

Eberhard-Karls-Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Arbeitsbereich für Theoretische Informatik / Formale Sprachen

Abhörsichere PIN-Eingabe in Verbindung mit einem Head- Mounted-Display

Bachelorarbeit

Vorgelegt von: Jakob Rübler
Matrikelnummer: 3394344

02. September 2013

Prüfer: Apl. Prof. Dr. Klaus Reinhardt

Betreuer: Dr. Bernd Borchert

Kurzzusammenfassung

Die vorliegende Arbeit beschäftigt sich mit dem Potential von Augmented-Reality-Techniken und möglicher Probleme bei der Implementierung und Anwendung. Als Beispielapplikation fungiert hierfür die eKaay-Applikation zum sicheren Login für Online Accounts. Es wird die Schnittstelle zwischen der eKaay-Applikation und einem Display, in diesem Fall einer Smart Watch, erkundet, um herauszufinden, inwieweit hier Gelegenheiten für einen weiteren Einsatz gegeben sind.

Des Weiteren soll ein Ausblick auf zukünftige Entwicklungen gegeben werden und auf die anstehenden Probleme, aber auch Möglichkeiten, hingewiesen werden.

Inhaltsverzeichnis

1 Einleitung	3
1.1 Motivation	4
1.2 Aufgabenstellung	4
1.3 Durchführung	5
2 Grundlagen	6
2.1 Augmented Reality	6
2.2 Das eKaay-Login-Verfahren	7
2.3 Smart Watches	9
2.4 Das Sony SDK zur Entwicklung von Applikationen	10
3 Implementierung einer Smart Extension	12
3.1 Architektur der Implementierung	12
3.2 Manifest einer Applikation	20
3.3 Die Klasse WatchExtensionReceiver	21
3.4 Die Klasse WatchExtensionService	21
3.5 Die Klasse WatchRegistrationInformation	22
3.6 Die Klasse ControlSmartWatch	22
4 Diskussion	23

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere weiterhin, dass ich diese Arbeit noch keinem anderen Prüfungsgremium vorgelegt habe.

Tübingen, den 2. September 2013

Jakob Rüßler

Danksagung

Ich möchte mich bei Prof. Klaus Reinhardt, meinem Prüfer, und Dr. Bernd Borchert, meinem Betreuer, recht herzlich für die Möglichkeit, diese Arbeit zu verfassen, bedanken. Weiterhin möchte ich meinen Brüdern Christoph und Martin und meinen Freunden Thomas und Jörn danken für ihre fortwährende Unterstützung.

1 Einleitung

Mit den bisherigen und angekündigten Fortschritten in der Augmented-Reality-Technik („erweiterte Realität“, AR), wie z. B. Google Glass, stellt sich die Frage, inwieweit diese Technologien in der breiten Bevölkerung angenommen und genutzt werden können. AR-Geräte bestehen in den meisten Fällen aus einer Kamera zur Wahrnehmung der Umgebung und einem Display, welches dem Nutzer Informationen darstellen kann. Eine Form davon stellen Head-Up-Displays (HUDs) dar. Dies sind Anzeigegeräte, die es ihren Anwendern ermöglichen, Informationen wahrzunehmen, ohne den Kopf bewegen zu müssen. Zuerst wurden sie im militärischen Bereich in Kampfflugzeugen eingesetzt, um Piloten sicher und schnell wichtige Informationen anbieten zu können, ohne dass diese dadurch in ihrer Flugfähigkeit eingeschränkt werden (Sung 2011). Dazu wurde die Frontscheibe als Display genutzt. Solche HUDs sind bereits vielfältig im Einsatz, sei es im Bereich Navigation, Militär oder Geologie, und nun versuchen immer mehr Unternehmen, die Möglichkeiten derartiger Technologien zu erforschen und erste Produkte zur Marktreife zu bringen. Hierfür werde ich im Laufe der Arbeit weitere Beispiele anführen. Die Möglichkeiten zum Einsatz von HUDs erscheinen sehr breit gefächert, vom Einsatz in Museen, in Form von eingeblendeten Erläuterungen zu bestimmten Ausstellungsstücken, bis hin zur Verwendung in Krankenhäusern, in denen Ärzten die Krankenakten der Patienten oder bei Operationen Messdaten oder Operationshilfen angezeigt werden.

Diese Arbeit beschäftigt sich im Besonderen mit der eKaay-App, einer Applikation zum sicheren, mobilen Login per Smartphone. Sie untersucht, inwieweit die Sicherheit des eKaay-Login-Verfahrens durch die Verwendung von AR-Technik und somit einer Anzeige auf einem externen Display erhöht werden kann. Hierfür wurde die graphische Ausgabe des Login-Verfahrens per Bluetooth auf ein externes Display verlegt.

