis vor allem für interessante, weil so bisher nicht gebaute ortsbasierte dezentrale Infosysteme (z.B. Spiele) sein. Sie würden ein Routingprotokoll implementieren, das auf der tatsächlichen Bewegung eines Rechners auf der Erde basiert. Das ist definitiv neu und machbar.

Corona - Warn-App

Die Android ASAP Library macht u.a. folgendes: Sie überprüft regelmäßig die Umgebung auf mögliche Kommunikationspartner. Derzeit unterstützen wir Bluetooth als Protokoll, Wifi-direct folgt. Änderungen in dieser Umgebung werden einer ASAP App über ein Listener-Interface bekannt gegeben. Die Library bietet damit die Möglichkeit zu speichern, wann und wie lange ein anderes Gerät in Bluetooth Senderreichweite war.

ASAP unterstützt End-to-End und Punkt-zu-Punkt Security. Es existiert außerdem eine Public Key Infrastruktur. Die API bietet Methoden zum Signieren / Verifizieren und Ver- und Entschlüsselung von Nachrichten an (<https://github.com/SharedKnowledge/ASAPJava/wiki/Cryptography>).

Das Aufzeichnen von GPS Tracks ist enorm einfach in Android realisierbar. Es liegen damit alle Technologien vor, um eine alternative Corona-Warn-App zu implementieren. Diese soll ermitteln, wann, wo und wie lange zwei Geräte (und damit deren Besitzer:innen) in räumlicher Nähe waren. Diese Daten können verschlüsselt und dank ASAP auch dezentral verteilt werden.

Auch existieren Apps, die erlauben, dass sich Menschen an bestimmten Orten registrieren, z.B. im Außenbereich von Restaurants.

Die Corona-Warn-App ist derzeit viel diskutiert. Ähnliche Anwendungen sind aber denkbar in ganz anderen Szenarien. Die Anmeldung an bestimmten Orten, aber insbesondere auch die Übermittlung und die Erkennung von Treffen in einem pur dezentralen System sind die theoretischen Herausforderungen dieser Arbeit. Die Implementierung wird dank ASAP im Rahmen einer Abschlussarbeit möglich sein.

Programmieren mit Tor

Das Tor-Projekt (<https://www.torproject.org/>) hat zum Ziel Empfänger und Sender IP basiert übertragener Daten zu anonymisieren. Es existieren APIs für TOR, z.B. Stem (<https://stem.torproject.org/api.html>), Thaliproject (https://github.com/thaliproject/Tor_Onion_Proxy_Library).

Die Aufgabe ist ebenso schnell formuliert wie sie Tiefe hat: Testen Sie existierenden APIs auf Anwendbarkeit. Es soll mit möglichst vielen APIs gezeigt werden, ob und wie stabil eine Verbindung über das TOR-Netzwerk zwischen zwei Entitäten aufgebaut werden kann. Es sollen Überlegungen zur Messung der Performance angestellt werden.

Es soll eine Software entstehen, die den Durchsatz der etablierten Verbindungen misst.

Das Thema eignet sich gut für alle, die sich schon immer mal mit Tor beschäftigen wollten, keine Scheu vor neuer Technologie und eventuell etwas instabilen Dokumentationen haben. Diese Arbeit

soll den aktuellen Stand der Dinge zum Thema Integrierbarkeit von Tor in eine eigene Java basierte Anwendung klären – ohne eine größere Anwendungen damit zu entwickeln.

GUI für SharkNet2

Im Projekt ASAP entstehen vor allem Bausteine zum Bau dezentraler Anwendungen. Es wird aber immer eine nicht-triviale Anwendung benötigt, um Software zu testen. Es geht dabei weniger um funktionale Tests – das leisten Unit-Test innerhalb der einzelnen Projekte schon sehr gut.

