Events Around Me

Ein Mobiler Datenschutzfreundlicher Veranstaltungskompass

Bachelor-Thesis von David Kelm

Juli 2012

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Fachgebiet Sicherheit in der Informationstechnik

CASED

Events Around Me
Ein Mobiler Datenschutzfreundlicher Veranstaltungskompass

Vorgelegte Bachelor-Thesis von David Kelm

Prüfer: Prof. Dr. Michael Waidner
Betreuer: Marco Ghiglieri, M.Sc.; Dipl. Inform. Lukas Kalabis

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 12. Juli 2012

(David Kelm)

Zusammenfassung

Durch die aktuell zunehmende Verbreitung von Smartphones wächst die Bedeutung von standortbasierten Diensten fortwährend. Immer mehr Soziale Netzwerke bieten aus diesem Grund Funktionen an, die auf Standortdaten basieren und verknüpfen auf diese Weise die nutzerspezifischen Daten mit Informationen über deren Aufenthaltsort. Da so die Bedrohung für die Privatsphäre der Nutzer steigt, besteht das Ziel dieser Arbeit darin, die relevantesten Probleme, die den Datenschutz der Location Based Social Networking Site (LBSNS) betreffen, zu analysieren und exemplarisch eine Anwendung zu implementieren, welche die Sicherheit der Daten gewährleistet sowie die Privatsphäre ihrer Nutzer respektiert und schützt.

In den nachfolgenden Kapiteln wird zunächst die Sensibilität der Lokationsinformationen einer Person dargestellt und erörtert inwiefern bei deren Verarbeitung eine gefährliche Bedrohung der Privatsphäre entsteht. Im Anschluss folgt eine Beschreibung einiger weit verbreiteter LBSNS sowie eine Analyse ihrer Datenschutzmechanismen.

Anschließend wird ein neues LBSNS namens *Events Around Me* vorgestellt, das im Rahmen dieser Arbeit entwickelt wurde. Dabei liegt der Schwerpunkt auf der datenschutzfreundlichen Verarbeitung sensibler Daten des Nutzers.

Events Around Me ist ein Veranstaltungskompass, der den Nutzern ermöglicht, die Anzahl der Personen an einem bestimmten Ort zu ermitteln und Informationen darüber zu erhalten, wie diese Personen ihn beurteilen. Auf diese Weise soll es möglich sein, Veranstaltungen in der Nähe zu finden und ein kurzes und schnelles Feedback zu der aktuellen Situation an diesem Ort zu erhalten. Dabei soll allerdings das Melden der eigenen Position keine Risiken für den Datenschutz des Nutzers bedeuten. Neben der Server Anwendung wird ebenso eine Android App implementiert.

Inhaltsverzeichnis

1 Einleitung **5**
- 1.1 Datenschutz . . . 6
- 1.2 Standortbasierte Dienste . . . 6
  - 1.2.1 GPS . . . 7
  - 1.2.2 Folgerungen für den Datenschutz . . . 7
- 1.3 Soziale Netzwerke . . . 8
  - 1.3.1 Facebook und Datenschutz . . . 8
- 1.4 Standortbasierte Soziale Netzwerke . . . 11
  - 1.4.1 Datenschutzprobleme der LBSNS . . . 12

2 Standortbasierte Soziale Netzwerke **13**
- 2.1 Facebook Places . . . 13
  - 2.1.1 Überblick . . . 13
  - 2.1.2 Datenschutz . . . 14
- 2.2 Foursquare . . . 14
  - 2.2.1 Überblick . . . 14
  - 2.2.2 Datenschutz . . . 15
- 2.3 Google Latitude . . . 16
  - 2.3.1 Überblick . . . 16
  - 2.3.2 Datenschutz . . . 17
- 2.4 Loopt . . . 18
  - 2.4.1 Überblick . . . 18
  - 2.4.2 Datenschutz . . . 19
- 2.5 Zusammenfassung . . . 19

3 Events Around Me **21**
- 3.1 Motivation . . . 21
- 3.2 Beschreibung der Anwendung . . . 21
- 3.3 Überblick Architektur . . . 22
- 3.4 Android Anwendung . . . 23
  - 3.4.1 Architektur . . . 23
  - 3.4.2 Implementierung . . . 29
- 3.5 Server Anwendung . . . 32
  - 3.5.1 Architektur . . . 32
  - 3.5.2 Implementierung . . . 34
- 3.6 Wirtschaftliche Möglichkeiten . . . 37
- 3.7 Mögliche Schwachstellen . . . 37
- 3.8 Verbesserungsmöglichkeiten . . . 39

4 Schlussfolgerung **41**

Abbildungsverzeichnis

1.1 Entwicklung der Marktanteile mobiler Betriebssysteme in Deutschalnd (Quelle: comScore 2012) . . . 5
1.2 Die Entwicklung von Privacy auf Facebook (Quelle: http://mattmckeon.com) . . . 9

2.1 Facebook Android App Screenshots . . . 13
2.2 Sichtbarkeit bei jedem Beitrag einschränken (Quelle: http://facebook.com) . . . 14
2.3 Foursquare Screenshots (Quelle: https://de.foursquare.com) . . . 15
2.4 Foursquare Datenschutz (Quelle: https://de.foursquare.com) . . . 16
2.5 Latitude Screenshots (Quelle: http://www.google.com/intl/de/mobile/latitude/) 16
2.6 Latitude Screenshot (Quelle: http://www.google.com/intl/de/mobile/latitude/) . 17
2.7 Loopt Screenshots (Quelle: androidxmarket.com) . . . 18
2.8 Loopt Settings (Quelle: https://www.loopt.com/user/settings) . . . 19

3.1 Übersicht Architektur von *Events Around Me* . . . 22
3.2 Verbreitung von Android Betriebssystemen im Mai 2012 (Quelle: http://developer.android.com /resources/dashboard/platform-versions.html) . . . 23
3.3 Anwendungsfälle im Rahmen von *Events Around Me* . . . 24
3.4 Screenshots zum Check-in der Android Anwendung *Events Around Me* . . . 25
3.5 Screenshots zur Karte der Android Anwendung *Events Around Me* . . . 26
3.6 Aufbau der Android Applikation *Events Around Me* . . . 27
3.7 Screenshots zum Menü der Android Anwendung *Events Around Me* . . . 28
3.8 Handshake bei *Events Around Me* . . . 29
3.9 Refresh der Karte bei *Events Around Me* . . . 30
3.10 Erweiterter Check-in bei *Events Around Me* . . . 31
3.11 Aufbau der Server Applikation von *Events Around Me* . . . 32

1 Einleitung

Als Smartphone wird eine mobiles Gerät bezeichnet, das mehr Möglichkeiten als einfache Handys bietet, z.B. mobiles Internet, während es jedoch stets klein und handlich bleibt. Dabei stellen die meisten neueren Smartphones schon Rechenleistungen in der Größenordnung von Laptops der unteren Preisklasse zur Verfügung. Diese begannen in den vergangenen Jahren den Markt der mobilen Geräte zu dominieren, bis sie Anfang 2012 in Deutschland einen Marktanteil von über 34% - bei den unter 30-jährigen sogar 51% - erreichten [1]. Viele aktuelle Smartphones besitzen einen GPS Empfänger oder andere Technologien, welche die exakte Bestimmung der Position eines Gerätes erlauben. Diese Enwicklung begünstigte eine verstärkte Verwendung der standortbasierten Dienste.

Die zunehmende Verbreitung von Smartphones bietet innovative Chancen und Möglichkeiten. Die Idee, einen Überblick über die Aufenthaltsorte der Freunde eines Nutzers zu geben, wirkt durch ihre Einfachheit bestechend. Einige Beispiele für solche LBSNS sind im Abschnitt 1.4 aufgeführt. Viele Anbieter von verbreiteten standortbasierten Anwendungen vernachlässigen jedoch den Datenschutz und somit zahlt der Nutzer - selbst bei einem kostenlosen Dienst - stets mit der unbewussten Herausgabe seiner sensiblen Daten. Diese Praxis schmälert für einen auf Datenschutz und Privatsphäre bedachten Anwender den Nutzen des Dienstes immens.

Aus diesem Grund besteht das Ziel dieser Arbeit darin, ein LBSNS zu entwickeln, das sowohl interessante und sinnvolle Nutzungmöglichkeiten bietet als auch den Schutz privater Daten gewährleistet. Zunächst werden wir dazu einige Definitionen einführen, um anschließend vergleichbare, bereits existierende und verbreitete Dienste analysieren um aus deren Schwächen einige Vorgaben für unser eigenes Modell herleiten.

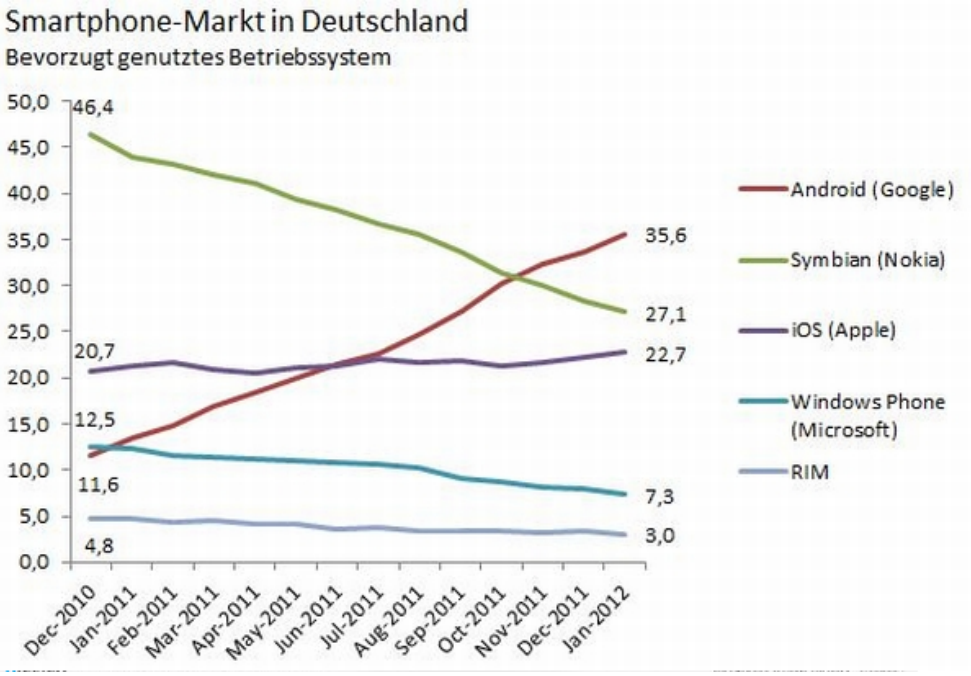

Abbildung 1.1: Entwicklung der Marktanteile mobiler Betriebssysteme in Deutschalnd (Quelle: comScore 2012)

Wie auf Abbildung 1.1 zu sehen ist, sind derzeit die mobilen Betriebssysteme Symbian (Nokia), Android (Google) und iOS (Apple) am weitesten verbreitet. Die Betrachtung der Trends zeigt, dass der Marktanteil von iOS konstant bleibt, der von Android stark steigt, während der von Symbian rückläufig ist. Der thematische Fokus wird in der weiteren Betrachtung daher auf den beiden Systemen Android und iOS liegen, wobei die Anwendung *Events Around Me* für Android implementiert wird.

1.1 Datenschutz

Als Datenschutz wird der Schutz von personenbezogenen Informationen bezeichnet. Jede Person soll, im Kontext des Rechts auf die informationelle Selbstbestimmung [2], über die Möglichkeit verfügen, selbst zu bestimmen, wer bestimmte Informationen über sie erhält. Im Besonderen sind dabei diejenigen Daten zu schützen, deren Missbrauch eine Gefährdung für die Privatspähre einer Person bedeutet.

Dazu ist es notwendig, sicherzustellen, dass in jedem Arbeitsschritt eines Dienstes die Konzepte für einen verantwortungsvollen Umgang mit Nutzerdaten beachtet werden. Im Zuge der Datenminimierung [3] wird z.B. vorgeschlagen, dass nur notwendige Daten gesammelt werden. Diese sollten darüber hinaus nur zweckgebunden gespeichert bleiben. Zudem muss, nach geltendem Recht, der Nutzer Auskunft über seine Daten erhalten können und stets die Möglichkeit besitzen, alle ihn betreffenden Daten zu löschen [4]. Eine Ausnahme hierfür bieten Daten, die aggregiert und anonymisiert gespeichert sind, und somit keine Rückschlüsse auf spezifische Personen zulassen.

Problematisch wird die Anwendung von nationalen Datenschutzrechten bei Unternehmen, die weltweit agieren. Für ein Unternehmen, dass eine Niederlassung in Deutschland betreibt und dort als verantwortliche Stelle Daten erhebt, verarbeitet oder nutzt, gilt das Bundesdatenschutzgesetz (BDSG) [4]. Befindet sich diese Niederlassung in einem anderem Mitgliedsstaat der Europäischen Union (EU), findet das dortige Datenschutzrecht Anwendung. Wenn sich die maßgebliche Niederlassung allerdings im EU-Ausland befindet, gilt das BDSG, wenn eine Datenerhebung im Inland erfolg. Im Allgemeinen wird dies angenommen, wenn zur Datenerhebung Mittel aus dem Inland genutzt werden, z.B. der Client Computer [5].

Für ein Unternehmen wie Facebook (FB) gilt folglich das BDSG, da die Datenverarbeitung und Entscheidung über diese nicht in der EU-Zentrale in Irland, sondern in den USA geschieht [6].

1.2 Standortbasierte Dienste

Ein standortbasierter Dienst oder auch Location Based Service (LBS) ist ein Dienst, der durch ein mobiles Gerät, heutzutage häufig ein Smartphone, zur Verfügung gestellt wird. Er verwendet den Ort des Gerätes, um dem Nutzer nützliche Informationen oder Funktionen anzubieten [7]. Ein solches Gerät benötigt immer ein Hardware-Modul, mit dem der Ort bestimmt werden kann und eine Komponente, um die gesammelten Daten zu verarbeiten.

Eine Beispiel für eine solche mobile Anwendung namens GPS Share[1] bietet die Funktionalität, einem beliebigen Kontakt per SMS die eigene, exakte Position mitzuteilen. Die App findet Anwendung, wenn sich zwei Personen, die sich in geringer Entfernung von einander aufhalten,

[1] http://kkinder.com/gps-share/

nicht finden können. Als Hilfestellung hierfür kann die eigene genaue Global Positioning System (GPS) Position ausgetauscht werden.

1.2.1 GPS

Das GPS ist laut Definition des Department of Defense [8] *"a space-based radionavigation system[..]. An unlimited number of users with a civil or military GPS receiver can determine accurate time and location, in any weather, day or night, anywhere in the world."*

Aktuell (Stand Mai 2012) sind 29 GPS-Satelliten aktiv, die stets so positioniert sind, dass von jedem Punkt auf der Erdoberfläche gleichzeitig eine Verbindung mit mindestens vier Satelliten hergestellt kann. Diese vier GPS-Satelliten müssen notwendigerweise in direkter Sichtlinie sein, damit ein GPS-Empfänger, wie ihn die meisten neueren Smartphones enthalten, ein ausreichendes Signal erhält, um seine aktuelle geographische Position zu berechnen [9]. Die Hauptkomponenten von GPS sind die absolute Position, welche für LBS die wichtigste Komponente darstellt, die relative Bewegung und die Übertragungszeit. Smartphones verfügen in der Regel noch über weitere Möglichkeiten zur Bestimmung der aktuellen Position, falls sie nicht mit den benötigten vier Satelliten kommunizieren können. Beispielsweise verfügen die Systeme iOS und Android neben GPS meist zusätzlich über die Ortungssysteme mittles Wi-Fi Verbindungen und Mobilfunkmasten Triangulation [7].

Zur Bereitstellung der Lokations-Daten wird ein einheitliches Format und Interface benötigt. Die im Gerät dafür zuständige Einheit muss folglich zunächst erkennen, welche Information die genaueste ist (GPS, Wi-Fi oder Mobilfunkmasten), den Ort aus diesen Daten berechnen und anschließend diese Informationen in einem für die LBS lesbaren Format zur Verfügung stellen. Normalerweise wird hierfür die Darstellung der Position als Längen- und Breitengrad in einem Koordinatensystem, durch welches alle Standorte auf der Erde eindeutig beschrieben sind, gewählt [8].

Zum Beispiel stellt die Android Plattform in ihrem Framework ein Paket zur Verfügung, welches den Entwicklern mit dem Gerät in Echtzeit gesammelte Lokations-Daten bereitstellt [10]. Die Benutzung dieses Frameworks ist auch ohne tiefere Kenntnisse in der Programmierung für die Android Plattform möglich; somit kann bei der Entwicklung der alleinige Fokus auf der Funktionalität des LBS liegen.