1.1 Motivation

Momentan erscheint die Augmented-Reality-Technik in Form von „wearable computers“, also am Körper tragbaren Computern, als folgerichtige und sinnvolle Weiterentwicklung der Interaktion des Menschen mit seiner Umgebung dargestellt. Es ist schließlich nicht nötig, sein Smartphone aus der Tasche zu holen, wenn die wichtigen Informationen auch direkt ins Sichtfeld eingeblendet werden können. Daraus ergibt sich die Idee, dass auch der Login-Vorgang zu Online-Accounts verbessert werden kann. In Kombination mit der eKaay-App würde eine Darstellung des Login-Bildschirms, welcher nicht mehr auf dem Smartphone potentiell für Außenstehende, sondern auf einem HUD nur noch für den jeweiligen Nutzer alleine sichtbar ist, für eine erhöhte Sicherheit sorgen. Bei einer AR-Brille könnte man möglicherweise sogar auf das Smartphone verzichten und über die eingebaute Kamera den für das eKaay-Verfahren benötigten QR-Code („quick response“, soviel wie „schnelle Antwort“, ein zweidimensionaler Code, der die kodierten Daten binär darstellt) scannen und die Kommunikation und Berechnung über den brilleninternen Prozessor laufen lassen. Dies wäre ein weiterer Schritt in die Richtung, Passwörter sicherer einzugeben. Der sicherste Ort für ein Passwort ist das eigene Gehirn, und je kürzer die Strecke ist, die ein Passwort außerhalb des Hirns „reisen“ muss, desto besser.

1.2 Aufgabenstellung

Das an der Universität Tübingen entwickelte abhörsichere PIN-Verfahren eKaay-PIN könnte zusammen mit der Augmented-Reality-Technik dazu verwendet werden, den abhörsicheren Login an beliebigen PCs noch bequemer zu machen. In dieser Arbeit soll die Schnittstelle zwischen Smartphone, auf dem weiterhin Login und Zugangsberechtigung verwaltet werden, und einem externen Display erkundet werden, um eine weitere Modularisierung des Login-Verfahrens zu ermöglichen. Dies hat eine größere Sicherheit zur Folge, da Verschlüsselung und Anzeige der Permutation auf diese Weise noch einmal getrennt werden können.

1.3 Durchführung

Zuerst stelle ich die Grundlagen der genannten und verwendeten Technologien vor. Anschließend gehe ich zu einer detaillierten Erläuterung der Implementierung der eKaay-Applikation für ein externes Display über und im Abschluss werde ich die potentiellen Anwendungen, Akzeptanz und Sicherheitsrisiken in Bezug auf Augmented Reality darstellen.

2 Grundlagen

2.1 Augmented Reality

Augmented Reality (auch „erweiterte Realität“, AR) ist die computergestützte Ergänzung der wahrgenommenen Realität. Im Allgemeinen bezieht sich dieser Ausdruck auf jegliche Sinneswahrnehmungen, meist wird jedoch besonders der visuelle Aspekt beachtet. Als Paradebeispiel wäre hier die Einblendung von zusätzlichen Informationen bei Sportübertragungen im Fernsehen, z.B. Football wie in Abbildung 1 erkennbar, zu nennen: Durch eine zusätzliche grafische Information wird angezeigt, wie weit die Angreifer den Ball mindestens bringen müssen, um einen Raumgewinn und weitere Angriffsmöglichkeiten zu erlangen.



Abbildung 1: Augmented Reality bei Fernsehübertragungen. Einblendung einer First-Down-Linie (in Gelb) bei einem amerikanischen Football-Spiel (Sung 2011).

Im Gegensatz zur Virtual Reality, welche den Nutzer in eine (möglicherweise völlig andere) Welt eintauchen lässt, liegt der Fokus der AR in der Erweiterung der aktuellen Realität durch zusätzliche Informationen. Daraus ergeben sich einige Schwierigkeiten für die mobile Anwendung: Es muss registriert werden, was der Nutzer wahrnimmt und wo der Nutzer sich aufhält. Dies kann beispielsweise per Kamera, welche einerseits die Umgebung im Ganzen wahrnimmt und den genauen Standort des Nutzers feststellt, sowie einem GPS-Sender erfolgen. Zusätzlich

muss versucht werden, zu erkennen, welche Objekte (z. B. Gebäude, Sehenswürdigkeiten oder Ähnliches) der Nutzer gerade genauer betrachtet, um nur die jeweils passenden Informationen einzublenden und das Display nicht zu überladen. Es eröffnen sich unzählige Szenarien, in denen diese Technik genutzt werden kann und teilweise auch schon genutzt wird:

- Medizintechnik (Medgadget 2012)
- Navigation (Chip.de 2010)
- Fernwartung (Arvika 2001)
- Bedienungsanleitungen (Arvika 2001)
- Archäologie/Geographie (Dähne und Karigiannis 2002)
- Ausstellungen, Museen, Schule (Miyashita u. a. 2002)
- Tourismus (Wikitude 2009)

2.2 Das eKaay-Login-Verfahren

eKaay ist ein seit 2009 von Mitarbeitern am Wilhelm-Schickard-Institut der Universität Tübingen entwickeltes Verfahren zur Benutzerauthentifizierung im Internet (Borchert und Reinhardt 2007; Borchert 2009). Es ermöglicht dem Benutzer das Einloggen in verschiedene Online-Dienste durch das Scannen eines QR-Codes auf einer Website mit Hilfe der Kamera eines Smartphones. In dieser Arbeit bezieht sich eKaay auf die trojanersichere 2-Kanal-Version mit PIN-Eingabe. Wenn man dieses zum Login auf einem Internetportal, beispielsweise dem Webmailer der Universität Tübingen, nutzen möchte, muss man dort in den Accounteinstellungen die Option für eKaay-Login aktivieren. Auf dem eKaay-Server wird nun ein Schlüssel generiert, der einmal auf dem Server gespeichert bleibt und einmal auf dem Smartphone des Nutzers. Zur Verifizierung ist das eigene Passwort erforderlich und vom eKaay-Server wird ein QR-Code generiert. Darin enthalten sind die Informationen für die Zahlenpermutation. Diese Permutation wird einmal auf dem Server gespeichert und einmal mit dem erstellten Schlüssel verschlüsselt und dem Nutzer im