Es geht um das Prüfen, welche Funktionen generell benötigt werden. Es passiert leider sehr oft, dass Elemente einer Software entwickelt werden, die lediglich die Entwicklergemeinschaft glücklich macht (was ja schön ist), die das Gesamtprojekt aber gar nicht braucht. Diese Prüfung ist SharkNet2, siehe <https://github.com/SharedKnowledge/SharkNet2Android/wiki>

SN2 ist eine Messenger-App wie es viele gibt. SN2 kann allerdings (und in Version 1 ausschließlich) auch auf Ad-hoc Netzwerken arbeiten, d.h. z.B. Bluetooth. Es ist eine Messenger-App die ohne Internetzugang funktioniert und keinerlei Infrastruktur benötigt. Es gibt auch dazu alternative Ansätze, aber aus aktueller Sicht, keinen der so radikal auf pure Dezentralität und Freiheit von der genutzten Infrastruktur setzt wie SN2.

Das Projekt hat nun eine Reife erreicht, dass die App mehr sein kann als nur ein Raum, in dem man Software-Architektur-Ideen prüfen kann. Die App soll nutzbar werden, ein Showcase, gern auch für den dauerhaften Einsatz.

Die App funktioniert recht gut. Das GUI der App ist aber grauenhaft. Die erste Aufgabe besteht im Design einer ansprechenden GUI. Die einzige Vorgabe: Das Icon ist gesetzt: <http://www.sharksystem.net>.

Das allein wäre kein ausreichendes Thema. SN2 ist ein komponentenbasiertes System. Es ist eine Plattform für die weitere Komponenten entwickelt werden können, siehe <https://github.com/SharedKnowledge/SharkNet2Android/wiki/SNComponents>

Im Rahmen der Arbeit soll gezeigt werden, wie neue entwickelte Komponenten existierende grafische Elemente nutzen können. Es soll eine Guideline vor allem technischer Natur entwickelt werden. Es soll auch aufgezeigt werden wie die grafischen Elemente einfach austauschbar sind und wie sogar eine personalisierte grafische Nutzeroberfläche entstehen kann.

Die Arbeit ist für alle geeignet, die sich gern mit grafischen Elementen beschäftigen, aber auch mit Software-Design und adaptiven Programmen. Sie müssen keine Designer:in sein. Wir sind noch immer ein IT-Studiengang. Wichtig ist dass sich jemand einmal Gedanken über die Strukturen und Nachnutzbarkeit macht, bevor weitere Komponenten entstehen.

Dezentrale Personenprofile

ASAP ist die Basis für pur dezentrale Systeme. Basierend auf ASAP entstand ein System, das Zertifikate von Nutzer:innen verteilt. Zertifikate sind die Basis für die Verschlüsselung und digitale Signaturen.

Wir haben allerdings derzeit noch kein System, dass Personenprofile versendet. Solche Profile beschreiben Personen näher. Hier sollen sich Personen nur selber beschreiben können, nicht andere.

Entscheidend ist, dass Personen auch ihre Interessen beschreiben können. Dieses Thema macht es zu einem nicht-trivialen Thema.

ASAP Systeme sind pur dezentral. Nutzer:innen und ihre Apps haben sich ggf. noch nie vorher getroffen. Wie erreicht man nun aber, dass sich mitteilen können, was sie interessiert, ohne dass sie sich vorher über die Worte ausgetauscht haben. Erstaunlicherweise gibt es eine denkbar nicht-standardisierte Lösung, die gut funktioniert: Hashtags.

Im Kern geht es in dieser um Vokabulare und deren Implementierung. Die Arbeit kann sich – je nach Ihrer Interessenlage – theoretisch auch gern in die Thematik des Semantic Web entwickeln. Hier beschäftigt man sich mit genau dieser Thematik. Auch hier kann man es beliebig komplex aber auch sehr sehr einfach machen. Im übrigen sind semantische Netze derzeit enorm interessant, weil sie das Mittel sind, um Sinnzusammenhänge darzustellen. Die Ergebnisse von Data-Mining etc. müssen ja irgendwie dargestellt und gespeichert werden. Eine Vertiefung in SW ist aber nicht zwingend nötig.