1.2.2 Folgerungen für den Datenschutz

Bei Lokations-Informationen handelt es sich um sehr schutzbedürftige Daten, aus denen sich eine noch größere Menge an sensibleren Informationen gewinnen lassen. Man stelle sich eine Person vor, die ein Jahr lang ein Gerät mit sich führt, das ihre Aufenthaltsinformationen protokolliert. Dadurch ist nicht nur feststellbar, an welchem Ort sich diese Person zu einem bestimmten Zeitpunkt aufgehalten hat. Mittels dieser Informationen werden auch noch weitere sensible Daten offengelegt: Weder der Wohnort, der Arbeitsplatz, die Fahrgewohnheiten (z.B. Auto/Rad/Bahn), noch die Urlaubszeit/-ziele oder die Größe des Freundeskreises bleiben verborgen. Anhand der Besuche von Restaurants lassen sich z.B. auch die Essgewohnheiten ablesen, der Kleidungsstil anhand der besuchten Geschäfte, die letzten Arztbesuche lassen Rückschlüsse auf Krankheitsverläufe zu.

Wenn zudem die Möglichkeit des Zugriffs auf die Lokations-Daten von Freunden und Kollegen besteht, können weitere Rückschlüsse mit einer viel höheren Trefferwahrscheinlichkeit gezogen

werden. Z.B. lässt sich in einem solchen Fall beantworten, wie oft die Person ihre Mutter besucht, ob sie ledig ist oder wer ihre besten Freunde sind.

Da der Schutz persönlicher Daten unterschiedlich priorisiert wird, schränkt jede Person den Bereich privater Daten auf unterschiedliche Weise ein [11]. Nichtsdestotrotz sind die so gewonnenen Informationen in höchstem Grade datenschutzsensibel, weshalb die betroffene Person stets selbst die Kontrolle darüber behalten sollte, wer diese Informationen über sie erhält.

Wie eingangs erwähnt, besitzt aktuell jeder Zweite der unter 30-Jährigen ein Smartphone[1]. Zudem möchte der Großteil der Gesellschaft mittlerweile weitgehend immer und überall verfügbar und online sein [12], sodass viele Nutzer ihr Smartphone stets bei sich tragen. Daraus folgt, dass ihr Aufenthaltsort häufig mit dem ihres Handys übereinstimmt. Da diese Geräte über die notwendige Ausstattung verfügen, um die eigene Position zu bestimmen, besteht der einzige Unterschied zwischen dem Gerät, das den Ort stets protokolliert und dem Smartphone darin, dass die Daten von dem protokollierenden Gerät gespeichert werden.

Im April 2011 hat Apple mit einer Sicherheitslücke im iOS auf dem iPhone, durch die in einer geheimen Log-Datei unverschlüsselt Ort und Zeitstempel gespeichert wurden [13], Aufsehen erregt. Diese Datei wurde zudem bei jeder Synchronisierung mit dem Computer übertragen. Sie konnte problemlos entwendet werden und bot somit Angriffsmöglichkeiten auf die Daten des iPhone-Nutzers. Das eigentliche Hauptproblem bestand jedoch darin, dass die Nutzer über die Existenz einer solchen Datei weder in Kenntnis gesetzt, noch um eine Einverständniserklärung bzgl. der Speicherung der sensiblen Daten gebeten wurden. Nach heftiger Kritik hat Apple ein Update herausgegeben, das sicherstellt, dass die Log-Datei nicht länger als sieben Tage gespeichert wird, und darüber hinaus der Nutzer die Möglichkeit hat, in den Einstellungen die Protokollierung dieser Daten zu verhindern [14]. Dieses Beispiel zeigt, dass durchaus Daten ohne die Kenntnis oder Zustimmung der Nutzer gespeichert werden. In diesem Fall verschmilzt das Smartphone mit dem protokollierendem Gerät. Kombiniert mit der Sensibilität der Daten ergibt sich bei der Verarbeitung von Lokations-Daten auf dem Smartphone eine gefährliche Bedrohung der Privatsphäre.

1.3 Soziale Netzwerke

Im Jahr 2011 gab etwa die Hälfe der Deutschen private Daten im Internet preis, den Hauptteil davon im Kontext von Sozialen Netzwerken. Allerdings halten 65% der Deutschen zwischen 14 und 29 Jahren ihre Daten dabei für nicht ausreichend gesichert [15].

Es ist weitgehend bekannt, dass Soziale Netzwerke mit Bedrohungen für den Datenschutz ihrer Nutzer zu kämpfen haben, da sie häufig eine immense Menge an Daten sammeln [16], und so besonders Gefahr laufen, dass ihre Daten missbraucht oder entwendet werden. Zudem ist es wichtig, dass die Nutzer einfache Möglichkeiten erhalten, auszuwählen, welche Personen Informationen über sie einsehen dürfen. Als Beispiel für ein Soziales Netzwerk zur Analyse wird hier Facebook herangezogen.

1.3.1 Facebook und Datenschutz

FB bot mit seinem gleichnamigen Sozialen Netzwerk eines der ersten auf dem Markt, das mittlerweile zu dem weltweit Größten angewachsen ist. Aktuell zählt FB über 900 Mio. aktive Nutzer

[17]. Das Unternehmen führte im Mai 2012 einen Börsengang durch und erhielt dafür eine Bewertung von ca. 104 Mrd. USD [18]. Unter Beachtung der Tatsache, dass der einzige Gegenwert, den FB bieten kann, die gesammelten Daten seiner Nutzer sind, ist es bemerkenswert, dass dieses Unternehmen einen solch immensen Wert besitzen soll.

Ein Unternehmen wie FB hat natürlicherweise die Intention Geld zu verdienen. Aufgrund der Tatsache, dass Facebooks gewinnbringendstes Gut die Daten seiner Nutzer sind, wird versucht so viele Daten wie möglich zu sammeln, um seinen Gewinn noch weiter zu vergrößern. Mit Hilfe dieser Informationsbasis kann das Unternehmen zielgruppengerichtete Werbung anbieten - die Haupteinnahmequelle für FB [19] [20]. Diese Tatsache kann für die Nutzer jedoch problematisch sein, da diese, wie bereits beschrieben, häufig keine Kenntnis über die Sammlung ihrer Daten und deren Ausbeutung besitzen.

Allerdings ist dies nur ein Aspekt der Kritik an FB, der der Firma unter anderem auch den *Big Brother Award 2011*[2] eingebracht hat. Dieser wird Firmen, Organistationen und Personen verliehen, um missbräuchlichen Umgang mit Technik und Informationen anzuzeigen. Ein weiteres Problem bietet der Datenschutz nach außen: Daten, die unter Umständen sensibel sind, sollten nicht von unerwünschten bzw. unautorisierten Außenstehenden einsehbar sein.

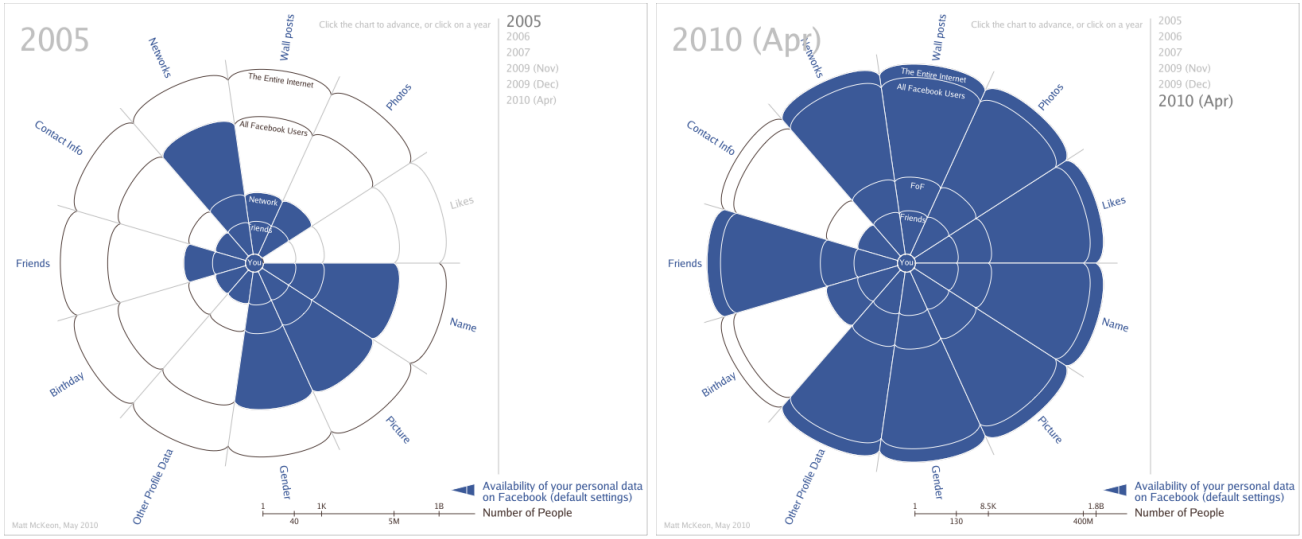

Abbildung 1.2: Die Entwicklung von Privacy auf Facebook (Quelle: http://mattmckeon.com)

In Abbildung 1.2 wird deutlich, wie sich über die Jahre die Standard FB Privacy-Einstellungen verändert haben. Der blaue Bereich markiert, welche Informationen für bestimmte Personen sichtbar sind: Der innerste Kreis zeigt die Daten an, die der Nutzer selbst sehen kann, dann folgen, von innen nach außen Freunde, Freunde von Freunden, alle FB Nutzer und schließlich das gesamte Internet. Mittlerweile regt sich allerdings immer stärkerer Widerstand gegen die Einführung neuer Nutzungsbedingungen [21], sodass FB in Zugzwang geraten ist, diese zunächst vermittelbar anzupassen. Auch die erneut angepassten Nutzungsbedingungen vom 20.04.2012 [22] wurden von vielen Nutzern kommentiert und abgelehnt [23]. Die darauffolgenden vorgeschlagenen Nutzungsbedingungen wurden ebenfalls abgelehnt, sodass im Mai 2012 eine Studentengruppe aus Wien[3] auf eine Abstimmung über die Nutzungsbedingungen drängte [24]. Da an dieser Abstimmung jedoch nach FB festgelegter Nutzerbeteiligung zu wenig Nutzer teil-

2 http://www.bigbrotherawards.de/2011
3 http://europe-v-facebook.org/DE/de.html, zuletzt aufgerufen am 22.05.2012

nahmen, sind die neuen Dokumente, trotz mehrheitlicher Ablehnung, Mitte Juni 2012 in Kraft getreten [25].

Target	Date Range	About Me	Account End Date
Account Status History	Adress	Alternate Name	Application
Chat	Checkins	Connections	Credit Cards
Currency	Current City	Date of Birth	Education
E-Mails	Events	Family	Favourite Quotes
Friend Requests	Friends	Gender	Groups
Hometown	Last Location	Linked Accounts	Locale
Logins	Machines	Messages	Minifeed
Name	Name Changes	Networks	Notes
Notification Settings	Notification	Password	Phone Numbers
Physical Tokens	Pokes	Political Views	Privacy Settings
Profile Blurb	Realtime Activities	Recent Activities	Registration Date
Relationship	Religious Views	Removed Friends	Screen Names
Shares	Status Updates	Vanity	Wallposts
Website	Work		

Tabelle 1.1: Nutzerdaten die Facebook speichert (Quelle: http://www.europe-v-facebook.org/)

Das Unternehmen wird auch häufig dafür gerügt, dass es den Nutzern zu schwer gemacht wird, Daten, die es einmal gesammelt hat, wieder zu löschen. Eine Übersicht, welche Daten FB nachweislich sammelt, ist in Tabelle 1.1 zu sehen. Erwartungsgemäß finden sich Informationen wie die Adresse oder der Name eines Nutzers unter diesen. Folgende Punkte bedürfen allerdings einer weiteren Erläuterung:

Checkins: Es werden alle Orte gespeichert, an denen der Nutzer jemals ”eingecheckt” (seine eigene Position gemeldet) hat.

E-Mails: Es wird eine Liste aller E-Mail-Adressen, die FB dem Nutzer zuordnet, erstellt. Nicht nur vom Nutzer eingegebene E-Mail-Adressen werden gespeichert. Auch von anderen Nutzern werden Angaben über die eigene Person gesammelt (z.B. wenn diese E-Mail-Adressen aus ihrem Adressbuch mit FB synchronisieren).

Friend Requests: Es werden alle getätigten und eingegangenen Freundschaftsanfragen gespeichert. Selbst nach einer Zurückweisung werden diese Daten nicht gelöscht.

Friends: Es wird vermutlich auch gespeichert, wie ”intensiv” eine Freundschaft ist (allerdings existieren hierzu keine Auskünfte von FB).

Machines: Jeder Computer erhält per Cookie eine eindeutige Nummer. Dadurch ist es möglich, verschiedene Nutzer einem Gerät zuzuordnen und zu erkennen, wer der Hauptnutzer ist.

Messages: Laut aktueller Datenschutzbestimmungen (Mai 2012) werden keine Nachrichten gelöscht. Selbst wenn der Nutzer eine Nachricht löscht, bleibt diese bei FB gespeichert [6].

Phone Numbers: Ähnlich wie bei den E-Mail-Adressen werden auch hier Telefonnummern gespeichert, die andere Nutzer angegeben haben.

Photos: Markierungen bleiben auch nach der ”Deaktivierung” gespeichert.

Realtime Activities: Alle Klicks, die auf der Facebookseite getätigt werden, werden gespeichert.

Removed Friends: Auch gelöschte Freunde bleiben gespeichert.

Die Auskunft über diese Daten wird nicht mehr ohne weiteres umfassend gewährt und ist aus *Europe vs Facebook*[4] entnommen. Diese Studentengruppe kritisiert Facebook stark und versucht Aufklärung über die Datenschutzprobleme von FB zu betreiben. FB selbst, hat, nach zu vielen Anfragen, begonnen, automatisiert Auskunft über die gespeicherten Daten zu erteilen[5], spart dabei allerdings einige Felder, die hier genannt sind, aus. Hierdurch wird es für die Nutzer um einiges einfacher, schneller und unkomplizierter, Auskunft über die eigenen Daten zu erhalten. Obwohl sie nicht mehr vollständig ist, sollte dies dennoch positive Erwähnung finden.

1.4 Standortbasierte Soziale Netzwerke

Standortbasierte Soziale Netzwerke oder LBSNS kombinieren die Möglichkeiten von Smartphones, im Speziellen die jederzeit mögliche Ortung, mit den Vorteilen von Sozialen Netzwerken. Die LBSNS kombinieren somit die nutzerspezifischen Daten mit geographischen Informationen, um ein noch detailliertes Bild über die Nutzer entwickeln zu können. Es folgt eine kurze Liste ausgewählter Möglichkeiten, die diese Technologie bietet:

- Freunden den eigenen aktuellen Ort mitteilen
- Den aktuellen Standort der eigenen Freunde ermitteln
- Informationen zu einem Ort posten, sodass andere Nutzer diese finden können
- Orte in der Nähe empfehlen oder Empfehlungen zu ihnen lesen
- Speziell zugeschnittene Werbung auf die Interessen und den Aufenthaltsort erhalten
- Automatisch beschriftete Bilder mit dem Ort, an dem sie aufgenommen wurden, hochladen
- Potentielle Partner aufgrund von ähnlichen Aufenthaltsorten finden
- Aktuell stattfindende Veranstaltungen in der Nähe einsehen

Es existieren verschiedene Möglichkeiten ein solches LBSNS aufzubauen. Z.B. kann ein bestehendes Soziales Netzwerk eine Erweiterung entwickeln, die Informationen zum Ort sammelt und Verknüpfungen erstellt; dies ist bei Facebook Places (FBP) der Fall. Es ist allerdings auch möglich, in direkter Weise ein LBSNS mit dem Ziel, als standortbasierter Dienst zu agieren, zu entwickeln. Problematisch ist in diesem Fall, dass zu Beginn keine Nutzer diesen Dienst nutzen. Da solche LBSNS häufig dieses Problem haben, die kritischen Masse zu erreichen, greifen sie

[4] http://www.europe-v-facebook.org/
[5] https://www.facebook.com/download/?h=AaDBxVHxpDR556rl

meist auf die APIs von FB, Twitter oder E-Mail-Anbietern zu, um Freundeslisten zu importieren und zu verknüpfen. Ein solches Beispiel stellt Foursquare[6] dar, das darauf basiert, dass Personen an verschiedenen Orten einchecken und für solche Handlungen jeweils Punkte verdienen.

1.4.1 Datenschutzprobleme der LBSNS

Durch die Vermischung der Lokations-Informationen mit denen der Sozialen Netzwerke multipliziert sich nicht nur der Wert der Daten, sondern auch das Risiko für den Datenschutz. Durch die Verknüpfung von Ortsdaten mit weiteren Informationen wie Beziehungen zwischen Nutzern, Interessen, Alter, Nachrichten und Aktivitäten kann sich der Inhaber solcher Informationen ein noch qualifizierteres Bild über das Leben einer Person verschaffen.

Auf diese Weise lassen sich zusätzliche Informationen gewinnen, die für viele Nutzer einen großen Wert besitzen und so sensibel sind, dass sie in jedem Fall geschützt werden sollten [20]. Allerdings tritt hierbei das Problem auf, dass die persöhnliche Grenze, welche Informationen Schutz bedürfen, sich durchaus von Nutzer zu Nutzer unterscheiden kann [11].