Browser angezeigt. Dort scannt der Nutzer den Code mit einem Smartphone und gelangt zu der PIN-Vergabe. Auf dem Smartphone wird wie in Abbildung 2 erkennbar ein Zahlenfeld mit den Ziffern von Null bis Neun in scheinbar zufälliger Anordnung angezeigt. Auf dem Desktopmonitor hingegen ist ein identisch aufgebautes Zahlenfeld aber nur mit grauen Felder ausgefüllt. Nun denkt sich der Nutzer eine PIN-Nummer, zwischen vier und zwölf Ziffern, davon mindestens zwei unterschiedlich, aus. Er schaut, in welchem Feld auf dem Handy die erste Ziffer der PIN-Nummer steht und klickt das entsprechende Feld auf dem Monitor an. Dies wiederholt er solange, bis die PIN-Nummer vollständig eingetragen ist. Mit einem Klick auf „Senden“ und einem weiteren Eingeben der PIN-Nummer zur Bestätigung, kann er sich nun einloggen. Somit ist für den Nutzer eine feste PIN-Nummer auf dem Server gespeichert. Für zukünftige Logins ist kein erneutes Vergeben einer PIN-Nummer nötig, es ist nur das Scannen des QR-Codes und anschließendes Eingeben der PIN-Nummer nach dem oben beschriebenen Schema erforderlich. Dem Server werden die Klickpositionen übermittelt, die er mit der PIN des Nutzers und dem Zahlenfeld vergleicht und bei Korrektheit der Daten zu dem Account gewährt.



Abbildung 2: Das eKaay Login-Verfahren mit PIN-Eingabe. Links die Anzeige des Monitors mit QR-Code und ausgegrautem Tastenfeld, rechts die Permutation auf dem Smartphone (SonyMobile 2013b).

Herkömmliche Login-Verfahren mit einfachem Passwortschutz sind anfällig für Schadsoftware wie Keylogger oder Trojaner. Diese können sich vom Nutzer unbemerkt auf dem Rechner installieren und fortan alle Tasteneingaben oder Aktionen auf dem Rechner mitlesen, abspeichern und somit Passwörter und Logindaten entwenden. Das eKaay-PIN-Verfahren verhindert dies, da es gänzlich irrelevant ist,

ob sich auf dem Desktoprechner, dem Smartphone oder sogar beidem Schadsoftware befinden. Auf dem Rechner kann nur die Position der Mausklicks registriert werden. Diese ist aber ohne die Bedeutung der Position nutzlos für einen Angreifer, und auf dem Smartphone kann lediglich die Permutation angeschaut werden, nicht jedoch die Mausklicks. Falls es nun einem Angreifer gelingt, beide Informationen, also Position der Mausklicks auf dem Rechner und dazugehörige Permutation vom Smartphone, zu erlangen, so ist es ihm theoretisch möglich, die PIN-Nummer des Nutzers herauszufinden. Diese allein gewährt ihm aber immer noch keinen Zugang zu dem Account, sondern nur in Kombination mit dem Schlüssel des Nutzers, der jedoch nur auf dem Server und dem Smartphone gespeichert ist. Durch die räumliche Trennung der einzelnen Schritte des Login-Verfahrens wird also für eine erhöhte Sicherheit gesorgt.

2.3 Smart Watches

Laut einer Studie von Nielsen nutzen 94% der Bevölkerung Deutschlands ein Mobiltelefon und davon sind bereits fast zwei Drittel (62%) Smartphones (Nielsen 2013). E-Mails abrufen, Fotos schießen, Videos aufnehmen, im Internet surfen, Apps nutzen – eine Vielzahl von Funktionen erfüllen Smartphones schon heute, mit steigender Tendenz. Doch gibt es immer wieder Situationen, in denen das Telefon nicht griffbereit ist oder es einfach unhöflich wäre, das Smartphone in die Hand zu nehmen. Neue, Smart Watches genannte Armbanduhren verbinden sich in der Regel per Bluetooth mit einem Smartphone und zeigen nicht nur die Uhrzeit, sondern zudem eine ganze Reihe einstellbarer Informationen wie beispielsweise ausgewählte RSS-Feeds, Tweets oder E-Mails, also Informationen vom Telefon an. Ein Blick auf die Uhr genügt, um die Informationen einzusehen und gegebenenfalls darauf zu reagieren. Das Smartphone bleibt dabei in der Hosentasche. „Vielleicht rennt man gerade zum Bus oder schleppt unzählige Taschen. Es gibt viele Situationen, in denen es einfacher ist, kurz aufs Handgelenk zu schauen. Und außerdem ist es eine diskrete Form, auf sein Telefon zu schauen“, erklärt Eric Migicovsky, Lead-Designer der Smart Watch „Pebble“ (Migicovsky 2013). In dieser Arbeit wurde eine Sony „SmartWatch“, sichtbar in Abbildung 3, mit OLED-Multitouch-Anzeige und einer

Auflösung von 128 x 128 Pixel, die bis zu 65.536 Farben anzeigen kann, verwendet (SonyMobile 2013a). Die SmartWatch misst 36 x 36 Millimeter und wiegt ohne Armband 15,5 Gramm. Per Bluetooth 3.0 kann sie mit Smartphones, auf denen Googles Betriebssystem Android läuft, verbunden werden.



