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.

Inhaltsverzeichnis

Projekt Shark/ASAP

Shark/ASAP ist ein Open-Source Developer-Framework. Damit lassen sich mobile dezentrale Anwendungen implementieren, die natürlich über das Internet nutzen **aber nicht müssen**. Die übergroße Mehrheit der mobilen Anwendungen greift auf einen Server zu und braucht dazu Internet-Access. Praktisch alle Peers von P2P Anwendungen laufen im Internet. Das ist alles in Ordnung wenn es um E-Commerce Anwendungen und digitale Währungen.

Das ist nicht mehr so *natürlich*, wenn es um Messenger-Dienste geht. Etwas dezentraleres als die menschliche Kommunikation kann man sich kaum denken. Viele Menschen reden, geben Infos weiter etc. pp. Das läuft seit einer Millionen Jahren stabil ohne Cloudservice und das soll auch so bleiben. Ich erwarte von einem Messenger, dass er immer funktioniert und dass er ein Kommunikationsweg nutzt, den ich in Ordnung finde und der gerade verfügbar ist. Das ist es was wir bauen. Machen Sie mit. Hier ein paar Themenvorschläge. Sprechen Sie mich gern an.

Prof. Dr.-Ing. Thomas Schwotzer (thomas.schwotzer@htw-berlin.de)
HTW Berlin, WH C 616.

Generell kann man die Arbeiten in zwei Klassen teilen: Sie arbeiten an a) Anwendungen für Shark/ASAP oder ermöglichen b) die Kommunikation über weitere Netzwerke.

Offene Themen – Anwendungsentwicklung

Shark ist ein Developer-Framework. Man kann also Anwendungen bauen, die in andere Anwendungen integriert werden können. Mit SharkComponent existiert dazu ein Komponentenmodel. Kein Grund davor zurück zu schrecken. Es ist nicht komplex. Es ist aber die Basis auf der wir Anwendungen, konkreter: Komponenten implementieren. Das würden Sie in den folgenden Arbeiten auch tun.

Integration von PGP in SharkPKI

Basierend auf Shark/ASAP wurde eine eigene SharkPKI implementiert, die auch stabil arbeitet. Vor allem Open-Source Projekte sollten natürlich offen sein für andere offene und freie Projekte.

Es liegt auf der Hand, dass man Pretty Good Privacy (PGP) in Shark nutzt. Die Idee von PGP ist einfach wie effektiv. Menschen erzeugen Keypaare und können diese auch zertifizieren. Oft aber werden die Keypaare lediglich auf zentrale Server gestellt. Von dort werden sie bezogen und im besten Fall werden die Fingerprints verglichen bevor sie genutzt werden.

Die SharkPKI arbeitet mit RSA Schlüsseln. Damit lassen sich natürlich auch PGP Key nutzen. Ziel der Arbeit ist es, ein Konzept zu entwickeln wie PGP mit der SharkPKI zusammen arbeiten kann. Prototypisch soll gezeigt werden, dass das auch programmierbar ist. Ein vollständig lauffähiger Prototyp wäre nett, aber erwartet wird nur eine Teilimplementierung.

SharkMessenger Android GUI

Der SharkMessenger ist DER Showcase für das Projekt. Die grundsätzliche Idee von ASAP ist die Verteilung von Nachrichten über Ad-hoc Netzwerke nach Prinzipien von Gossip- und

opportunistischen Protokollen. Der Messenger funktioniert nunmehr recht stabil. Unit- und eine wachsende Anzahl von Lasttests beweisen das.

Es gibt eine rudimentäre GUI im Projekt SN2. Die Software ist nun stabil genug, um an einem Release zu bauen, der erste Schritt ist eine Alpha-Version. Das ist der Job. Es gibt bereits ein Programmiergerüst. Sie werfen alles weg, fangen von vorn an oder arbeiten sich ein und machen es besser.

Ihr Ziel ist die Erstellung eines funktionierenden Prototyps inklusive von End-to-End Tests auf Android.

Das Projekt eignet sich besonders für **Projekt im Bachelor und Master** und insbesondere auf für **Independend Coursework im AI-Master**. Es lässt sich zu einer Abschlussarbeit ausbauen.