Die Öffnung der Programmierschnittstellen der Smartphone Betriebssysteme, allen vorran Android, ermöglicht es heute jedem Programmierer, auch ohne tiefergehende Kenntnisse der Plattformen, eine Anwendung zu entwickeln und sie auf dem ”market” zu veröffentlichen. Da diese Apps im Falle des Google Play Stores[7], dem Standard App Store von Android, keiner ausreichenden Kontrolle unterliegen, und die APIs von Facebook oder Twitter es möglich machen, einfach und ohne großen Aufwand den Datenbestand der Sozialen Netzwerke zu nutzen, ergibt sich eine weitere Bedrohung für die Privatsphäre der Nutzer: Ein Entwickler, der nicht auf die Privatsphäre achtet, und eine neue Idee für ein interessantes LBSNS hat, kann dabei diese Schnittstellen nutzen, und alle Daten, die auf diese Weise übertragen werden, möglicherweise sogar ganz legal (je nach Nutzungsbedingungen) sammeln. Diese kann er daraufhin weiterverkaufen, ungeachtet der Interessen oder Intentionen des Käufers.

Durch die automatische Übertragung von Informationen sammeln LBSNS häufig ein größeres Volumen an Daten als herkömmliche Soziale Netzwerke, die vor allem explizit freigegebene Informationen speichern. Daher sollten LBSNS, viel mehr noch als alleinstehende Soziale Netzwerke, Möglichkeiten für den Nutzer bereitstellen, um den Zugriff auf die gesammelten Daten zu regeln, um so das notwendige Maß an Datenschutz zu gewährleisten.

Idealerweise Orientieren sich die LBSNS am Prinzip der *Dataminimization* und sammeln ausschließlich Daten, die für die Ausführung des Dienstes benötigt werden [3]. Allerdings entsteht der Wert und der Umsatz dieser Anbieter in erster Linie durch die gesammelten Daten. Aus diesem Grund sind die auf Gewinn ausgerichteten Unternehmen stets darauf fixiert, eine maximale Menge an Daten zu sammeln.

In der Anwendung *Events Around Me* besteht die Zielsetzung hingegen darin, lediglich solche Daten zu sammeln, die für den Dienst tatsächlich benötigt werden. Auf diese Weise soll die Privatsphäre der Nutzer einem möglichst geringen Risiko ausgesetzt und den beschrieben Problemen begegnet werden.

[6] http://foursquare.com
[7] https://play.google.com/store

2 Standortbasierte Soziale Netzwerke

Die bereits beschriebenen Anforderungen, die ein LBSNS erfüllen sollte (allen voran die Wahlmöglichkeiten für den Datenschutz) werden in diesem Kapitel an den real existierenden FBP, Foursquare, Google Latitude und Loopt überprüft.

Das Wesen aller hier genannten LBSNS besteht darin, dass Nutzer ihren Aufenthaltsort mit Freunden teilen. Zusätzlich sind jedem Anbieter noch einige Eigenheiten gemein, die ihn von den Mitbewerbern abheben sollen.

Da das Ziel dieser Arbeit lautet, ein LBSNS zu entwickeln, das überprüfbare datenschutztechnische Sicherheit bietet, liegt der Fokus auf den Datenschutzproblemen der LBSNS. Zum Bereich der technischen Sicherheit der mobilen Anwendungen sei an dieser Stelle lediglich erwähnt, dass die Apps aller vier Anbieter stets über eine SSL Verschlüsselung kommunizieren. Dies wurde durch eine Analyse des Datenverkehrs mittels WireShark[1] festgestelllt.

2.1 Facebook Places

FBP ist eine Erweiterung für das Soziale Netzwerk von FB. Dieser Dienst hat mit der Datenbank von FB ein immenses Potential.

2.1.1 Überblick

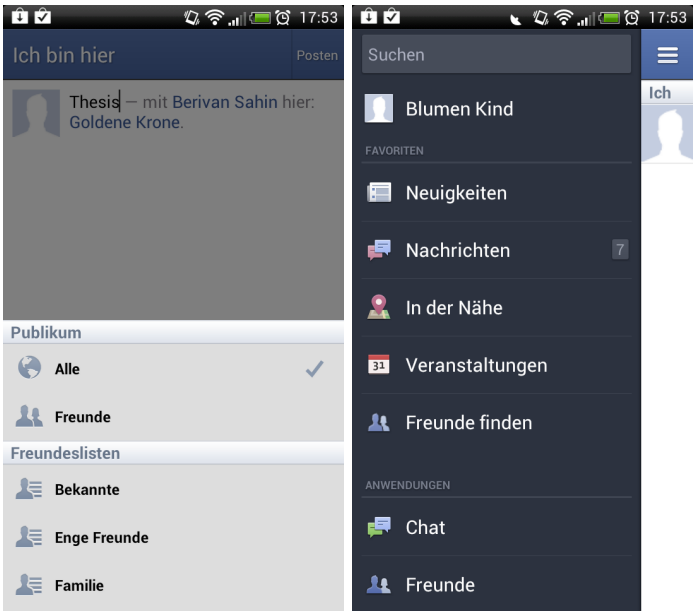

Abbildung 2.1: Facebook Android App Screenshots

FBP erhält seine Daten durch einen aktiven Check-in - das Melden der eigenen Position - der FB-Nutzer. Mit Hilfe der FB-App oder der mobilen Website[2] greift die Anwendung auf die

[1] www.wireshark.org/
[2] touch.facebook.com

Lokations-Daten zu. Zudem ist es möglich zu überprüfen, welche Freunde sich in der Nähe aufhalten, indem die Option ”In der Nähe” geöffnet wird (Abbildung 2.1 rechts). Wenn der Check-in Button betätigt wird, öffnet sich ein Formular, das dem Nutzer ermöglicht, weitere Informationen, etwa welche weiteren Personen sich an diesem Ort aufhalten oder was dort ”gerade gemacht” wird, anzugeben (Abbildung 2.1 links).

2.1.2 Datenschutz

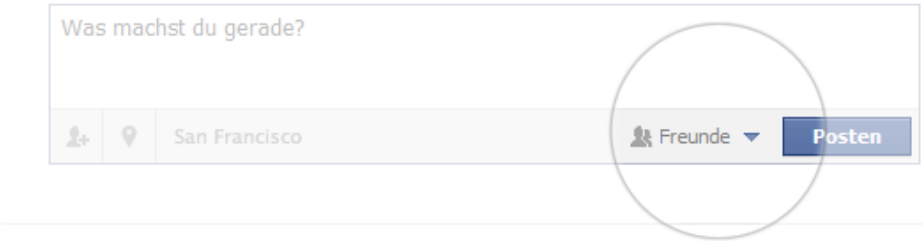

Abbildung 2.2: Sichtbarkeit bei jedem Beitrag einschränken (Quelle: http://facebook.com)

In den Datenschutzeinstellungen von FB ist es möglich einzuschränken, Beiträge, in denen ein Nutzer von Freunden makiert wurde, erst nach der Überprüfung und Freigabe durch den Nutzer anzeigen zu lassen. Zudem hat der Nutzer, wie in Abbildung 2.2 zu sehen, bei jedem Einchecken (wie auch bei allen anderen Einträgen innerhalb von FB) die Option, die Sichtbarkeit einzuschränken. Es ist positiv hervorzuheben, dass es dabei auch möglich ist, die Beiträge nur für spezielle Personen freizugeben. Allerdings legt die Standardeinstellung fest, dass die Beiträge für die gesamte Öffentlichkeit, auch außerhalb von FB, sichtbar sind. Die Erweiterung FBP besitzt bzgl. des Datenschutzes keine zusätzlichen Einstellungen.

In jedem Fall besitzt allerdings Facebook selbst vollen Zugriff auf jegliche angegebene Daten.

2.2 Foursquare

Foursquare ist ein LBSNS, das ähnlich wie FBP darauf basiert, dass die Nutzer an ihrem Ort aktiv einchecken. Da es sich hierbei um einen speziell als LBSNS entwickelten Dienst handelt, bietet Foursquare die Möglichkeit, Freundeslisten von FB oder Twitter zu importieren, um schneller ein größeres Netzwerk aufzubauen. Nach eigenen Angaben hatte der Dienst im April 2012 20 Millionen registrierte Nutzer [26].

2.2.1 Überblick

Foursquare bietet ebenfalls die Möglichkeit, entweder über die App oder die mobile Website[3] einzuchecken. In Abbildung 2.3 ist auf der linken Seite das Check-in Formular zu sehen. Jede andere Person, die auch gerade an diesem Ort eingecheckt hat, ist mit Namen (Synonyme erlaubt) und Bild sichtbar.

Der Dienst besitzt zudem ein Ranking-System, bei dem Punkte durch jeden Check-in gesammelt werden. Sobald ein Nutzer innerhalb der letzten 60 Tage die meisten Check-ins an einem

[3] https://m.foursquare.com

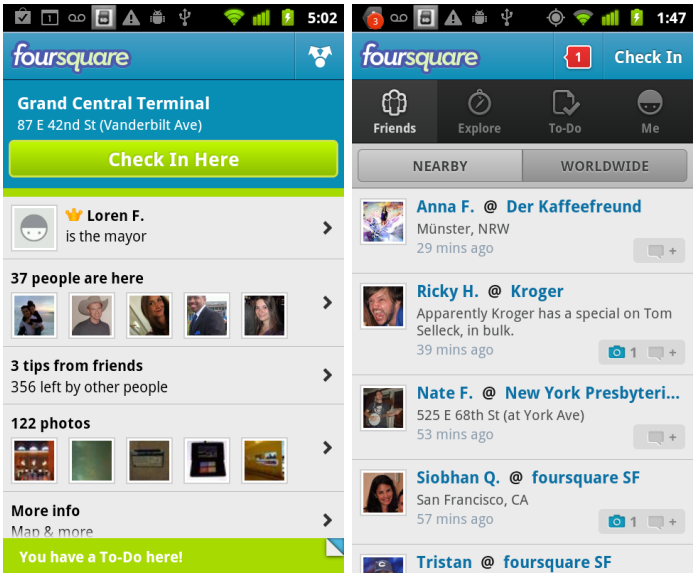

Abbildung 2.3: Foursquare Screenshots (Quelle: https://de.foursquare.com)

Ort getätigt hat, erhält er den Titel des Bürgermeisters für diesen Ort. Der Bürgermeister eines Ortes erhält bei einigen Geschäften Vergünstigungen, so kann z.B. das erste Getränk kostenlos sein. Mit diesem Punktesystem kann der Nutzer zudem verschiedene weitere Abzeichen sammeln. Darüber hinaus existieren drei verschiedene Level des Superuser-Status, die durch regelmäßiges Einchecken und Hinterlassen von anderen Informationen freigeschaltet werden können.

Da die Check-ins häufig auch zu FB und Twitter weitergeleitet werden, erreichen sie viele Anwender auch außerhalb der Grenzen von Foursquare. Die Webseite *PleaseRobMe*[4] ist ein Beispiel dafür, auf welche Weise sich diese Informationen ausnutzen lassen. Sie kann die Einträge von Foursquare und Twitter nach Ortsangaben durchsuchen und so feststellen, welche Häuser vermutlich leer stehen. Allerdings ist hier hinzuzufügen, dass sie lediglich das Ziel hat, Aufmerksamkeit auf die Datenschutzrisiken solcher Dienste zu fokussieren, und nicht realen Schaden zu verursachen.

2.2.2 Datenschutz

Die Einflussnahme des Nutzers auf den Schutz seiner Daten ist bei Foursquare sehr begrenzt. Lediglich die Sichtbarkeit der Kontaktinformationen und einige Funktionen bzgl. der Veröffentlichung des eigenen Standorts lassen sich einschränken. Wie in Abbildung 2.4 zu sehen, sind standardmäßig alle Funktionen aktiviert, sodass eine maximale Menge an Informationen freigegeben wird. Selbst wenn der Nutzer die Einstellungen angepasst und alle Optionen deaktiviert hat, bleibt jeder Check-in stets für alle Freunde sichtbar. Für diese Option ist keine Möglichkeit der Einschränkung gegeben. Zudem hat Foursquare auch stets Zugriff auf die Standortdaten eines jeden Nutzers.

4 http://pleaserobme.com/

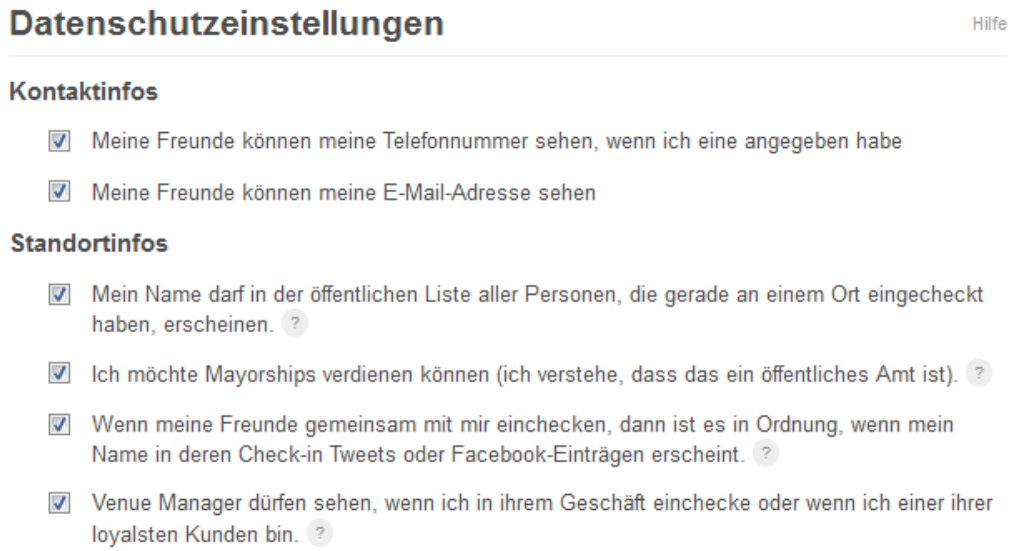

Abbildung 2.4: Foursquare Datenschutz (Quelle: https://de.foursquare.com)

2.3 Google Latitude

Google Latitude stellt ein LBSNS dar, das die Daten von Google (im Speziellen Google Mail und Google+) nutzt, um eine große Nutzerbasis zu erhalten. Hier existieren verschiedene Auswahlmöglichkeiten zur Verfolgung des Standorts.

2.3.1 Überblick

Abbildung 2.5: Latitude Screenshots (Quelle: http://www.google.com/intl/de/mobile/latitude/)

Bei Google Latitude kann der Nutzer den aktuellen Aufenthaltsort von Kontakten, die diesen für den Nutzer freigegeben haben, einsehen (Links und Mitte Abbildung 2.5).

Der Dienst verfügt über verschiedene Optionen, um den aktuellen Standort des Nutzers festzustellen. Wie auf dem rechten Bild der Abbildung 2.5 zu sehen ist, ist es möglich, dass der aktuelle Standort automatisch aktualisiert und geteilt wird; der Nutzer kann aber auch selbst einen Standort festlegen. Zudem kann der eigene Standort verborgen werden.

Mittlerweile wurden in Google Latitude ähnliche Funktionen wie bei Foursquare implementiert und veröffentlicht, die es unter anderem erlauben, Punkte zu sammeln und verschiedene Statuslevel für einen Ort zu erreichen [27] [28]. Der Nutzer erhält auf Basis dieser Statuslevel Angebote von lokalen Unternehmen. Zudem existiert ein Check-in Leaderboard [29], das einen Wettstreit um die meisten Check-ins - ähnlich wie bei Foursquare - fördert.

2.3.2 Datenschutz

Durch die Auswahlmöglichkeiten (rechts Abbildung 2.5) wird eingestellt, wie frequent der eigene Standort erkannt und geteilt wird.

Abbildung 2.6: Latitude Screenshot (Quelle: http://www.google.com/intl/de/mobile/latitude/)

Durch die weitergehenden Optionen in Abbildung 2.6 wählt der Nutzer, welche Kontakte seine aktuelle Position sehen können. Die Möglichkeit, den Aufenthalt an spezifischen Standorten nur für ausgewählte Kontakte freizugeben, fehlt bei Latitude allerdings. Ein vom Dienst erkannter Standort ist also für alle freigegebenen Kontakte einsehbar. Andere Personen sind nicht in der Lage, einen Nutzer an einem Ort zu markieren, weshalb für diesen Anwendungsfall kein Schutz gewährleistet werden muss.

Google verspricht dabei, dass jeweils nur der aktuelle Standort gespeichert wird [30], wobei dies allerdings nicht mehr der Fall ist, wenn eingecheckt wird (um Beispielsweise Check-in Punkte zu sammeln), oder der Nutzer explizit die Location-History aktiviert.

Zudem bietet Google mit dem Dashboard[5] jedem Nutzer einen schnellen und unkomplizierten Einblick, welche Daten das Unternehmen speichert. Darüber hinaus ist in der Datenschutzerklärung[6] noch detaillierter und ausführlicher aufgeführt, welche Daten es sammelt.

2.4 Loopt

Loopt ist ein LBSNS, das sich darauf fokussiert seinen Nutzern die aktuellen Aufenthaltsorte und Aktivitäten von Freunden mitzuteilen. Dieser Dienst erlaubt es, sich mit FB und Twitter zu verbinden und so Freunde zu finden und Statusmeldungen automatisiert posten zu lassen. Aktuell besitzt Loopt mehr als fünf Millionen Nutzer [31], wurde allerdings im März 2012 von der Green Dot Corporation übernommen [32] und wird in den folgenden Monaten geschlossen werden [33].