Notwendig aber ist – wir sind ein IT-Studiengang – eine Implementierung. Es soll eine SN2-Komponente entstehen, die die Verteilung von Persönlichkeitsprofilen ermöglicht. Was man mit dieser Komponenten aus Anwendungssicht macht ist die nächste Frage, die in dieser Arbeit nicht beantwortet werden muss (ob es eine Businessplattform zum Finden von Geschäftspartner, eine Variante einer Find-a-Friend-App ist aus technische Sicht egal.)

Wir bewegen uns in pur dezentralen Systemen. Sie entwickeln eine App, die definitiv zweifelsfrei allen Datenschutzgesetzen der Welt genügt. Das ist gut und davon gibt es kaum welche.

Siehe auch

<https://github.com/SharedKnowledge/SharkNet2Android/wiki>

<https://github.com/SharedKnowledge/SharkNet2Android/wiki/SNComponents>

<https://github.com/SharedKnowledge/ASAPAndroid/wiki>

https://de.wikipedia.org/wiki/Semantic_Web

Einfache Gestenerkennung für Nachrichtenaustausch

Das sehr geschätzte Berliner Unternehmen Art + Com brachte vor Jahren einmal eine wunderbare iPhone App heraus, die durch ihre schlichte Genialität begeisterte (Hoccer). Es war ein lokaler Messenger, ein Austausch für Kontaktinformationen und eine App mit der man lokal Nachrichten hinterlassen konnte.

Gesten ersetzen das Tippen auf dem Gerät.

Gestenpaar 1: Ein Wurfgeste (wie: Wurf eines leichten Gegenstandes aus der Hüfte heraus) sorgte dafür, dass Kontaktinformationen oder eine kleine Nachricht gesendet wurde. Ein Fanggeste (wie: Hochhalten eines Federball-, Tennis- o.ä. Schlägers) erlaubte die Nachricht aufzufangen. Beide Gesten mussten gleichzeitig innerhalb einer maximalen Distanz durchgeführt werden.

Gestenpaar 2: Es wird eine Geste vollführt, als ob am oberen Rand des Handy etwas kleines klebt und man dieses abschütteln will. Das Handy wird ruckartig nach unten bewegt. Damit wurde eine vorher erstellte Nachricht an den Ort gebunden, an dem man sich gerade befand. Eine schaufelnde

Geste (man bewegt das Handy wie eine kleine Schaufel und als ob man etwas Kies aufnehmen wollte) nimmt eine Information auf, die an diesem Ort hinterlegt wurde.

Hoccher wurde leider vor geraumer Zeit eingestellt. Wir wollen die Idee wieder beleben. Im Rahmen der Arbeit soll eine Software-Komponente entstehen. Sie soll gestartet werden können. Sie soll danach die Bewegungen des Handys überwachen. Sobald eine Geste erkannt wird, soll eine Listener gerufen werden.

Die Bewegungssensoren sollen als Basis der Gestenerkennung dienen. Es wird davon ausgegangen, dass man ohne weitere Frameworks, insbesondere welche zur Mustererkennung auskommt. Aber das gilt es zu prüfen. Die Komponente soll Teil von SN2 werden und muss daher unter GPL 3.0 publiziert werden können. Tests sollen die Trefferrate der Gestenerkennung statistisch nachweisen.

Projekt OHDM

Rendern historischer Kartendaten

Im September 2020 wird die erste Version von OHDM auf einem Server der HTW installiert. Das Rendern der Daten erfolgt durch den Geoserver. Der wiederum nutzt CSS, die Kommilitonen von Ihnen in einen anderen Github-Projekt fanden

(<https://github.com/OpenHistoricalDataMap/OHDMConverter/tree/master/src/ohdm2geoserverrendering/resources/css>)

Diese Styles sind nicht perfekt. Alternativ werden die Daten durch Mapnik gerendert. Mapnik wird in OSM eingesetzt und daher für uns von besonderem Interesse. Es ist zu prüfen, ob und wie dessen Styles übernommen werden können.