Abbildung 3: Die Sony „SmartWatch“. Auf ihr lassen sich verschiedene Informationen des Smartphones anzeigen und man kann Eingaben tätigen, ohne das Handy aus der Tasche nehmen zu müssen (SonyMobile 2013b).

2.4 Das Sony SDK zur Entwicklung von Applikationen

Das Sony Add-On SDK (Software Development Kit, (SonyMobile 2013b)) ist eine Erweiterung für das bisherige Android SDK, welches eine Sammlung unterschiedlicher APIs (Application Programming Interfaces) darstellt, die bisher nur einzeln über unterschiedliche SDKs erreichbar waren. Enthalten sind die APIs *Control*, *Widget*, *Sensor*, *Registration&Capabilities* und *Notification*. Hiermit lassen sich Applikationen für Sonys Peripheriegeräte, sogenannte „Smart Accessories“ wie die SmartWatch oder andere, entwickeln. Diese Applikationen lassen sich in unterschiedliche Kategorien einteilen, je nachdem, welche Funktionen sie haben und welche API sie nutzen. *Notification*-Apps sind dafür gedacht, *Events* auf dem Smartpho-

ne wie z.B. eine empfangene SMS oder einen eingehenden Anruf über die Smart-Watch an den Nutzer weiterzuleiten. *Widgets* ermöglichen es einer App, auf dem Handy gewisse Daten auf der Uhr anzuzeigen, z. B. einen Musikspieler, der auf der Uhr Liedtitel und -länge einblendet. Über die *Sensor-API* erhält man Zugriff auf den Beschleunigungssensor der Uhr. Die *Control-API* ermöglicht den stärksten Zugriff auf die SmartWatch, sie gibt einer Applikation vollständige Kontrolle über das Display, die LEDs und Tasteneingaben. Dies kann zum Beispiel dazu genutzt werden, mit dem Musikspieler zu interagieren und Lautstärke oder Titelauswahl zu steuern. Die *Registration&Capabilities-API* muss von jeder Extension umgesetzt werden, worauf ich im folgenden Kapitel genauer eingehe.

3 Implementierung einer Smart Extension

3.1 Architektur der Implementierung

Da die Rechenkapazitäten von Smartphones heutzutage immer größer werden, versucht man sich diese Eigenschaft bei der Entwicklung von, bei Sony „Smart Accessories“ getauften, zusätzlichen Peripheriegeräten für ein Smartphone, etwa einer Smart Watch, zu Nutze zu machen. Die Idee ist, die Hardware außerhalb des Smartphones so simpel wie möglich zu halten und den größten Teil der Logik auf dem Smartphone selbst zu verwalten. Daraus folgend ist der überwiegende Teil der Funktionalität in einer Applikation auf dem Smartphone, der „Accessory Host Application“ (AHA), realisiert. Es existiert eine AHA pro „Smart Accessory“ (SA), die Kommunikation geschieht über Bluetooth. Die AHA kontrolliert, was von dem SA angezeigt wird.

Die AHA erlaubt es, darüber zu entscheiden, was gesendet und angezeigt wird sowie was bei der Interaktion mit der SmartWatch geschieht. Der angezeigte Inhalt ist typischerweise kein Teil der AHA, sondern gehört zu eigenen Applikationen, den „Smart Extension Apps“ (SEA). Diese SEAs kommunizieren mit der AHA über die Smart Extension APIs und sind somit nicht spezifisch an nur ein SA gebunden, sondern im Allgemeinen kompatibel mit weiteren (unterschiedlichen) SAs. Abbildung 4 bietet einen Überblick der Kommunikationswege von AHA, SEA und SA.

Eine AHA fungiert als Schnittstelle für die „Smart Extension APIs“ und bietet die Möglichkeit,

- mit der Smartwatch zu kommunizieren,
- ein User Interface zu generieren,
- Signale der SEA zu verwalten,
- über das Smartphone Einstellungen für die Smartwatch zu treffen (Aktivieren/Deaktivieren von SEA, Vibration oder Ähnliches)

Ein Teil der APIs ist Intent-basiert (*Control*, *Widget* und *Sensor*). Intents sind abstrakte Beschreibungen von Prozeduren, die ausgeführt werden sollen. Es kann eine *Activity* gestartet, ein *BroadcastReceiver* angesprochen werden oder es können