2.4.1 Überblick

Abbildung 2.7: Loopt Screenshots (Quelle: androidxmarket.com)

Loopt wartet, im Gegensatz zu Foursquare und Google Latitude, nicht mit Punkt-Systemen für Check-ins auf, sondern wirbt damit, Insiderinformationen zu bestimmten Orten schnell und unkompliziert zur Verfügung zu stellen. Es stellt zu diesem Zweck kurze Fragen sobald eingecheckt wurde, etwa ”Is there a long line? What’s the best happy hour special?” [34]. Dies bietet zwar keine revolutionäre neue Funktionalität, allerdings ist durch die Kürze der so erhaltenen Bewertungen und Kritiken eine signifikante Verbesserung der Nutzerfreundlichkeit im Vergleich zu anderen Online-Bewertungsportalen, z.B. Qype[7], gegeben.

Darüber hinaus bietet Loopt, wie in Abbildung 2.7 rechts zu sehen, Deals an, bei denen Loopt-Partner Gegenstände verschenken oder Rabatte anbieten.

5 www.google.com/dashboard
6 http://www.google.com/policies/privacy/ in der Version vom 1. März 2012
7 http://www.qype.com/

Die Standardfunktionalität - Freunde zu sehen und zu finden - ist durch ”Pings” aufgewertet. Mit dieser Funktion kann der Nutzer Freunde ”anpingen” um herauszufinden, an welchem Ort derjenige sich gerade aufhält. Zudem wird ihm, wenn sich Freunde in der Nähe aufhalten, eine Informationsmeldung zu ihrem aktuellen Standort angezeigt.

2.4.2 Datenschutz

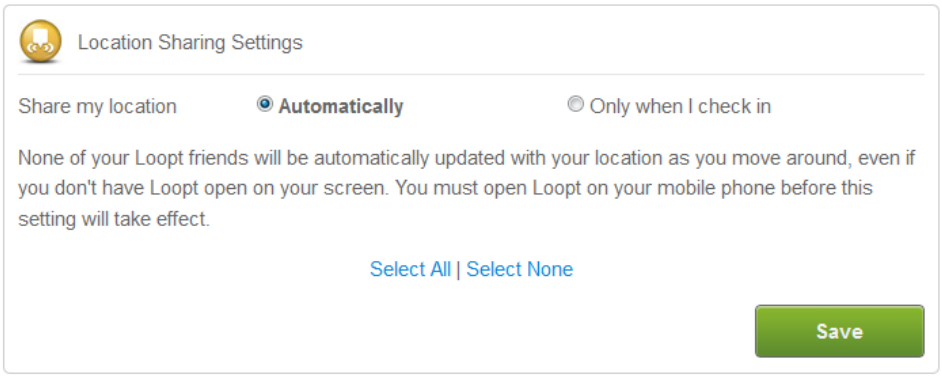

Abbildung 2.8: Loopt Settings (Quelle: https://www.loopt.com/user/settings)

Die einzige Auswahlmöglichkeit für den Datenschutz ist bei Loopt dadurch gegeben, dass der Nutzer einstellen kann, ob der aktuelle Ort automatisch aktualisiert wird oder ob explizit eingecheckt werden muss. Standardmäßig ist eingestellt, dass automatische Aktualisierungen des Ortes stattfinden, wie in Abbildung 2.8 zu sehen ist.

Darüber hinaus ist es nicht möglich einzuschränken welche Freunde den Standort eines Nutzers einsehen können. Durch den ”Ping” kann allerdings die Position nur mit einer bestimmten Person geteilt werden.

Die Informationsmeldung über den Aufenthalt von Freunden in der Nähe ist datenschutzkritisch einzuschätzen. Auf diese Weise wird ein Nutzer faktisch gezwungen zu verfolgen, an welchen exakten Orten sich Freunde in der Nähe aufhalten.

2.5 Zusammenfassung

Zusammenfassend wird deutlich, dass viele LBSNS dem Nutzer und seinen Wünschen nach Datenschutz nicht ausreichend Beachtung schenken. Weder die Standardeinstellungen zum Schutz der Privatsphäre sind ausreichend, noch wird eine Art des Schutzes der Daten gegenüber den Betreibern zur Verfügung gestellt. Zudem ist auch beim technischen Schutz der Daten teilweise nicht genug Sicherheit gewährleistet, sodass unter anderem FB einige Sicherheitslücken in seinen mobilen Anwendungen beheben musste [35] [36].

Die soeben beschriebenen Probleme lassen sich zum Großteil durch die Anwendung des Prinzips der *Dataminimization* [3] verhindern. Hierbei besteht das Ziel darin, lediglich solche Daten zu sammeln, die die Funktionen des Dienstes benötigen, und auch diese nur so lange wie nötig zu speichern. Jegliche gesammelten Daten bergen zudem kein oder nur ein minimales Sicherheitsrisiko, unabhängig davon wie umfangreich sie sind, solange sie anonym gespeichert und

aggregiert werden. Die in 1.2.2 aufgeführten privaten Informationen, die aus den Lokationsinformationen gewonnen werden können, lassen sich auf diese Weise aufgrund der Anonymität nicht mehr ableiten.

Aus diesem Grund wird das Ziel des hier entwickelten LBSNS darin bestehen, die Menge der gesammelten nutzerspezifischen Daten zu minimieren und diese anonym zu aggregieren, um somit die Angriffsflächen bestmöglich zu reduzieren. Es existieren jedoch auf dem Markt bereits viele sichere Anwendungen, von denen jedoch die wenigsten von einer breiten Masse genutzt werden, da häufig ein höheres Maß an Sicherheit eine verringerte Benutzerfreundlichkeit impliziert. Dieses Phänomen ist dabei nicht auf mobile Anwendungen beschränkt [37]. Aufgrund dieser Defizite wird in der vorliegenden Arbeit der Hauptfokus darauf gelegt, Datenschutz und Benutzerfreundlichkeit zu vereinen.

3 Events Around Me

In den vorigen Kapiteln wurde zunächst die Sensibilität von Lokationsinformationen und die entstehenden Bedrohungen für die Privatsphäre, die bei der Verarbeitung von diesen auftrefen, dargestellt. Darauffolgend wurden einige weit verbreitete LBSNS beschrieben und bzgl. ihrer Datenschutzmechanismen analysiert.

In diesem Kapitel wird ein neues LBSNS namens *Events Around Me* präsentiert, welches im Rahmen dieser Arbeit entwickelt wurde. Dabei liegt das Hauptaugenmerk, neben dem verantwortungsvollen Umgang mit den sensiblen Daten der Nutzer, auf der Benutzerfreundlichkeit.

Events Around Me stellt dabei einen Veranstaltungskompass dar, der es den Nutzern ermöglichen soll, die Anzahl der Personen an einem bestimmten Ort herauszufinden, und zu erfahren, wie sie diesen Ort beurteilen. Implementiert wird sowohl ein Server als auch, unter der Verwendung der Android Plattform, ein Client.

3.1 Motivation

Trotz der bereits erläuterten Probleme bietet die zunehmende Verbreitung von Smartphones innovative Chancen und Möglichkeiten. Aus diesem Grund ist es sinnvoll, die Verwendung ortsbasierter Dienste weiter zu perfektionieren, um von diesen Vorteilen profitieren zu können ohne die Sicherheit der Daten des Nutzers zu gefährden.

Es existieren noch weit mehr interessante Ideen und Konzepte für LBSNS, einige Beispiele wurden bereits in 1.4 aufgeführt. Lediglich der Datenschutz vieler verbreiteter Anwendungen wird vernachlässigt und so zahlt der Nutzer, selbst bei der Verwendung eines kostenlosen Dienstes, stets mit seinen wertvollen und sensiblen Daten.

Aus diesem Grund besteht das Ziel dieser Arbeit darin, ein LBSNS zu entwickeln, das einerseits interessante Nutzungsmöglichkeiten bietet, und andererseits jedem Anwender die Funktionalität in vollem Umfang zur Verfügung stellt, ohne dass seine Privatsphäre beeinträchtigt wird. Als zusätzliches Ziel wird Wert auf die Benutzerfreundlichkeit gelegt, da viele sichere Anwendungen existieren, von denen aber die wenigsten von einer breiten Masse genutzt werden, da häufig ein höheres Maß an Sicherheit eine verringerte Benutzerfreundlichkeit impliziert [38].

3.2 Beschreibung der Anwendung

Mit Hilfe der Applikation *Events Around Me* soll der Nutzer befähigt werden, Veranstaltungen in seiner Nähe einzusehen und ein kurzes und schnelles Feedback zu der aktuellen Situation an diesem Ort zu bekommen.

Zunächst entsteht das Problem, die kritische Masse zu erreichen. Dabei muss allerdings nicht auf die APIs von FB oder Twitter zugegriffen werden, da für unseren Dienst durch die zugesicherte Anonymität keine Anmeldung nötig ist. Die Zusammenarbeit mit diesen Diensten ist des Weiteren nicht geplant, da sich das Ziel einer datenschutzfreundlichen Anwendung nicht realisieren lässt, wenn für die im Hintergrund laufenden Systeme keine Datenschutzfreundlichkeit gewährleistet werden kann. Somit ist es nicht möglich etwaige Hilfestellungen zu nutzen, um die kritische Masse zu erreichen.

Der besondere Nutzen von *Events Around Me* besteht im Wesentlichen aus zwei Komponenten. Erstens haben Nutzer der Android App die Möglichkeit, der Anwendung ihren aktuellen Aufenthaltsort mitzuteilen, und zudem weitere Angaben zu machen. Diese Funktion wird im weiteren Verlauf als *Check-in* bezeichnet. Erwähnt werden sollte an dieser Stelle, dass die Funktion, bestimmten Personen den eigenen Aufenthaltsort mitzuteilen, außerhalb des Aufgabenbereichs des *Check-ins* liegt. Für diesen Einsatzzweck müssen die Dienste anderer Anbieter herangezogen werden, die bereits implementiert sind oder sich noch im Entwicklungsstatus befinden.

Zweitens können Nutzer in der Android Applikation oder auf der Website einsehen, welche Informationen andere Nutzer gemeldet haben. Dies ist möglich, indem sie sich die Karte anzeigen lassen, auf welcher Markierungen an den Orten gesetzt sind, an denen andere Nutzer eingecheckt haben. Dabei bleibt verborgen, welche Personen diese *Check-ins* getätigt haben, um die Anonymität zu wahren.

Die Anwendung zielt auf eine junge Zielgruppe, die mit einem Smartphone und solchen Diensten (LBS) aufwachsen und zudem häufig spontan und kurzfristig entscheiden, an welchem Ort sie ihre Zeit verbringen. *Events Around Me* soll ihnen hier eine Hilfe zur Entscheidungsfindung bieten.

3.3 Überblick Architektur

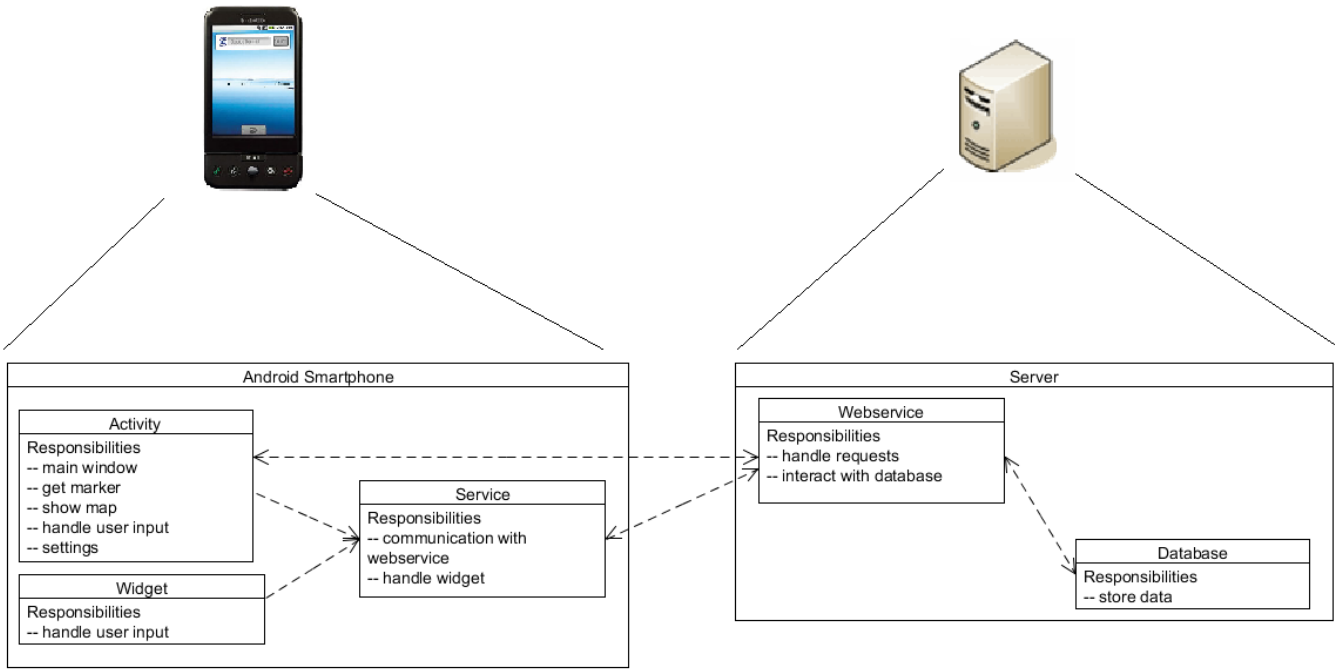

Abbildung 3.1: Übersicht Architektur von *Events Around Me*

Abbildung 3.1 gibt einen Überblick über die Architektur von *Events Around Me*. Auf dem Client werden die betreffenden Informationen gesammelt, die für das erfolgreiche Melden eines Standorts an den Server nötig sind. Die Übertragung der Daten zum Server ist durch die Nutzung von Hypertext Transfer Protocol Secure (HTTPS) und einer zusätzlichen RSA Verschlüsselung gesichert. Der Server empfängt die Daten, verarbeitet sie weiter und sendet eine Antwort zurück, wobei bei der Verarbeitung stets alle Check-ins solange anonym in einer Datenbank gespeichert

bleiben, wie sie aktiv sind. Ein Check-in wird inaktiv, sobald ein Nutzer einen neuen Check-in getätigt hat bzw. der Check-in nach 3 Stunden abgelaufen ist.

Darüber hinaus lässt sich zwischen den Client und den Server ein Anonymisierungsproxy schalten, da der Server keine der Informationen über das Gerät des Clients, die normalerweise bei jeder Standard Hypertext Transfer Protocol (HTTP) Anfrage (TCP) mitgesendet werden, für seine Berechnungen benötigt. Unser System basiert somit auf einer klassischen Client-Server Architektur, mit einer zentralen Datenbank. Der Client ist, Android entsprechend, in Java implementiert. Der Server wurde in Ruby implementiert.

Zunächst folgt eine detailliertere Beschreibung der Android Anwendung, anschließend wird die Server Anwendung erläutert und darauffolgend werden mögliche Schwachstellen von *Events Around Me* aufgelistet. Resultierend lassen sich im weiteren Verlauf Rückschlüsse auf Verbesserungsmöglichkeiten ziehen.

3.4 Android Anwendung

Die Android Anwendung soll die in Kapitel 3.4.1 genannten Anwendungsfälle realisieren sowie eine passende Schnittstelle zum Server implementieren.

3.4.1 Architektur

Android Version

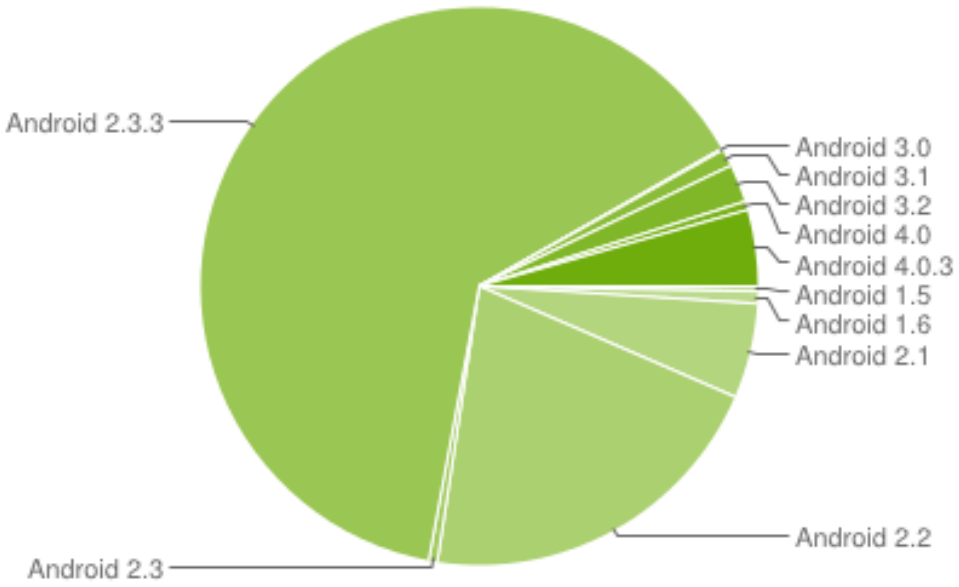

Abbildung 3.2: Verbreitung von Android Betriebssystemen im Mai 2012 (Quelle: http://developer.android.com /resources/dashboard/platform-versions.html)

Als Entwicklungsplattform wird Android in der Version 2.3.3[1] genutzt. Dies ist dadurch begründet, dass Android eines der derzeit am weitesten verbreiteten Smartphone Betriebssysteme

[1] http://developer.android.com/sdk/android-2.3.3.html

darstellt und in den letzten Monaten einen rasanten Aufstieg erfahren hat, dessen Ende noch nicht absehbar ist, wie Abbildung 1.1 bereits zu entnehmen war. Die Version 2.3.3 wurde gewählt, da sowohl die Base64 Kodierung, als auch die UTF-8 Kodierung auf Android erst ab dieser Version standardmäßig verfügbar sind, und beide zur Verschlüsselung der Kommunikation benötigt werden. Zudem besitzen, wie auf Abbildung 3.2 erkennbar, aktuell ca. 75% der Android Smartphones ein Betriebssystem höher als 2.3.3 und sind somit für eine solche Anwendung kompatibel. Die Verbreitung der Version 4 ist aktuell noch zu gering, um eine Anwendung, die viele Nutzer erreichen soll, auf Basis dieser Plattformversion zu entwickeln.