Alternativ entstand im Rahmen einer Abschlussarbeit von Herrn Erbert ein System, das aus einer enorm simplen Beschreibungssprache für Style, die komplexeren und oft sehr redundanten Styles generiert, siehe http://www.sharksystem.net/htw/FP_ICW_BA_MA/index.html, konkret hier: http://www.sharksystem.net/htw/FP_ICW_BA_MA/2019_Ebert_Bachelorarbeit.pdf. Der Code befindet sich hier: <https://github.com/OpenHistoricalDataMap/SLDlight>

Sie interessieren sich für Rendering von Daten. Es wird nicht erwartet, dass sie grafisch Herausragendes leisten – das ist aber sehr gern genommen. Wichtig ist, dass Sie sich technisch mit der Thematik des Renderns von Geodaten einmal ganz grundsätzlich beschäftigen wollen und keine Scheu haben, sich einige Ansätze anzuschauen und dazu beizutragen, dass OHDM besser wird.

Es sind bereits einige sehr erfolgreiche und interessante Arbeiten in OHDM entstanden und es laufen derzeit ebenfalls Arbeiten. Sie sind nicht allein.

Import historische Geometrien

OHDM 1.0 ist lediglich ein Viewer, der historische Daten visualisiert. Das ist natürlich nicht ausreichend. OSM bezieht seine Daten von Nutzer:innen, die mit OSM Tools die kartenrelevanten Geometrien, meistens mit OSM-Tools aufnehmen und auf dem zentralen Server speichern. Es gibt manchmal auch Massen-Uploads aus verschiedenen anderen Quellen, seien es öffentliche Stellen oder Unternehmen, die ihre Daten frei verfügbar machen.

Es zeigt sich, dass es eine Fülle geometrische Daten historischer Gegebenheiten gibt.

Landesgrenzen finden sich recht schnell. Aber man findet auch historische Grundrisse teilweise sehr alter Gebäude bis hin zu prähistorischen Siedlungen.

Es existiert ein Tool zum Import historischer Daten. Dieses Tool reicht lange nicht aus. Es erlaubt lediglich, dass eine Geometrie in die OHDM importiert wird, siehe

<https://github.com/OpenHistoricalDataMap/OHDMConverter/wiki/H2D>.

Es fehlen wichtige Funktionen. Beim Import werden einige Informationen über die Geometrien benötigt: Zeitraum der Gültigkeit, welche Art von Geometrie ist das. Diese Informationen müssen Nutzer:innen möglichst bequem eingeben können. Es existiert prototypisch eine Webanwendung dazu: <https://github.com/OpenHistoricalDataMap/OHDMImportTool>. Die kann, muss aber nicht genutzt werden.

Zu jedem Import muss irgendeine Form eines Logeintrags entstehen, um den Import rückgängig machen zu können. Der OMDMConverter sieht das bereits z.T. vor, aber eben nur zum Teil.

Es wird ein Admin-Tool benötigt, das es schnell und sicher erlaubt, die Importe anzuzeigen und ggf. rückgängig zu machen. Ebenfalls ist vor einem zu prüfen, um es Widersprüche geben würde. Zwei Geometrien, die an der gleichen Stelle zur gleichen Zeit den gleichen Geometrietypen darstellen widersprechen sich wahrscheinlich. Aber eben nur wahrscheinlich.

Sie arbeiten gern mit Datenbanken. Sie können programmieren, verstehen aber auch den Scharm von einfachen, effektiven Tools, die das Leben einer/s jeden Administrator:in schöner macht. Sie nehmen die Anstrengung in Kauf, sich in Software anderer einzulesen, weil Sie Open-Source-Projekte generell gut finden gern in einem mitarbeiten.

Sie reihen sich gern in die Reihe ganz ausgezeichneter Arbeiten ein, die die Basis von OHDM sind wie es derzeit läuft: http://www.sharksystem.net/htw/FP_ICW_BA_MA/index.html#ohdm

Nebenbei macht es Ihnen Freude an einem System zu arbeiten, das es derzeit wirklich nicht gibt – immer wieder erstaunlich – das aber so viele Leute gern nutzen würden. Dann könnte das Thema passen.