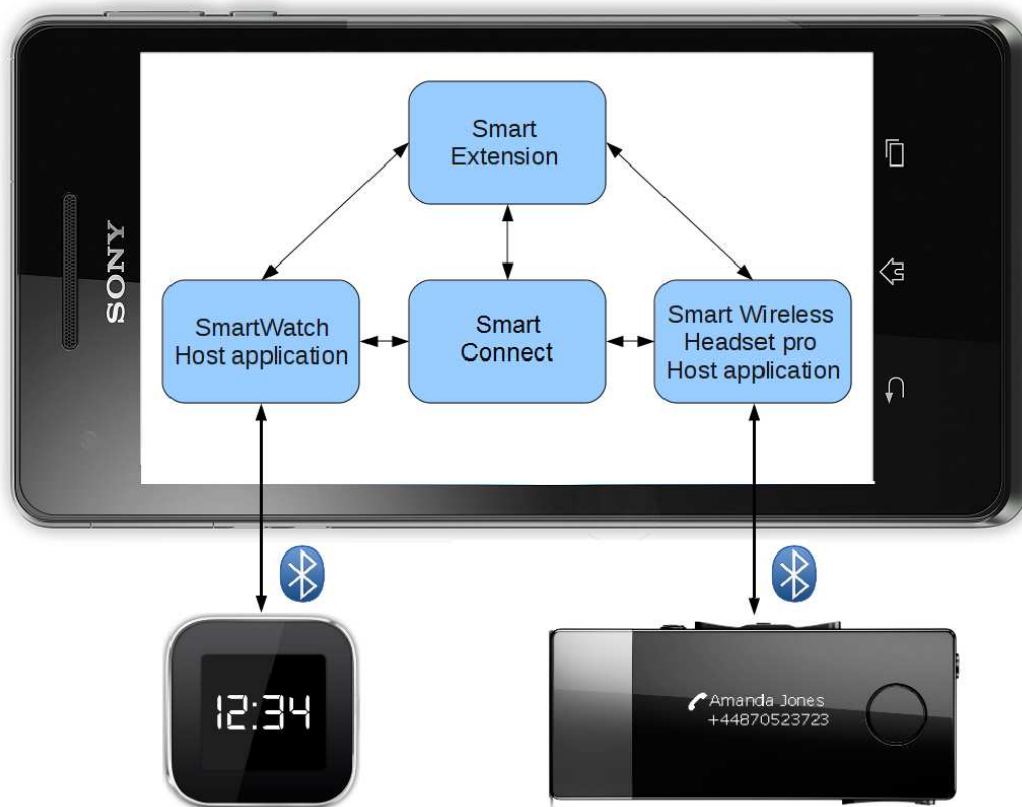


Abbildung 4: Ein Überblick der Architektur der Smart Extensions. SEAs kommunizieren über jeweils eine AHA mit einem SA. Sowohl SEA als auch AHA tragen ihre Daten in Smart Connect ein (SonyMobile 2013b).

weitere *Services* gestartet werden. Die anderen APIs basieren auf *ContentProvi-
 dern (Registration&Capabilities und Notification)*. Diese sind Grundbaustein vieler
 Android-Applikationen und ermöglichen eine einfache Weiterleitung von Daten
 zwischen unterschiedlichen Applikationen. Eine SEA muss sich registrieren, bevor
 es Kontakt zu der AHA aufnehmen kann. Das bedeutet, jede SEA muss mindestens
 die *Registration&Capabilities* API zusätzlich zu der eigentlichen API implementie-
 ren. Ein Überblick ist in Abbildung 5 zu finden. Bevor sich die SEA registrieren
 kann, benötigt sie bestimmte Informationen über die AHA. Darin enthalten müs-
 sen die Fähigkeiten der AHA, wie beispielsweise die Größe des Displays, sein. Aus
 diesem Grund registrieren sich also auch die AHA mit ihren Eigenschaften bei

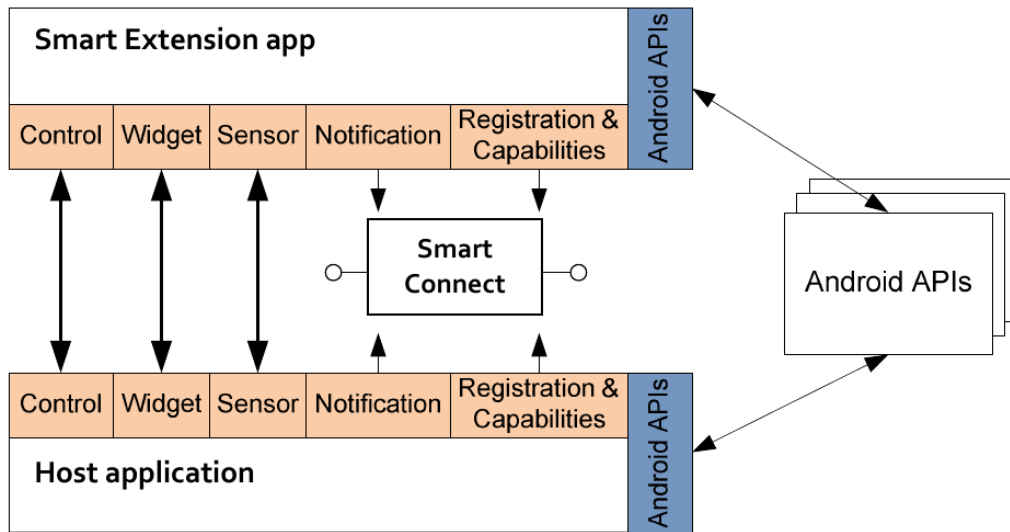


Abbildung 5: Übersicht über die Kommunikation zwischen SEA und AHA. Für komplexere Funktionen können weitere Android APIs miteingebunden werden (SonyMobile 2013b).