Anwendungsfälle

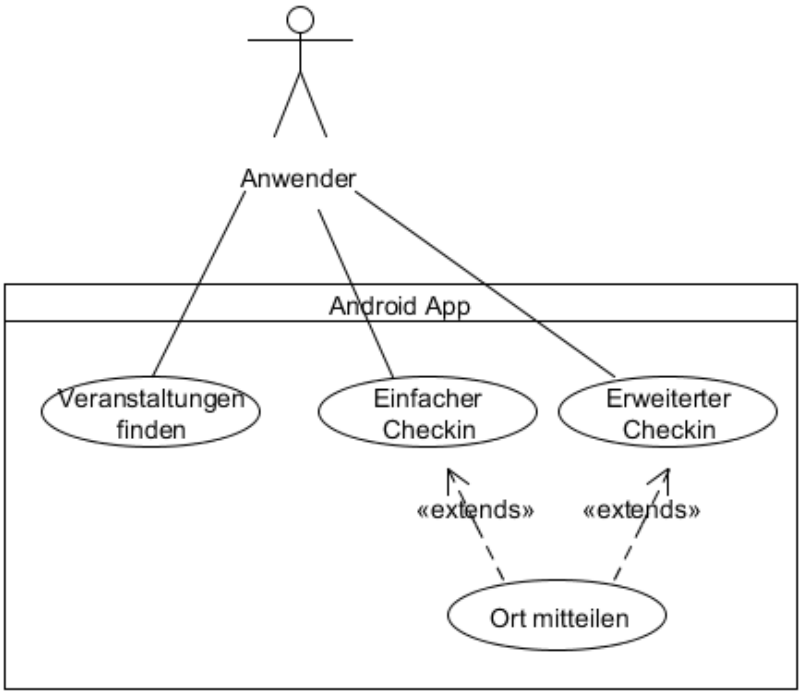

Abbildung 3.3: Anwendungsfälle im Rahmen von *Events Around Me*

In Abbildung 3.3 ist ein Use-Case Diagramm zu sehen. Die Einteilung in die beiden Anwendungsfälle ”Veranstaltungen finden” und ”Ort mitteilen” wurde in 3.2 bereits kurz beschrieben.

Die Möglichkeit einen Ort mitzuteilen wird in zwei speraten Anwendungsfällen betrachtet:

Im ersten Fall, dem ”einfachen Check-in”, möchte der Nutzer nur eine schnelle Meldung versenden, an welchem Ort er sich gerade befindet. Aus diesem Grund möchte er hier keine weiteren Optionen angeboten bekommen.

Im zweiten Fall, dem ”erweiterten Check-in”, legt der Nutzer nur bedingt Wert auf die Effizienz des Vorgangs und möchte noch weitere Informationen zu seinem aktuellen Aufenthaltsort angeben. Daher besitzt er die Möglichkeit an den Check-in folgende zusätzliche Informationen anzuheften (siehe Abbildung 3.4 links):

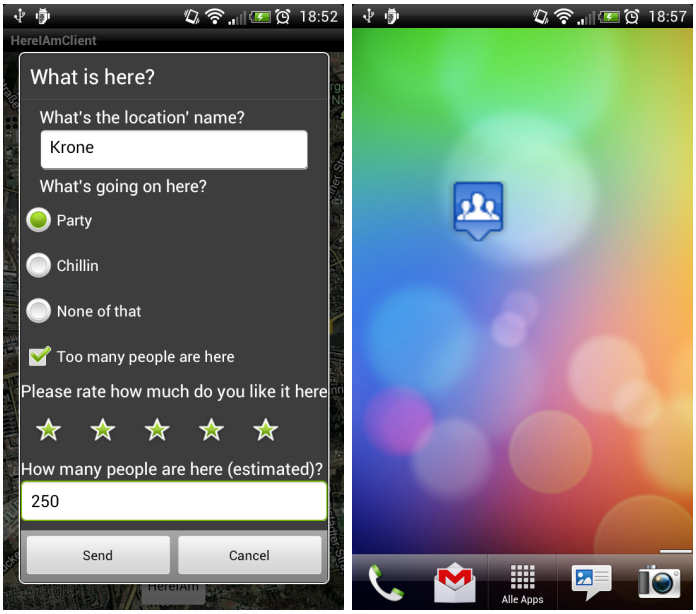

Abbildung 3.4: Screenshots zum Check-in der Android Anwendung *Events Around Me*

- Name des Ortes
- Beschreibung des Ortes
- Angabe, ob sich derzeit zu viele Menschen an diesem Ort aufhalten
- Bewertung des Ortes
- Anzahl der Personen, die sich an diesem Ort schätzungsweise aktuell aufhalten

Um den einfachen Check-in weiter zu beschleunigen, wird im Rahmen der Applikation ein Widget[2] zur Verfügung gestellt, das durch eine simple Berührung bereits einen Check-in ausführt und darüber hinaus auf der Startseite eines Handys abgelegt werden kann (siehe Abbildung 3.4 rechts).

Der weitere Anwendungsfall ”Veranstaltungen finden” behandelt die Funktion, dass der Anwender die Karte öffnet und sich dort anzeigen lässt, wieviele Nutzer sich an einem bestimmten Ort gemeldet haben. Diese Funktion ist in Abbildung 3.5 verdeutlicht. Der linke Screenshot zeigt den Bildschirm nach dem Starten der App. Auf der Karte sind an allen Orten, an denen sich Nutzer gemeldet haben, Markierungen gesetzt. Unter jeder Markierung wird der Name des Ortes angezeigt, sofern er angegeben wurde oder der Anwendung der Ort bekannt ist. Berührt der Nutzer an dieser Stelle eine der Markierungen, öffnet sich ein Informationsfenster, welches auf dem rechten Bild von Abbildung 3.5 zu sehen ist. Die Eingaben, die von Nutzern beim erweiterten Check-in gemacht wurden, werden hier bedarfsgerecht aufgearbeitet und angezeigt.

Aufbau der Applikation

In Abbildung 3.6 ist zu erkennen, dass die Android Applikation im Wesentlichen aus drei Teilen besteht:

- Activity

[2] http://developer.android.com/guide/topics/appwidgets/index.html

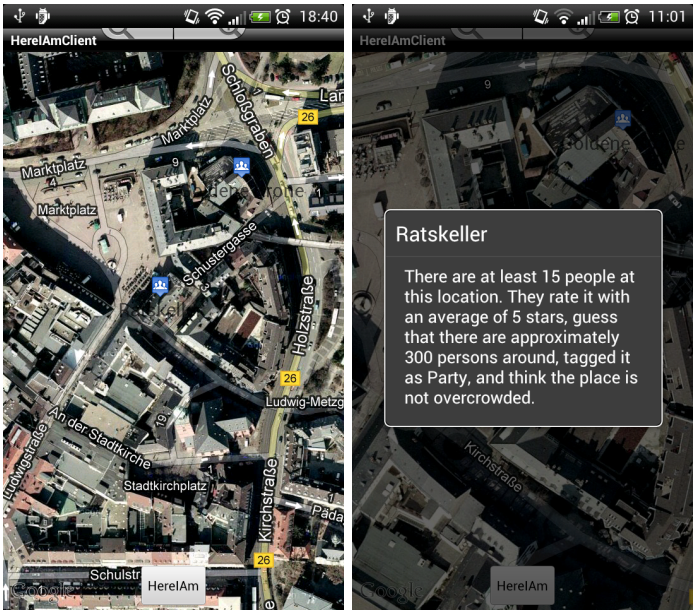

Abbildung 3.5: Screenshots zur Karte der Android Anwendung *Events Around Me*

- Widget
- Service

Die *Activity* stellt hierbei die Oberfläche, die in Abbildung 3.5 zu erkennen ist, für den Nutzer zur Verfügung. Wenn der Nutzer dieses Fenster öffnet, wird ihm zunächst seine nähere Umgebung auf der Karte angezeigt. Falls in dieser Veranstaltungen gemeldet sind, werden auch die entsprechenden Markierungen plaziert. Mit Hilfe des Buttons *HereIAm* am unteren Bildrand kann der Nutzer einen Check-in ausführen. Sofern in den Einstellungen der erweiterte Check-in aktiviert ist, zeigt die Activity zunächst das erweiterte Check-in-Formular aus Abbildung 3.4 an.

Sobald der Nutzer bei geöffneter Activity auf seinem Smartphone den *Menü*-Button betätigt, erhält er die in Abbildung 3.7 links zu sehenden vier Auswahlmöglichkeiten. Beim Berühren des *Refresh*-Buttons wird die Karte neu geladen. Der *About*-Button öffnet die im rechten Screenshot in Abbildung 3.7 zu sehende Beschreibung der Anwendung. Die Betätigung des *Stop App*-Buttons erlaubt es dem Nutzer, den Prozess zu beenden. Lediglich ein gestarteter Check-in kann hier noch im Hintergrund ablaufen. Der *Settings*-Button öffnet ein neues Fenster, in dem verschiedene Auswahlmöglichkeiten vorhanden sind (Abbildung 3.7 Mitte). Der Nutzer kann an dieser Stelle den beschriebenen erweiterten Check-in aktivieren, einen Filter einstellen, sodass ihm ausschließlich Veranstaltungen angezeigt werden, die er wünscht, und einen Anonymisierungsproxy zu aktivieren (nicht implementiert). Darüber hinaus kann er sich den Kommunikationslog anzeigen lassen, woraus eine zusätzliche Transparenz resultiert (nicht implementiert).

Das *Widget* wurde bereits erwähnt und ist in Abbildung 3.4 zu sehen. Durch eine kurze Berührung dieses Symbols ist es dem Nutzer möglich, einen einfachen Check-in an seinem aktuellem Ort durchzuführen.

Der *Service* stellt bei *Events Around Me* den Großteil der Funktionalität zur Verfügung. Alle Interaktionen des Nutzers werden vom Widget und der Activity an den Service weitergeleitet,

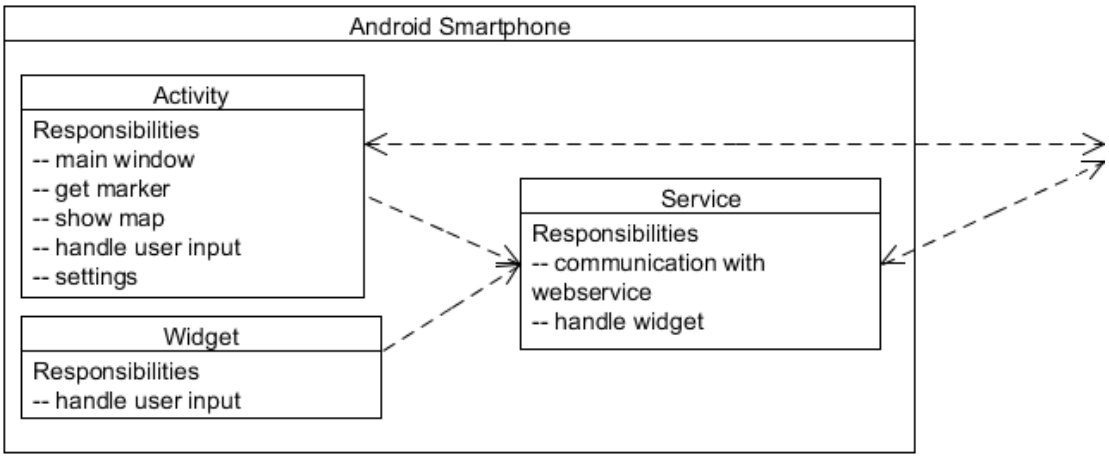

Abbildung 3.6: Aufbau der Android Applikation *Events Around Me*

der die entsprechende Aktion durchführt. Nähere Informationen disbezüglich sind im folgenden Abschnitt aufgeführt.

Programmablauf

In Abbildung 3.8 ist der Ablauf des Handshakes visualisiert. Der Webservice generiert beim Starten ein RSA Keypair. Sobald vom Nutzer die Activity - die Oberfläche - geöffnet wird, startet diese den Service, der einen RSA Public Key vom Server fordert. Dieser wird zunächst in einer lokalen Datei gespeichert und dann als Public Key eingelesen. Diese Datei bleibt im internen Speicher des Systems, auf den nur die Anwendung zugreifen kann, gespeichert. Dieser Handshake muss folglich nur einmal pro Client gestartet werden.

Wie in Abbildung 3.9 zu sehen, wird die Karte geladen, sobald der Nutzer die Anwendung bzw. die Activity öffnet. Das Laden der Markierungen ist der einzige Fall, in dem die Activity selbst mit dem Webservice kommuniziert, da der im Hintergrund laufende Service keine Möglichkeit besitzt, auf die in der Activity angezeigte Karte zuzugreifen. Der Webservice fordert von der Datenbank alle Lokationen an, an denen eingecheckt wurde, um anschließend diese Daten an die Activity weiterzureichen. Erst zu diesem Zeitpunkt kann die Activity durch diese Informationen die Karte mit Markierungen füllen.

Nach dem Schlüsselaustausch kann der Nutzer jederzeit einen einfachen oder erweiterten Check-in initiieren. In Abbildung 3.10 ist beispielhaft der Programmfluss des erweiterten Check-ins dargestellt. Dieser muss stets vom Nutzer gestartet werden, der daraufhin das in Abbildung 3.4 links dargestellte Formular ausfüllt, um weitere Daten anzugeben. Die auf diese Weise erlangten Informationen reicht die Activity an den Service weiter. Dieser fügt den Daten anschließend den per GPS oder Netzwerk erfassten aktuellen Aufenthaltsort hinzu, und sendet diese Informationen verschlüsselt an den Webservice. Die Prozesse, die auf dem Server ablaufen, werden später genauer beschrieben; an dieser Stelle abstrahieren wir von diesen und betrachten lediglich die Antwort des Servers, in der er den erfolgreichen Check-in bestätigt.

Es wird an dieser Stelle eine Möglichkeit zur Identifizierung benötigt, da beachtet werden muss, dass ein Nutzer nicht mehrmals eingecheckt sein darf. In der Version von *Events Around Me* vor der Abschlusspräsentation am 05.07.2012 generiert der Client bei jedem Check-in eine

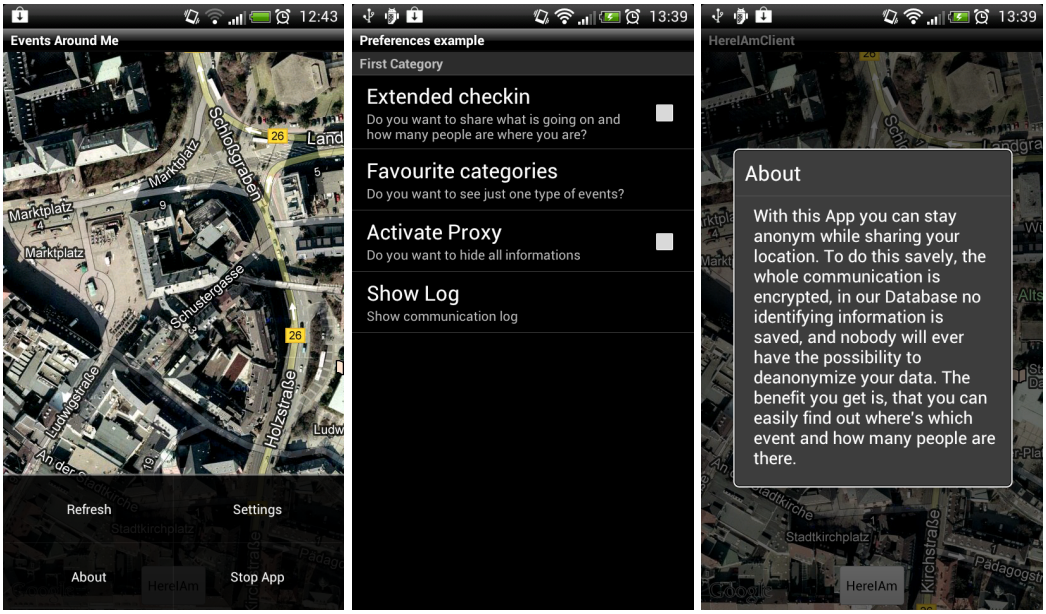

Abbildung 3.7: Screenshots zum Menü der Android Anwendung *Events Around Me*

neue Universal Unique-ID (UUID)[3]. Die alte UUID ist im internen Speicher der Anwendung gespeichert und wird mit jedem neuen Request ebenfalls verschlüsselt mitversendet. Auf diese Weise ist es dem Server möglich, den alten Check-in, sofern er noch vorhanden ist, zu entfernen, bevor der neue hinzugefügt wird. In der aktuellen Version, ist die Identifizierung über eine ID jedoch nicht mehr notwendig. Hier wird eine Check-out Funktion zur Verfügung gestellt, sodass das Smartphone lediglich die Koordinaten des letzten Check-ins speichert, und diese bei einem neuen Check-in mitübertragen werden. So kann letztlich verhindert werden, dass durch eine Verkettung der IDs und Standortinformationen Rückschlüsse auf das Bewegungsprofil des Anwenders möglich sind. Diese Maßnahme unterstützt somit die Anonymität.