SharkPKI Android GUI

Der SharkMessenger auf Android ist DER Showcase für das Projekt. Die App beinhaltet eine PKI. Wir müssen Schlüssel verteilen, um End-zu-End-Verschlüsselung und Signaturen zu realisieren. Die SharkPKI existiert und läuft stabil. Eine existieren nicht mehr als ein paar schlechte Zeilen Code in Android, die diese PKI für Endnutzer:innen benutzbar macht.

Ihr Ziel ist die Erstellung eines funktionierenden Prototypens einer GUI inklusive von End-to-End Tests auf Android.

Das Projekt eignet sich besonders für **Projekt im Bachelor und Master** und insbesondere auf für **Independend Coursework im AI-Master**. Es lässt sich zu einer Abschlussarbeit ausbauen.

Offene Themen – Netzwerkprogrammierung

ASAP ist ein Routingprotokoll mit denen SharkPeers Daten austauschen und weiter leiten. ASAP ist das IP von Shark wenn man so will. ASAP benötigt lediglich eine Punkt-zu-Punkt-Verbindung zwischen zwei Geräten. Das kann eine z.B. Bluetooth-Verbindung in einem Ad-hoc Netzwerk sein, eine Long Range Wifi Verbindung aber auch eine TCP-Verbindung. Mehr Details: https://github.com/SharedKnowledge/ASAPJava/wiki/IntroConnections

ASAPHub – UI und Lasttests

Es schreibt sich so einfach auf: Peers können über TCP Daten in Shark austauschen. Wie aber soll das konkret gehen? Ein Smartphone erhält regelmäßig eine neue IP-Adresse genau wie Rechner, die über einen DHCP-Server mit einer versorgt werden. Diese Adressen sind regelmäßig nur in einem Subnetz gültig und generell ist es aus Sicherheitsgründen keine wirklich gute Idee, ernsthaft eine App anzubieten, die auf Handys einen TCP-Port aufmacht. Sie müssten den Anwender:innen dann auch noch erklären wie das mit der Firewall geht. Das Problem ist vermutlich klar. Geht so nicht.

Das Problem ist auch nicht neu und daher gibt es TCPRelays: Ein Programm erlaubt einen Verbindungsbau von zwei Seiten. Es schaltet diese beiden Seiten zusammen und schickt einfach ohne weitere Bearbeitung die Daten von der einen zur anderen Seite. Ja klar, kann so eine Software auch den gesamten Datenstrom protokollieren, analysieren und verfälschen. Daher ist Verschlüsselung so wichtig, aber darum geht es in dem Projekt nicht.

ASAPHub ist die Komponente in Shark/ASAP, die genau das leistet. Die Software arbeitet auch stabil. Es gibt zwei Modi, die hier aber nicht erläutert werden. Es geht um etwas anderes.

Shark/ASAP steht für Transparenz und Resilienz. Alle Messenger Apps routen ihre Daten über einen Server oder eine Reihe von Peers. Alle kommerziellen Anbieter erklären, dass das alles kein Problem ist, weil man ja keine Daten sieht, die Server in der Schweiz stehen und sowieso ist alles End-to-End verschlüsselt. Das soll nicht vertieft werden, weil meistens im Markting und nicht der Fachabteilung entstanden – aus gutem Grund.

Ich will Transparenz. Ich will dass unsere zentrale Einheit, die die Shark-Kommunikation über *das Internet* ermöglicht, für alle lesbar öffentlich macht, welche Daten sie sieht. Ich will das alle das sehen können und dann entscheiden, ob das für ihre Apps so in Ordnung ist. Was es in den meisten Fällen ja auch ist. Aber man sollte man klar und offen darüber reden und sich besser noch die Daten konkret anschauen.

Das ist Ziel des Projektes. Sie erweitern den existierenden Hub und erzeugen aus den Logfiles lesbare HTML-Seiten. Sie überlegen sich bei der Gelegenheit wie man Lasttests mit dem Hub machen kann (Spoiler: Es gibt Lasttest vom SharkMessenger. Die erledigen den Job praktisch bereits.) und diese auswertet.

Noch anonymere Kommunikation über Tor

Wenn man schon an einem sicheren dezentralen System baut, sollte man auch über Tor kommunizieren können (https://www.torproject.org/). Können wir leider bisher nicht. Tor 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: Suchen und 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. Sie bauen eine Software, die eine Punkt-zu-Punkt-Verbindung zwischen zwei Geräten herstellen kann und zeigen das Senden und Empfanger beliebiger Daten funktioniert. Eine Integration in das Shark/ASAP Projekt ist nicht notwendig.