Smart Connect, der Verwaltungsapplikation von Sony für ihre SA. Dies geschieht beim Start der AHA durch den ihrer Eigenschaften in eine Datenbank in Smart Connect. Diese können dann von weiteren Applikationen abgefragt werden, um zu überprüfen, ob eine Kompatibilität besteht. Ein Übersicht der Felder, die in der Datenbank vorhanden sind bietet Abbildung 6.

Eine Applikation kann sich erst dann als Extension registrieren, wenn mindestens eine AHA auf dem Smartphone installiert ist. Dies ist notwendig, um zu verhindern, dass die Extension Daten in die Smart Connect-Datenbank schreibt, ohne dass eine AHA vorhanden ist und somit auch kein SA. Dies könnte zu Fehlern in der Datenhaltung aufgrund von fehlerhaften Einträgen führen. Tabelle 1 bietet einen Auszug aus der Datenbank.

Eine Registrierung als SEA bedeutet, dass in der Datenbank die Felder *column*, *data inserted by* und *data readable by* ausgefüllt sind. Dies geschieht während der Installation der Applikation durch einen Aufruf des *registration interface* von Smart Connect, oder wenn ein „EXTENSION_REGISTER_REQUEST_INTENT“ empfangen wird. Dieser Intent wird sowohl beim Start von Smart Connect als auch, wenn

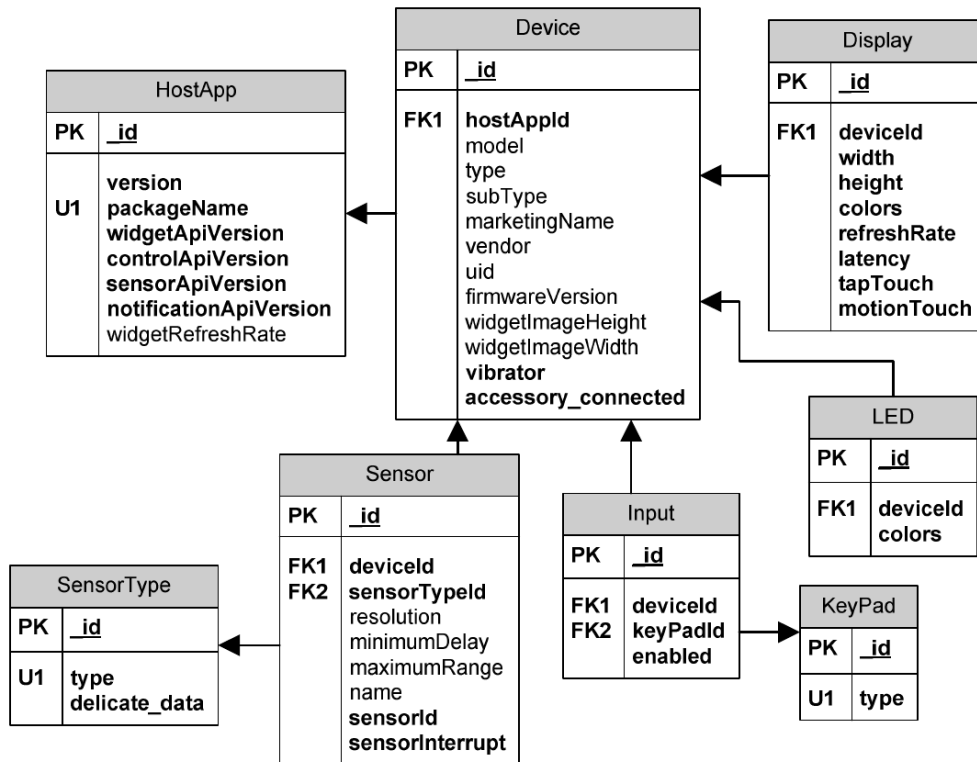


Abbildung 6: Die capabilities-Datenbank in Smart Connect. Hier tragen SEA und AHA ihre Eigenschaften ein, um eine korrekte Kommunikation untereinander zu gewährleisten (SonyMobile 2013b).

Smart Connect erkennt, dass eine neue Applikation installiert wurde oder eine neue AHA ihre Eigenschaften in die Datenbank eingetragen hat, ausgesendet. Der Registrierungsprozess wird in Abbildung 7 verdeutlicht.