Folgende Daten teilt der Client dem Server bei jedem Check-in mit:

- *Old Lat*
- *Old Long*
- *ID*
- *latitude*
- *longitude*
- *name*: Name, mit dem der Nutzer diesen Ort bezeichnet
- *tag*: Tags, mit denen der Nutzer die Veranstaltung an diesem Ort beschreibt
- *full*: Boolean, ob es sich bereits zu viele Personen an diesem Ort aufhalten
- *rating*: Integer (0-5), Bewertung dieses Orts durch den Nutzer
- *estimated num*: Integer, der die Schätzung des Nutzers angibt, wie viele Personen sich aktuell an diesem Ort aufhalten

[3] http://docs.oracle.com/javase/6/docs/api/java/util/UUID.html

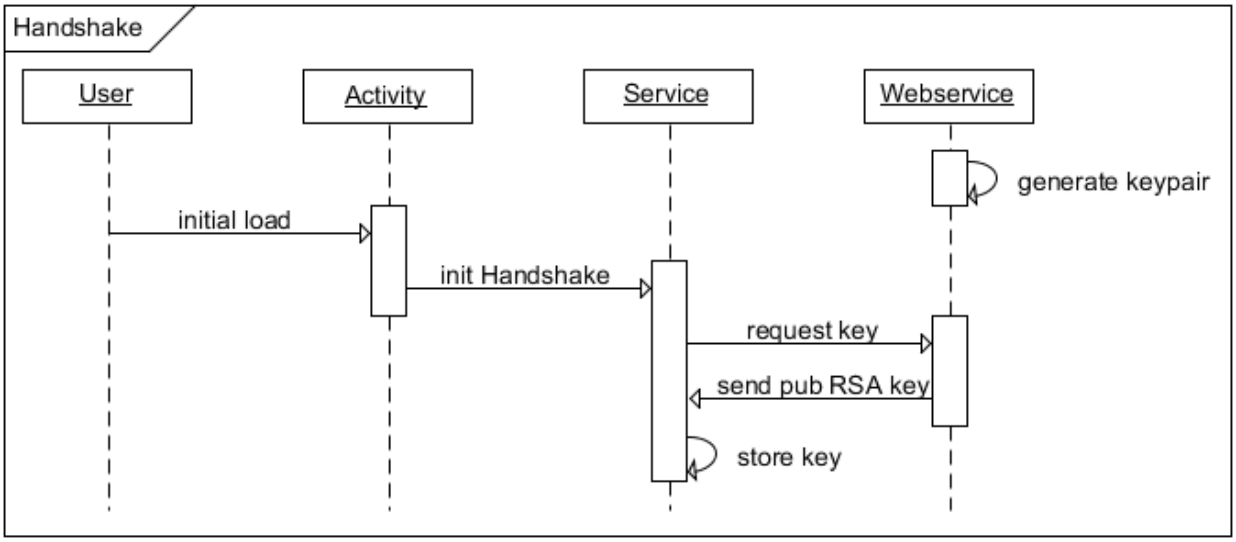

Abbildung 3.8: Handshake bei *Events Around Me*

- *GPS*: Boolean, ob das Signal per GPS empfangen wurde

Die beiden IDs wurden bereits erläutert, während die Felder *Old Lat* & *Old Long* bzw. *latitude* & *longitude* zur Darstellung der Koordinaten des letzten/aktuellen Check-ins dienen.

3.4.2 Implementierung

In diesem Abschnitt werden die wichtigsten genutzten Bibliotheken und einige Schwierigkeiten, die bei der Entwicklung der Android Anwendung aufgetreten sind, sowie der Umgang mit diesen, beschrieben.

Genutzte Bibliotheken sind unter anderem:

- **java.security**: RSA-Verschlüsselung der Daten
- **org.apache.http**: HTTP-Kommunikation
- **android.location**: Ortung
- **android.widget**: Widget
- **android.service**: Service
- **anroid.activity**: Activity
- **android.preference**: Einstellungen
- **com.google.android.maps**: Map Darstellung
- **cwac-locpoll**: Wakeful Thread

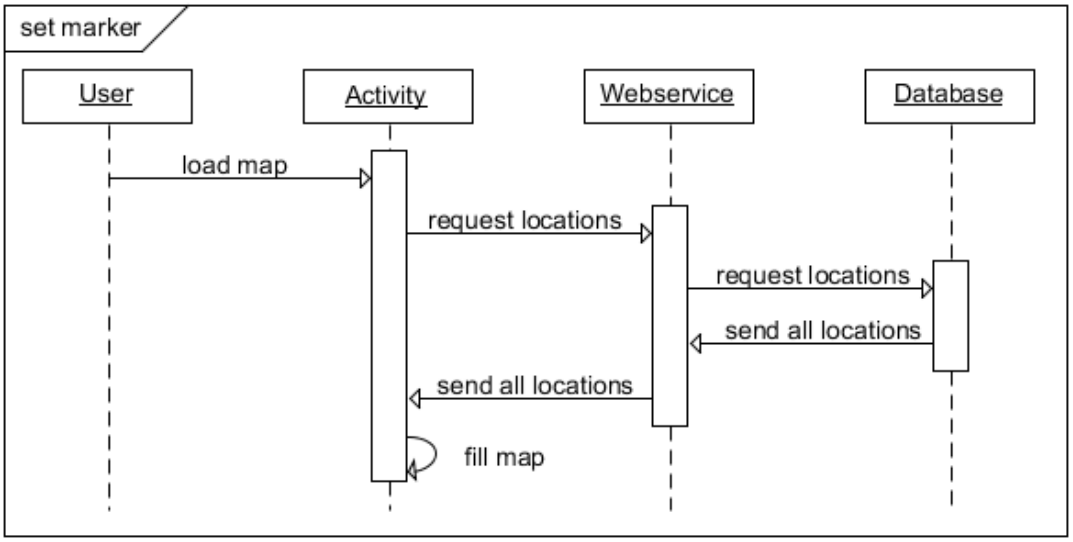

Abbildung 3.9: Refresh der Karte bei *Events Around Me*

Es wurde bereits beschrieben, dass die Android Anwendung auf einem Service, einem Widget und einer Activity basiert. Ein Service ist ein Dienst, der es erlaubt, Prozesse im Hintergrund ablaufen zu lassen ohne der Kenntnis des Anwenders. Der Nutzer kann, während der Service (der im Rahmen einer Anwendung gestartet wurde) im Hintergrund seine Arbeit verrichten, ohne Einschränkung mit anderen Anwendungen interagieren.

Eine Activity stellt eine einzelne Oberfläche dar, mit der der Nutzer interargieren kann. Diese kann in den Hintergrund treten, sodass sie jederzeit fortgesetzt werden kann, wenn sie wiederhergestellt wird.

Ein Widget bietet eine Art Abkürzung für den Nutzer. Es ermöglicht vom Home Screen direkten Zugang zu Funktionen der Anwendung, ohne zunächst die Activity starten zu müssen.

Die Ortung verläuft bei Android durch Location Listener. Nach deren Registrierung wird, sobald ein Update bzgl. des Ortes empfangen wird, die entsprechende Listenerfunktion ausgeführt. Es ist möglich, verschiedene Listener für verschiedene Location Provider zu erstellen. So kann beispielsweise unterschieden werden, ob ein Locationupdate per Netzwerk oder per GPS empfangen wurde. In Abschnitt 1.2.1 wurde bereits verdeutlicht, dass die Ortung via GPS derzeit die beste Möglichkeit bietet, den Standort eines Nutzers exakt zu erfassen. Da sich jedoch für die Ortung mittels GPS vier der Satelliten in Sichtlinie befinden müssen, entfällt diese Option an vielen Orten (z.B. innerhalb von Gebäuden). Um den Nutzern zu ermöglichen, ihren Aufenthaltsort auch ohne GPS Empfang zu melden, wird im Falle der Nichtverfügbarkeit des GPS Signals, der Standort über WiFi-Verbindungen oder Mobilfunknetz Triangulation berechnet. Diese Option bietet allerdings eine deutlich geringere Genauigkeit. Daher wird die Ortung via GPS bevorzugt, sodass lediglich, wenn diese mehr als 45 Sekunden benötigt, die anderen Optionen Verwendung finden. Die Schranke wurde auf 45 Sekunden bestimmt, da nach einigen Stichproben festgestellt werden konnte, dass selbst bei einem Aufenthalt unter freiem Himmel mindestens 30 Sekunden benötigt werden um eine valide GPS Position bestimmen zu können. Daher muss diese Schwelle größer als 30 Sekunden sein, darf dabei allerdings auch nicht zu groß und damit zu ineffizient und benutzerunfreundlich sein.

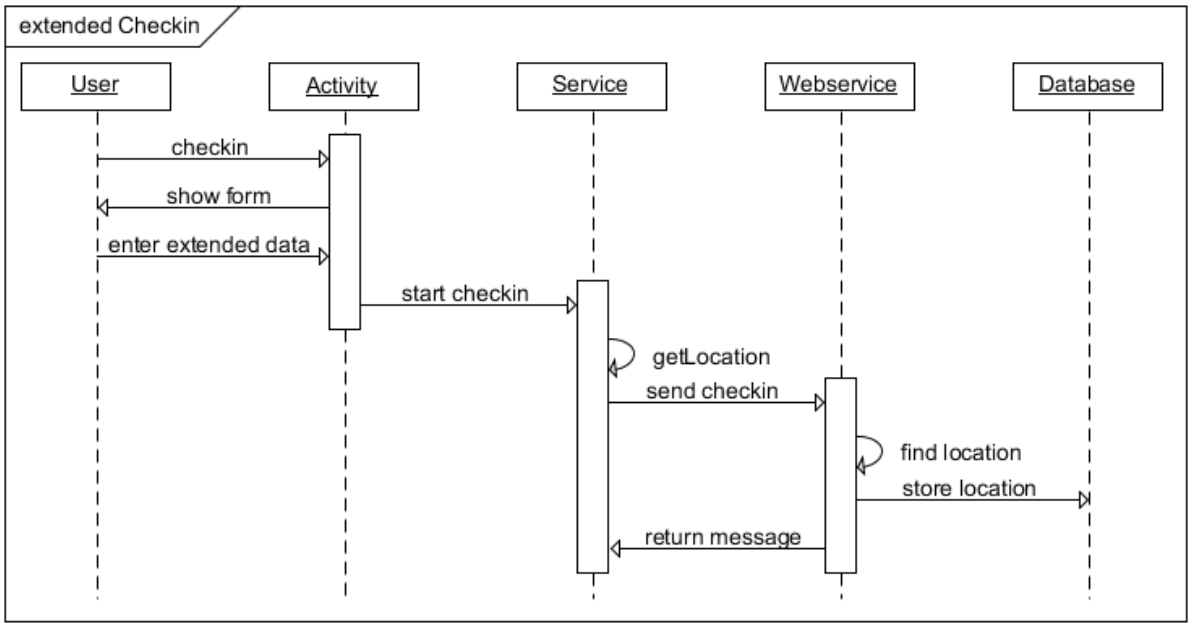

Abbildung 3.10: Erweiterter Check-in bei *Events Around Me*

Die bei den Einstellungen gewählten Präferenzen werden ebenso wie die *Alten Koordinaten* des letzten Checkins im Bereich der *Shared Preferences* persistent gesichert. Sie sind somit auch dann noch verfügbar, wenn die Anwendung abgestürzt ist. Der Zugriff ist dabei - durch das zugrundeliegende Linux System - auf die Anwendung beschränkt.

Die in Abbildung 3.5 abgebildete Karte innerhalb der Activity wurde mittels der Google Maps API[4] eingebunden. Da diese Bibliothek Zugriff auf die Daten von Google gewährt, ist es zunächst notwendig, dass der Entwickler einen Account bei Google erstellt und einen Google Maps API Key registriert. Ausschließlich mit einem solchen Key kann der Download der Karte erfolgen. Er wird benötigt, um die Anwendung valide zu signieren, da jede Android Anwendung, bevor sie auf einem Gerät installiert wird, signiert werden muss. Der hierbei genutzte Key kann ausschließlich zu Debug Zwecken verwendet werden. Für einen möglichen offiziellen Release ist die Registrierung eines neuen Keys erforderlich.

Im Abschnitt 3.4.1 wurde der Programmfluss innerhalb des Programms erläutert. Dabei ist der Service für das Abrufen der aktuellen Position zuständig. Da dieser im Hintergrund läuft, die Logik zur Ortung jedoch ein ”waches” Smartphone benötigt, ist sicherzustellen, dass das Smartphone nicht in den Schlafmodus fällt. Zu diesem Zweck wird hier auf die Open-Source Bibliothek cwac-locpoll[5] in der Version vom 05.05.2012 zurückgegriffen, dessen Klasse *WakefulThread* es ermöglicht, einen Thread zu starten, der die Aktivierung des Geräts gewährleistet. Diese Bibliothek wurde unter der Apache Lizenz, Version 2.0[6] entwickelt.

[4] https://developers.google.com/maps/documentation/android/
[5] https://github.com/commonsguy/cwac-locpoll
[6] http://www.apache.org/licenses/LICENSE-2.0

3.5 Server Anwendung

Wie bereits in Abschnitt 3.3 beschrieben, basiert unser System auf einer klassischen Client-Server Architektur mit einer zentralen Datenbank. Jegliche Kommunikation läuft über den hierzu implementierten REST-Webservice, der auf HTTP-Request reagiert.

3.5.1 Architektur

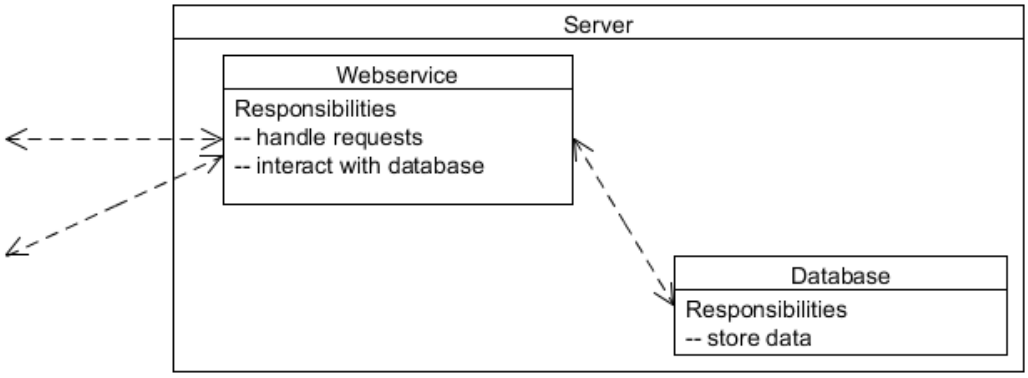

Abbildung 3.11: Aufbau der Server Applikation von *Events Around Me*

In Abbildung 3.11 ist zu sehen, dass der Server aus zwei Teilen besteht:

- Webservice (WS)
- Database

Der WS wurde in Ruby implementiert. Bei seinem Start wird zunächst das OpenSSL Zertifikat geladen, sodass der Server nur HTTPS Anfragen behandelt. Das Zertifikat besitzt dabei folgende Einträge:

Common Name: EventsAroundMe

Country: DE

State: Hessen

Location: Darmstadt

EMailAdress: dachk@web.de

Organization Unit: CASED

Organization: TU Darmstadt

Dieses Zertifikat wird ebenso innerhalb der Android Anwendung, für eine erfolgreiche HTTPS Kommunikation, benötigt. Die Client Anwendung akzeptiert außer dem hier eigens erstellten Zertifikat lediglich die Standard SSL Zertifikate, die von einer vertrauenwürdigen Certificate Authority (CA) stammen. Da allerdings zusätzlich die Standard SSL Zertifikate akzeptiert werden,

ist es notwendig, eine weitere Verschlüsselung zu implementieren, welche die Kommunikation gegen eine Man-in-the-Middle Attacke schützt [39]. Diese Verschlüsselung wurde mittels RSA umgesetzt.

Die im Hintergrund laufende Datenbank besitzt vier verschiedene Tabellen:

- Entry
- Location
- History
- Poi

Auf den genaueren Aufbau dieser wird im folgendem Abschnitt eingegangen; an dieser Stelle seien lediglich die Funktionen beschrieben. Die Klasse *Entry* speichert die Uhrzeit der aktuellen Check-ins, um zu ermöglichen, dass zu alte Check-ins gelöscht werden. Die Klasse *Location* bildet eine Gruppe ab. Falls ein neuer Check-in getätigt wird, zu dem noch keine Gruppe existiert, wird eine neue erstellt. In der Klasse *History* werden periodisch alle Location Einträge gesichert, während die Klasse *Poi* aus GPS Exchange Format (GPX) Dateien eingelesene Points of Interests (POI) speichert. Auf diese Weise können Orte automatisch mit einem Namen beschriftet werden. Sobald an einem solchen POI eingecheckt wurde, wird der Name dieses POIs übernommen.

HTTP Type	Request Name	Return Daten	Beschreibung
GET	/loadnames	-	Lade POI aus GPX Dateien in DB
GET	/init	Public RSA Key	Wird beim Handshake aufgerufen
GET	/Location	Alle Orte	Debug
GET	/Entry	Alle Einträge	Debug
GET	/History	Alle History Einträge	Debug
POST	/Entry	Alle Einträge in der Nähe	Positionierung der Markierungen
GET	/delall	-	Leere Entry und Location
POST	/Location	-	Check-in

Tabelle 3.1: HTTP Requests die vom Webservice zur Verfügung gestellt werden

Der WS stellt verschiedene HTTP Anfragen zur Verfügung. Eine Übersicht ist in Tabelle 3.1 aufgeführt.

Die Funktion /init stellt die für den Handshake benötigte Funktionalität bereit. Zusätzlich wird beim ersten Aufruf dieser Funktion ein neuer Thread (*Period Check*) gestartet, der alle 60 Minuten die Datenbank nach zu alten Einträgen überprüft (es wurde beispielhaft eine Ablaufzeit von 3 Stunden angenommen) und diese löscht, und zudem die aktuellen Einträge von allen Lokationen in der History speichert.

Die weiteren GET-Methoden aus der Tabelle sind weitestgehend selbsterklärend weshalb auf diesen nicht weiter eingegangen wird.

Die POST-Methode /Entry liefert alle notwendigen Informationen für die Markierungen auf dem Smartphone. Um letztlich mehr Anonymität zu gewährleisten, kann hier ein Schwellwert definiert werden, sodass lediglich Gruppen mit einer bestimmen Mindestanzahl an Personen angezeigt werden. So kann einerseits Missbrauch vermieden werden, und andererseits wäre es mit nur einem Check-in an einem bestimmen Ort möglich, eine eindeutige Zuordnung zu

einem Nutzer zu treffen. Beispielsweise lässt sich so auch verfolgen, zu welchen Zeitpunkten ein Haus leersteht, um es auszurauben. Dieser Schwellwert ist nicht nur statisch festgelegbar, sondern kann auch mit Hilfe der History, z.B. durch eine durchschnittliche Anzahl von Personen an diesem Ort an einem bestimmten Wochentag, dynamisch berechnet werden. Diese POST-Methode wird genutzt, um die notwendigen Daten zur Darstellung der Markierungen an das Smartphone zu übertragen. Um das Modell skalierbar zu halten, werden hierbei die aktuellen, ungefähren Koordinaten des Nutzers an den WS gesendet und nur solche Einträge übermittelt, die im Umkreis des Nutzers liegen. Derzeit werden allerdings zu Testzwecken alle Einträge übertragen.

Die POST-Methode /Location stellt die eigentliche Kern-Funktion des Webservices dar. Sie dient der Ausführung eines Check-ins. Die in 3.4.1 aufgeführten Angaben, die einen erweiterten Check-in begleiten, werden hier verarbeitet und gespeichert. Zunächst muss die Anfrage entschlüsselt werden. Im Anschluss wird versucht, den letzten Check-in des Nutzers auf Basis der *Old Lat* und *Old Long* Koordinaten zu entfernen. Danach wird überprüft, zu welcher Gruppe der neue Check-in gehört. Dazu werden die Koordinaten des Check-ins extrahiert und zunächst getestet, ob an diesem Ort bereits eine Gruppe existiert. Der Radius (*$group_range*) variiert dabei, je Genauigkeit der Ortung (GPS/Netzwerk). Falls derzeit keine Gruppe an diesem Ort existiert, muss eine neue erstellt werden. In diesem Fall wird zunächst die Gruppen ID berechnet, anschließend die Klasse Poi und dann die History nach Einträgen an diesem Ort durchsucht. Wenn eine der Suchen erfolgreich war, werden deren Koordinaten und Name übernommen und mit diesen Informationen eine neue Gruppe erstellt. Daraufhin wird ein neuer Eintrag getätigt, der der ID des Check-ins die betreffende Gruppe zuordnet, um diesen Eintrag bei einem erneutem Check-in dieses Nutzers wieder löschen zu können und somit ein Ausloggen zu ermöglichen.

Die Angaben, die bei einem erweiterten Check-in getätigt werden, dürfen teilweise nicht ohne Weiteres übernommen werden, um Missbrauch zu vermeiden. Es darf keiner einzelnen Person möglich sein, die Bewertung eines Ortes entscheidend zu beeinflussen. Daher werden die Angaben der geschätzen Anzahl an Personen an dem Ort, sowie das Rating jeweils gemittelt, um ungefälschte Aussagen präsentieren zu können und so Missbrauch zu erschweren.

3.5.2 Implementierung

Die Serveranwendung von *Events Around Me* ist in Ruby implementiert. Die Sprache alleine ermöglicht es allerdings nicht, eine Website oder einen Webservice zu implementieren. Hierzu ist es notwenig, sogenannte Gems einzubinden. Im Falle unseres WS benötigen wir Sinatra[7], DataMapper[8] und WEBrick[9], um im Zusammenspiel einen Webservice inklusive Datenbank aufzubauen.

Sinatra ist eine OpenSource Webapplikationsbibliothek für Ruby. Sie ermöglicht es, schnell und mit minimalem Aufwand eine Webapplikation zu entwickeln. Mit ihrer Hilfe wird der Server auf die eingehenden HTTPS Anfragen programmiert.

Der DataMapper ist ein ebenfalls OpenSource geführter objekt-relationaler Mapper, der sich vor allem durch seine Schnelligkeit auszeichnet. Er wird benötigt, um die relationale Datenbank zu erstellen, mit ihr zu kommunizieren und sie zu bearbeiten. Die Datenbank stellt eine Sqlite3 Datenbank mit vier Tabellen dar.

[7] http://www.sinatrarb.com/
[8] http://datamapper.org/
[9] http://www.ruby-doc.org/stdlib-1.9.3/libdoc/webrick/rdoc/index.html

ID	Time	Group
String	String	Integer

Tabelle 3.2: Entry

Die Tabelle Entry wurde nach der Präsentation dieser Arbeit angepasst. In der ursprünglichen Tabelle (Tabelle 3.2) wurden einzelne Check-ins gespeichert. Die gespeicherte ID wurde vom Client generiert und ist eine 128Bit lange UUID[10]. Diese besteht aus Zeichen und Zahlen und wurde daher als String gespeichert.

ID	Time	Group
Serial	String	Integer

Tabelle 3.3: Entry

Die neue Version der Tabelle wurde umstrukturiert, da auf diese Weise eine gesteigerte Anonymität gewährleistet werden kann. Nun wird diese Tabelle lediglich benötigt, um den *Periodic Check* weiterhin durchführen zu können. Es wird keine ID mehr mit dem Smartphone des Anwenders verknüpft, aber dennoch bei jedem Check-in ein Zeitstempel gespeichert, der die zeitliche Zuordnung eines Check-ins zu einer Gruppe erlaubt.

Group	Latitude	Longitude	Num	Tag	ToFull	Rating	Estimatednum	Name
Serial	Float	Float	Integer	String	Boolean	Integer	Integer	String

Tabelle 3.4: Location

Jede Gruppe wird wiederum in der Tabelle *Location* (Tabelle 3.4) gespeichert. Dabei sind auch ausführlichere Angaben gespeichert, die die Maximierung des Nutzens der Anwendung bewirken sollen. Durch diese Kombination von *Entry* und *Location* ist sichergestellt, dass ein Eintrag nicht eindeutig zu einer genauen Position zugeordnet werden kann (nur im Umkreis von *$group_range* einer Gruppe). Zudem ist nach dem *Period Check* oder einem erneuten Check-in eines Nutzers, der einen Logout mit der *Old ID* triggert, kein Nutzer mehr identifizierbar - selbst wenn ein Angreifer alle verfügbaren Informationen des Clients besitzt.

In Tabelle 3.5 ist die *History* abgebildet, die durch den *Period Check* stündlich mit allen gespeicherten Gruppen gefüllt wird. Bei jedem Check-in wird neben der Tabelle *Poi* auch die History nach bereits bestehenden Einträgen an diesem Ort durchsucht, um ggf. Informationen automatisch übernehmen zu können.

Die Tabelle *Poi* (Tabelle 3.6) hält die wichtigsten POIs Deutschlands gespeichert. Falls die Koordinaten eines Check-ins innerhalb des Radius eines POI liegt, werden die Koordinaten und Name dieses Eintrags für die Gruppe übernommen.

Eine weitere Ruby Programmbibliothek welche hier genutzt wird, ist WEBrick. Diese stellt einen HTTP-Webserver zur Verfügung und ist in der Standardbibliothek von Ruby enthalten. Da hier allerdings eine HTTPS-Verbindung benötigt wird, ist es notwendig, den Webserver für unsere speziellen Zwecke anzupassen. Der WEBrick Server muss daher mit einigen Optionen gestartet werden:

[10] http://docs.oracle.com/javase/6/docs/api/java/util/UUID.html

Group	Latitude	Longitude	Num	Tag	ToFull	Rating	EstimatedNum	Name	Date
Serial	Float	Float	Integer	String	Boolean	Integer	Integer	String	String

Tabelle 3.5: History

ID	Latitude	Longitude	Name
Serial	Float	Float	String

Tabelle 3.6: Poi

```
CERT_PATH = 'ca/newca/'

# Options for starting WEBRick inkl HTTPS cert
webrick_options = {
        :Port            => 8443,
        :Logger          => WEBrick::Log::new($stderr, WEBrick::Log::DEBUG),
        :DocumentRoot    => "/ruby/htdocs",
        :SSLEnable       => true,
        :SSLVerifyClient => OpenSSL::SSL::VERIFY_NONE,
        :SSLCertificate  => OpenSSL::X509::Certificate.new(File.open(File.join(CERT_PATH,
 "cacert.pem")).read),
        :SSLPrivateKey   => OpenSSL::PKey::RSA.new(File.open(File.join(CERT_PATH,
"private/cakey.pem")).read),
        :SSLCertName     => [ [ "CN",WEBrick::Utils::getservername ] ]
}
```

Mit diesen wird der Server für die SSL Verschlüsselung eingerichtet, wofür ihm das benötigte Zertifikat mitgeteilt wird.

Um die GPX-Dateien (POI) zu parsen, wurde die Bibliothek Nokogiri[11] verwendet. Diese ist eine freie Bibliothek, um XML Dokumente zu parsen und via XPath oder CSS3 zu durchsuchen.

Mit dem GPX-Format können GPS-Daten gespeichert werden. Eine Datei mit POIs in Deutschland wurde von Cloudmade[12] bezogen, die die freien Daten von OpenStreetMap[13] Open Source zur Verfügung stellt.

Eingebunden in die Testdatenbank wurden allerdings nicht alle POIs. Enthalten in dem bezogenen Paket sind:

- ~~Automotive~~
- Eating Drinking
- ~~Government and Public Services~~
- Health care
- Leisure

[11] http://nokogiri.org/
[12] http://downloads.cloudmade.com/europe/western_europe/germany#downloads_breadcrumbs
[13] http://www.openstreetmap.org/

- Lodging
- Night Life and Business
- Shopping
- Sports
- ~~Tourism~~

Die in dieser Aufzählung gestrichenen Punkte wurden in der Testdatenbank nicht berücksichtigt. Die POIs zu ”Automotive” beschreiben zum Großteil Bahnübergänge oder Parkplätze, während die ”Government and Public Services” POIs primär Bushaltestellen oder Stromleitungen enthalten, welche für *Events Around Me* keine nützlichen Informationen darstellen. Auch die beim ”Tourism” überwiegend beschriebenen ”Significant tree” werden in unserer Anwendung keine Berücksichtigung finden.

3.6 Wirtschaftliche Möglichkeiten

Der Dienst versucht Anonymität in einem solchen Maße zur Verfügung zu stellen, wie es für andere LBSNS nicht mehr sinnvoll wäre, da diese so ihre Funktionalität nicht anbieten könnten. Um dieses erfolgsversprechende Konzept zu realisieren, muss allerdings die Finanzierung berücksichtigt werden.

Für *Events Around Me* lassen sich in der Hauptsache zwei Finanzierungsmodelle realisieren:

- Werbung
- Datenerhebung

Während die Finanzierung durch Werbung keiner weiteren Erklärung bedarf - es können simple Werbebanner in der App platziert werden oder auffälligere Markierungen an Betreiber vermietet werden - muss die Möglichkeit mittels Datenerhebungen Geld zu verdienen, bei einer Anwendung die gänzlich auf Anonymität baut, erklärt werden: Eingangs wurde für *Events Around Me* das Ziel definiert, die Menge der gespeicherten Daten weitestgehend zu reduzieren.

In der vorliegenden Applikation werden zwar keine Informationen gespeichert, die auf einen bestimmten Nutzer zurückzuführen sind, dennoch werden einige andere Informationen gespeichert - primär in der *History*. In dieser wird nicht nur stündlich protokolliert, wie viele Personen sich an bestimmten Orten aufhalten, sondern auch alle weitergehenden Angaben. So lässt sich, sofern die Verbreitung der Anwendung ausreichend ist, ein umfassendes Bild über die Beliebtheit von beispielsweise Restaurants und Bars herausfiltern. Es ließe sich ebenso ein weitergehenderes Feedback für die entsprechenden Betreiber ermitteln. Die auf diese Weise gesammelten Daten sind in höchstem Grade marktrelevant und können für Studien o.Ä. verwendet werden.

3.7 Mögliche Schwachstellen

Bei einem offiziellen Release der Anwendung werden vorraussichtlich Probleme beim Erreichen der kritischen Masse entstehen. Im Rahmen der Anwendung können die APIs von Facebook und Twitter nicht eingebunden werden, sodass die Anwendung selbst, ohne Hilfsmittel, ein

ausreichend großes Netzwerk von Nutzern aufbauen muss. Um vor allem zu Beginn zusätzliche Daten für Markierungen zu erhalten, ließen sich, ähnlich wie die Seite von *PleaseRobMe*[14] es tut, die öffentlich gemachten Check-ins von FB, Twitter oder Foursquare einlesen um auf deren Basis Markierungen in der Karte von *Events Around Me* anzuzeigen.

Es wurden bereits verschiedene Schwachstellen von *Events Around Me* durch einige Sicherheitsvorkehrungen in der Architektur beseitigt. Um zu bspw. zu verhindern, dass ein Angreifer die Kommunikation abhören kann, wird sowohl eine SSL Verschlüsselung, als auch eine RSA Verschlüsselung verwendet. So wird es für mögliche Angreifer erschwert, durch Belauschen der Kommunikation Rückschlüsse über z.B. das Bewegungsprofil eines Nutzers zu ziehen. Um die in 3.5.1 erwähnte Man-in-the-Middle Attack zu verhindern, wird zu Beginn ein RSA-Schlüsselpaar vom Server generiert, von welchem dem Client der öffentlichen Schlüssel übermittelt wird, mit dem er seine Daten vor dem Versenden verschlüsseln kann.

Da in der derzeitigen Testimplementierung zu Debug-Zwecken noch alle Einträge inkl. ID abgerufen werden können, könnte eine weitere Bedrohung dadurch entstehen, dass ein Angreifer periodisch alle Einträge abfragt und diese Informationen speichert. Wenn er zudem Zugriff auf die IDs eines spezifischen Nutzers erhält, ist er in der Lage, zu einem Nutzer ein Bewegungsprofil zu erstellen. Dieser Bedrohung wird dadurch vorgebeugt, dass in einer für den produktiven Einsatz geplanten Version der Anwendung die Möglichkeit, Einträge abzurufen, nicht angeboten wird. Zudem werden verschiedene Sicherheitsvorkehrungen getroffen, damit die ID eines Nutzers nicht einsehbar ist. Neben der Kommunikationsverschlüsselung ist auch der interne Speicher, in dem die aktuelle ID stets gespeichert bleibt, gegen Zugriffe von außen durch das dem Smartphone unterliegende Unix System gesichert [40]. Lediglich im Falle eines vom Nutzer *gerootetem* Smartphone ist es möglich, in diese Daten Einsicht zu erhalten. Jede App kann bei einem solchen Gerät Root-Rechte erhalten. Dieses Risiko gehen Nutzer allerdings in jedem Fall, in dem sie sie ihr Gerät *rooten*, ein.