Die Control API ist die mächtigste vorhandene API und ermöglicht es, dass die SEA vollständigen Zugriff und Kontrolle über das Display der Uhr hat. Da immer nur eine SEA gleichzeitig aktiv sein kann, muss die Lebensdauer der SEAs von AHAs kontrolliert werden. Eine SEA kann nicht beliebig starten, es muss sichergestellt sein, dass keine weitere Applikation bereits aktiv ist. Wenn die AHA bereit ist die SEA zu starten, so sendet sie einen „CONTROL_START_INTENT“ aus. Ab diesem Zeitpunkt kontrolliert die SEA das SA vollständig, also das Display, LEDs, Vibration sowie Tasteneingaben. Die AHA dient nur als Weiterleitung der Kom-

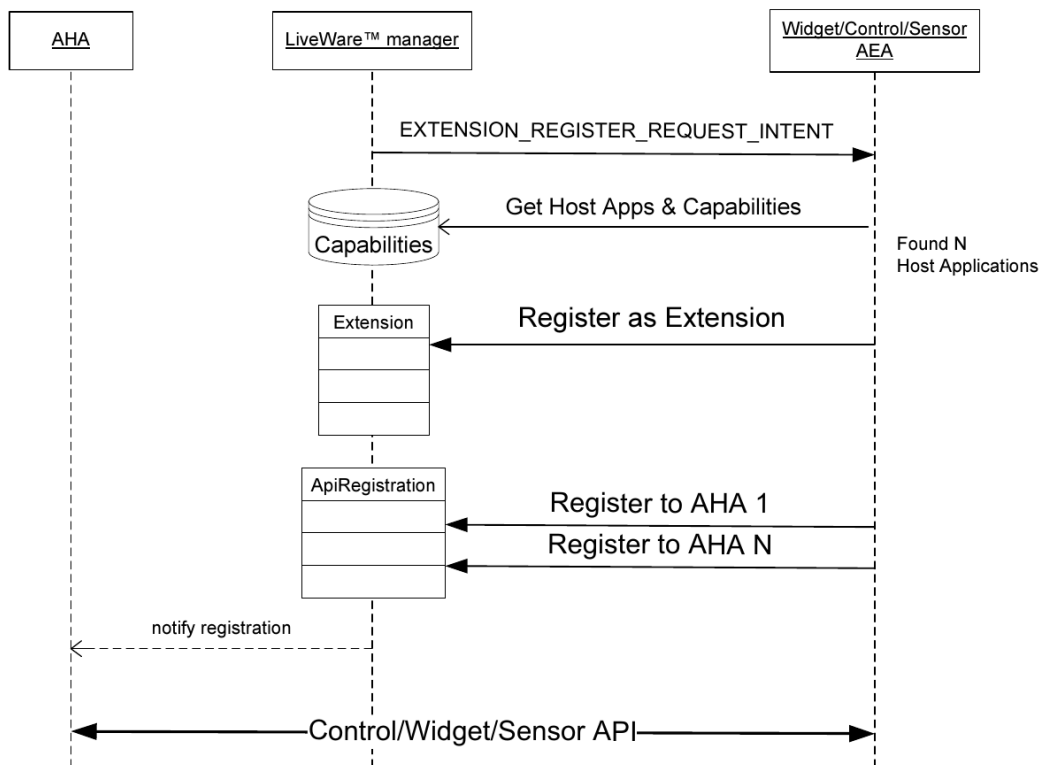


Abbildung 7: Der Smart Connect Extension Registrierungsprozess. Der LiveWare Manager ist der frühere Name von Smart Connect und wurde bis Android 3 verwendet (SonyMobile 2013b).

munikation zwischen SEA und SA. Falls eine höher priorisierte Applikation Zugriff benötigt, so sendet die AHA einen „CONTROL_PAUSE_INTENT“. Pausiert hat die Applikation keine Kontrolle über das Display, LEDs, Vibration oder Tasten. Sobald die AHA einen „CONTROL_RESUME_INTENT“ sendet, kann die SEA wieder den Betrieb aufnehmen. Wenn die SEA beendet wird, so sendet sie einen „CONTROL_STOP_REQUEST_INTENT“ and die AHA und erhält als Antwort darauf einen „CONTROL_STOP_INTENT“. Eine grafische Repräsentation der Lebensdauer einer SEA bietet Abbildung 8.

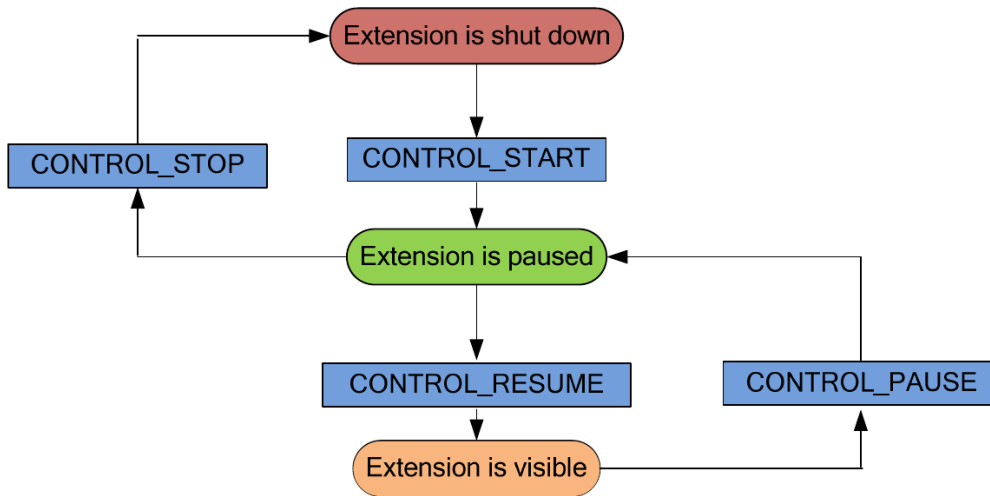


Abbildung 8: Lebensdauer einer Control Extension. Mögliche Zustände sind aktiv/sichtbar, pausiert/nicht sichtbar und ausgeschaltet (SonyMobile 2013b).