In der Testimplementierung ist es darüber hinaus möglich, die komplette History-Tabelle abzufragen. Da es sich hierbei jedoch um Informationen über den Besucherstatus von u.a. Bars und Restaurants handelt, beschreiben diese wertvolle Daten. Ziel wird daher auch hier sein, diese Abrufmöglichkeit vor einem produktiven Einsatz einzuschränken.

Unabhängig von der technischen Implementierung und den sich daraus ergebenden Bedrohungen birgt auch das Gesamtkonzept einige Schwachstellen. So ist es möglich, die App zu missbrauchen, da komplette Anonymität gewährt wird. Beispielsweise lässt sich mit Hilfe von Emulatoren der Android App einen beliebigen Aufenthaltsort vortäuschen. So kann etwa der Besitzer einer Bar einen Eintrag einstellen, um noch mehr Kunden zu sich zu locken, und mit anderen Einträgen benachbarte Bars diskreditieren. Um dies zu erschweren, werden die geschätzte Zahl an Besuchern und das Rating gemittelt. Allerdings bieten diese Mechanismen keinen ausreichenden Schutz gegen diese Art von Missbrauch. Erst wenn genug ”echte” Nutzer an einem Ort *eingecheckt* haben, kann hier keine wirksame Beeinflussung mehr stattfinden. Aus diesem Grund ist es nötig, dass für einen produktiven Einsatz der Anwendung der zuvor bereits thematisierte (dynamische) Schwellwert, der definiert, welche Markierungen angezeigt werden, realisiert wird.

Ein weiterer Punkt, der die Performanz des Systems beeinflusst, ist die Filterung von Markierungen auf dem Smartphone. In allen bisherigen Testfällen war die Anzahl der Markierungen auf der Karte gering und somit ergab sich kein Problem dabei, dass bei jedem Laden der Karte

[14] www.pleaserobme.com

erneut alle Markierungen übertragen werden, und der Filter, der einige Markierungen entfernt, lediglich auf dem Smartphone angewandt wird. Wenn diese Anwendung allerdings deutschlandweit eingesetzt wird, ist durch die Vielzahl der dadurch entstehenden Markierungen eine enorme Performanzsteigerung zu erwarten, wenn vor dem Senden der Markierungen an das Gerät bereits auf dem Server ein Filter angewandt wird.

Die Repräsentativität der Aussagen über einen bestimmten Ort muss differenziert betrachtet werden. Die Gruppe der Anwender besteht in Fall von *Events Around Me* nicht aus einer zufällig gezogenen Stichprobe. Die Hypothese, ob und inwiefern Besitzer eines Smartphones einen Ort systematisch anders bewerten, als Personen die kein Smartphone besitzen, muss dafür zunächst in weiteren Studien überprüft werden.

3.8 Verbesserungsmöglichkeiten

Die Funktionalität der Anwendung ist noch sehr begrenzt. Bisher wird lediglich der Check-in und das Anzeigen der Karte zur Verfügung gestellt.

Ein Feature, welches bei einer solchen App wünschenswert wäre, ist die Navigation zu Markierungen auf einer Karte. Vor allem bei Smartphones, die häufig auch die Funktion eines Navigationsgeräts erfüllen, liegt diese Funktion nahe.

Eine weitere Möglichkeit, die Usability der Anwendung zu verbessern, wird automatische Updates des Aufenthaltsortes geboten. Dazu könnte der Service (nach Aktivierung durch den Nutzer) im Hintergrund alle 20 Minuten den Aufenthaltsort aktualisieren. Probleme ergeben sich allerdings hierbei bei der Performance des Geräts, der Abschottung gegen sogenannte Task Killer, die im Hintergrund laufende Dienste beenden, die notwendig wäre, und bei dem Strom Verbrauch einer solchen zusätlichen Funktionalität.

Zudem könnte ein Veranstaltungsfinder implementiert werden, der auf Basis der derzeitigen Position und eines angegebenen Radius aktuelle und in naher Zukunft stattfindende Veranstaltungen vorschlägt, und ggf. auch die Navigation dorthin ermöglicht. Zu diesem Zweck könnten die Programme von Veranstaltungsverzeichnissen wie z.B. Partyamt[15] geparst und auf diese Weise zu einem Ort auch einen Flyer oder den Link zur Website des Geschäfts angezeigt werden.

Da zudem durchaus viele Nutzer existieren, die ihren Freunden ihren aktuellen Aufenthaltsort mitteilen wollen, und diese Funktion in der Anwendung bisher noch nicht eingebunden ist, ist eine Anbindung an Facebook denkbar. Durch diese würde *Events Around Me* zwar nicht mehr die zweifelsfreie Bezeichnung als *datenschutzfreundlich* verdienen, da wir in diesem Fall die entsprechenden Daten aus der Hand geben und keine Aussagen mehr über die Verarbeitung der Daten treffen können, allerdings wird kein Nutzer zu dieser Verbindung gezwungen, und die durch diese Verknüpfung zur Verfügung stehenden Informationen über den Nutzer (im wesentlichen Daten von Facebook) von Seiten *Events Around Me*'s nicht verwendet werden, ginge man mit einer solchen Funktion lediglich auf diejenigen Nutzer zu, die diese Möglichkeit auch nutzen wollen.

Um darüber hinaus noch mehr Anwender zu erreichen, wäre es notwendig, auch für andere mobile Systeme eine Client Anwendung anzubieten. Wie in Abbildung 1.1 zu sehen, bietet Android zwar die größte Verbreitung, aber dennoch besitzen nur etwas mehr als 30% der deutschen Smartphonebesitzer ein solches Betriebssystem. Nutzer von anderen Geräten bleiben bisher ausgeschlossen.

[15] http://partyamt.de

Die Performanz des Servers ist durch den Einsatz des Webservers WEBrick kritisch zu beurteilen. Dieser ist relativ langsam und um einen produktiven Einsatz zu verwirklichen, müsste eine Portierung zu beispielsweise Thin[16] in Betracht gezogen werden. Sein Vorteil ist lediglich, das er bei allen Ruby-Versionen standardmäßig vorhanden und nutzbar ist, sodass somit eine einfache und schnelle Realisierung ermöglicht wird.

[16] http://code.macournoyer.com/thin/

4 Schlussfolgerung

Die im Rahmen dieser Arbeit entwickelte Applikation zeigt, dass Anwendungen auf eine solche Weise entwickelt werden können, dass der Endnutzer weitestgehend die Kontrolle über seine Daten behält. Allerdings bleibt zumindest zweifelhaft, ob dies wirklich im Interesse der Diensteanbieter ist, da sie dadurch ihre (einzige bzw. wichtigste) Einnahmequelle verlieren. Schließlich bedürfen die meisten Dienste einer dauerhaften Finanzierung. Sowohl Entwickler als auch Endnutzer werden sich daher in Zukunft mit der Frage auseinandersetzen müssen, wie sie ihre Anwendungen finanzieren können bzw. wollen. In diesem Rahmen können Werbeeinblendungen, Daten oder alternative Finanzierungsmodelle genutzt werden. Die Existenz einer ausreichend großen Auswahl an Alternativen ist wünschenswert, damit jeder Nutzer diese Entscheidung für sich selbst treffen kann. Ein Standard für die Austauschbarkeit der Anwendungen wäre zu diesem Zwecke erstrebenswert, sodass dieselbe Funktionalität mit unterschiedlichen Programmen genutzt werden kann, ohne besondere Portierungsmaßnahmen unternehmen zu müssen.

Glossary

Android Android ist ein Betriebssystem für mobile Geräte wie Smartphones, Mobiltelefone, Netbooks und Tablets, die von der Open Handset Alliance (Hauptmitglied: Google) entwickelt wird. Basis ist der Linux-Kernel. Es handelt sich hierbei um freie Software, die quelloffen entwickelt wird [41].

Base64 Base64 beschreibt ein Verfahren zur Kodierung von 8-Bit-Binärdaten (z. B. ausführbare Programme, ZIP-Dateien) in eine Zeichenfolge, die nur aus lesbaren Codepage-unabhängigen ASCII-Zeichen besteht [42].

CA Certificate Authority.

EU Europäischen Union.

FB Facebook.

FBP Facebook Places.

GPS Global Positioning System.

GPX GPS Exchange Format.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

LBS Location Based Service.

LBSNS Location Based Social Networking Site.

POI Points of Interests.

REST Representational State Transfer.

RSA RSA ist ein asymmetrisches kryptographisches Verfahren, das sowohl zur Verschlüsselung als auch zur digitalen Signatur verwendet werden kann. Es verwendet ein Schlüsselpaar, bestehend aus einem privaten Schlüssel, der zum Entschlüsseln oder Signieren von Daten verwendet wird, und einem öffentlichen Schlüssel, mit dem man verschlüsselt oder Signaturen prüft. Der private Schlüssel wird geheim gehalten und kann nur mit extrem hohem Aufwand aus dem öffentlichen Schlüssel berechnet werden [43] [44].

Ruby Ruby ist nach der Ruby-Community [45] eine ”dynamische, freie Programmiersprache, die sich einfach anwenden und produktiv einsetzen lässt. Sie hat eine elegante Syntax, die man leicht lesen und schreiben kann.".

SSL Secure Sockets Layer.

UTF-8 UTF-8 ist die am weitesten verbreitete Kodierung für Unicode-Zeichen.

UUID Universal Unique-ID.

WS Webservice.

XML Extensible Markup Language.

Literaturverzeichnis

[1] Presseinfo Besitz von Smartphones. Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM). April 2012.

[2] BVerfG. Volkszählungsurteil. Dezember 1983.

[3] Organization for Economic Cooperation and Development (OECD). Guidelines Governing the Protection of Privacy and Transborder Flow of Personal Data, 1980.

[4] Bundesdatenschutzgesetz § 1 Abs. 5 Kollisionsregeln. http://www.gesetze-im-internet.de/bdsg_1990/

[5] Bundesdatenschutzgesetz § 4 Abs. 1. http://www.gesetze-im-internet.de/bdsg_1990/

[6] Facebook Inc. Nutzungsbedingungen Ziff. 17 Abs. 1. https://www.facebook.com/legal/terms in der Version vom 8. Juni 2012.

[7] Kölmel, B.: Location Based Services. In: Pousttchi, K.; Turowski, K. (Hrsg.): Mobile Commerce: Anwendungen und Perspektiven, Proceedings zum 3. Workshop Mobile Commerce, Bonn, 2003, S. 88-101.

[8] Department of Defense (USA). GPS Standard Positioning Service Performance Standard. Oktober 2008.

[9] E. Kaplan, C. Hegarty. Understanding GPS: Principles and Applications. Norwood, MA: Artech House, 1996.

[10] Google Android Developer Guide: Obtaining User Location. http://developer.android.com/guide/topics/location/obtaining-user-location.html

[11] Marco Ghiglieri, Hervais Simo, Michael Waidner: Technical Aspects of Online Privacy, 2012.

[12] Go-Smart-Studie 2012. TNS Infratest, Trendbüro, Google, Otto Group. 2012.

[13] The Guardian. IPhone tracking promts privacy fears. 20. April 2011. http://www.guardian.co.uk/technology/2011/apr/20/iphone-tracking-prompts-privacy-fears, zuletzt aufgerufen am 28.04.2012.

[14] Sophos - NakedSecurity. Apple iOS update quases location tracking bug. 05. Mai 2011. http://nakedsecurity.sophos.com/2011/05/05/apple-ios-update-quashes-location-tracking-bug/, zuletzt aufgerufen am 03.05.2012.

[15] Datenschutz im Internet. Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM). 2011.

[16] Andreas Poller: Privatsphärenschutz in Soziale-Netzwerke-Plattformen; Fraunhofer-Institut für Sichere Informationstechnologie; Darmstadt 2008. http://sit.sit.fraunhofer.de/studies/de/studie-socnet-de.pdf.

[17] Amendment No. 4 to Form S-1 REGISTRATION STATEMENT. UNITED STATES SECURITIES AND EXCHANGE COMMISSION. April 2012.

[18] NASDAQ Community. Facebook prices at 38 USD, high end of revised range; third-largest US IPO ever. http://community.nasdaq.com/News/2012-05/facebook-prices-at-38-high-end-of-revised-range-thirdlargest-us-ipo-ever.aspx?storyid=142183, zuletzt aufgerufen am 04.07.2012.

[19] Rainer Böhme, Andreas Pfitzmann: Digital Rights Management zum Schutz personenbezogener Daten?; Datenschutz und Datensicherheit – DuD 32/5 (2008) 342-347.

[20] Constanze Kurz, Frank Rieger: Die Datenfresser: Wie Internetfirmen und Staat sich unsere persönlichen Daten einverleiben und wie wir die Kontrolle darüber zurückerlangen; S. Fischer Verlag, Frankfurt 2011.

[21] Heise Online. Facebook muss neue Nutzungsbedingungen überprüfen. 23. März 2012. http://www.heise.de/newsticker/meldung/Facebook-muss-neue-Nutzungsbedingungen-ueberpruefen-1479259.html, zuletzt aufgerufen am 28.04.2012.

[22] golem.de. Facebook legt überarbeitete Nutzungsbedingungen vor. 21. April 2012. http://www.golem.de/news/datenschutz-facebook-legt-ueberarbeitete-nutzungsbedingungen-vor-1204-91316.html, zuletzt aufgerufen am 29.04.2012.

[23] Facebook Site Governance. https://www.facebook.com/fbsitegovernance, zuletzt aufgerufen am 22.05.2012.

[24] golem.de. Nutzer lassen Facebooks Datenschutzrichtlinie durchfallen. 18.05.2012. http://www.golem.de/news/studentengruppe-nutzer-lassen-facebooks-datenschutzrichtlinie-durchfallen-1205-91882.html, zuletzt aufgerufen am 22.05.2012.

[25] golem.de. Facebook setzt neue Regelungen in Kraft. 11.06.2012. http://www.golem.de/news/abstimmung-gescheitert-facebook-setzt-neue-regelungen-in-kraft-1206-92448.html, zuletzt aufgerufen am 22.05.2012.

[26] Foursquare About. https://de.foursquare.com/about/, zuletzt aufgerufen am 30.04.2012.

[27] Win Future. Google Latitude führt Check-In-Punktesystem ein. 20. Februar 2012. http://winfuture.de/news,68213.html, zuletzt aufgerufen am 30.04.2012.

[28] Google Support. Status levels and offers . http://support.google.com/gmm/bin/answer.py?hl=de
GB&ctx=topic&answer=1650348, zuletzt aufgerufen am 30.04.2012.

[29] Google Support. Check-ins Leaderboard. http://support.google.com/gmm/bin/answer.py?hl=en-GB&answer=1727362&topic=1650398&ctx=topic, zuletzt aufgerufen am 18.05.2012

[30] Electronic Frontier Foundation. EXCLUSIVE: Google Takes a Stand for Location Privacy, Along with Loopt. März 2009. https://www.eff.org/deeplinks/2009/03/exclusive-google-takes-stand-location-privacy-alon, zuletzt aufgerufen am 30.04.2012

[31] Loopt About. http://www.loopt.com/about/, zuletzt aufgerufen am 02.05.2012

[32] Marketwatch. Green Dot to acquire Loopt. 09.03.2012. http://www.marketwatch.com/story/green-dot-to-acquire-loopt-2012-03-09, zuletzt aufgerufen am 02.05.2012

[33] Loopt Blog. Thank you. März 2012. http://www.loopt.com/blog/2012/03/thank-you/, zuletzt aufgerufen am 02.05.2012

[34] Loopt About. Loopt Qs. http://www.loopt.com/about/qs/, zuletzt aufgerufen am 18.05.2012

[35] PC Games Hardware. Sicherheitsexperte entdeckt gravierende Lücken in Facebook App. 08. April 2012. http://www.pcgameshardware.de/aid,876979/Sicherheitsexperte-entdeckt-gravierende-Luecken-in-Facebook-App/Sicherheit/News/, zuletzt aufgerufen am 02.05.2012

[36] The Registert. Android phone privacy shocker. 24.02.2011. http://www.theregister.co.uk/2011/02/24/android_phone_privacy_shocker/, zuletzt aufgerufen am 02.05.2012

[37] Lorrie Cranor, Simson Garfinkel: Security and Usability; O'Reilly Media, Inc., 2005.

[38] Lorrie Cranor, Simson Garfinkel: Guest Editors' Introduction: Secure or Usable?; 2004 IEEE.

[39] Peter Burkholder. SSL Man-in-the-Middle Attacks. Februar 2002.

[40] Google Android Developer Guide: Application Fundamentals. http://developer.android.com/guide/topics/fundamentals.html

[41] Licenses. Android Open Source Project. http://source.android.com/source/licenses.html

[42] The Base16, Base32, and Base64 Data Encodings. The Internet Society. Oktober 2006.

[43] RSA-Kryptosystem. Wikipedia. http://de.wikipedia.org/wiki/RSA-Kryptosystem, zueltzt aufgerufen am 14.06.2012.

[44] R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.1977.

[45] Ruby. http://www.ruby-lang.org/de/, zuletzt aufgerufen am 22.05.2012