Tabelle 1: capabilities-Datenbank aus Smart Connect. Hier tragen SEA und AHA ihre Eigenschaften ein, diese sind somit für andere Applikationen einsehbar (SonyMobile 2013b).

Column	Info	Presence	Type	Data inserted by	Data readable by
name	Displayable name of the extension that may be presented, for example in settings	Required	Text	Extension	Smart Connect, host applications and the extension that added it

Column	Info	Presence	Type	Data inserted by	Data readable by
configurationActivity	Package name and Class name of the Activity that contains the settings for the extension	Optional	Text	Extension	Smart Connect, host applications and the extension that added it
configuration-Text	Short text to describe the current configuration state of the extension	Optional	Text	Extension	Smart Connect, host applications and the extension that added it
iconLargeUri	URI of the Android launcher icon representing the extension. This icon is used by the host application when listing extensions	Optional	Text	Extension	Smart Connect, host applications and the extension that added it
extensionIconUri	URI of the icon representing the extension. This icon is used on the accessory UI. The size is 34x34 pixels	Optional	Text	Extension	Smart Connect, host applications and the extension that added it

Column	Info	Presence	Type	Data inserted by	Data readable by
extensionIconUri-BlackWhite	URI of the monochrome icon representing the extension. This icon is used on the accessory UI. The size is 18x18 pixels	Optional	Text	Extension	Smart Connect, host applications and the extension that added it
extension_key	Used for security reasons for the extension's benefit. This key is sent as an extra-data in Intents sent to the extension from a host application. This enables the extension to verify that the sender has valid access to the registration content provider	Required	Text	Extension	Smart Connect, host applications and the extension that added it

Column	Info	Presence	Type	Data inserted by	Data readable by
notificationApiVersion	API Version. If the extension uses the notification API, this field should tell what version of the Notification API that it uses	Required	Integer	Extension	Smart Connect, host applications and the extension that added it
packageName	The package name of an extension. If an extension supports shared user id, the package name must be specified	Optional	Text	Extension	Smart Connect, host applications and the extension that added it

3.2 Manifest einer Applikation

Jede Android-Applikation muss eine Datei namens „AndroidManifest.xml“ in ihrem Wurzelverzeichnis vorliegen haben. Dieses Manifest enthält wichtige Informationen über die Applikation für das Android-System, die benötigt werden, bevor der Code der Applikation ausgeführt werden kann. Es enthält den Java-Paketnamen, welcher als eindeutige Identifikation der Applikation dient. Die unterschiedlichen Komponenten (*activities*, *services*, *broadcast receiver* und *content provider*) der Anwendung werden beschrieben und dem Android-System mitgeteilt. Somit sind die Komponenten und ihre Parameter bekannt und sie können entsprechend gestartet werden. Die benötigten *permissions* der Applikation zur Interaktion mit weiteren Apps werden genauso deklariert wie das minimale Level der Android-API. Außerdem werden die Bibliotheken aufgelistet, die mit der App verbunden sind.

Das Smart Extension Framework beruht auf Android API Level 7 (entsprechend Android Version 2.1). Ein Framework bietet eine wiederverwendbare und erweiterbare Struktur für Entwickler. Diese können es in ihre Applikation einbauen und es so erweitern, dass es ihren spezifischen Ansprüchen genügt. Alle Smart-Extension-Applikationen müssen, um Zugang zu den Registration- und Notification-Content-Providern zu erhalten, in der Manifest-Datei angeben, welche *permission* sie nutzen, in diesem Fall ist dies „com.sonyericsson.extras.liveware.aef.EXTENSION_PERMISSION“. Außerdem muss mindestens ein *service* deklariert sein und es muss ein *receiver* registriert werden. Alle Komponenten, die Intents zugeschickt bekommen, müssen einen *intent-filter* definieren. Mindestbestandteil dieses Filters muss „EXTENSION_REGISTER_REQUEST“ sein. Je nach Komplexität der Smart Extension App können noch mehr Filter dazukommen.

3.3 Die Klasse WatchExtensionReceiver

Die Datei WatchExtensionReceiver.java erweitert den android.content.BroadcastReceiver Service. Der *ExtensionReceiver* startet den *ExtensionService*, wenn irgendeiner der im Manifest angegebenen Intents empfangen wird. Er ist als statischer *broadcast receiver* implementiert, um die Beendigung des Service zu ermöglichen, falls keine Interaktion erfolgt. Ein dynamischer *receiver* erfordert einen durchgängig aktiven Service.

3.4 Die Klasse WatchExtensionService

Die Datei WatchExtensionService.java erweitert den com.sonyericsson.extras.liveware.extension.util.ExtensionService. Die Methode „createControlExtension()“ wird aufgerufen, wenn auf dem Display eine Control Extension angezeigt werden soll. *WatchExtensionService* gibt daraufhin ein *ControlSmartWatch*-Objekt zurück und über die *SmartExtensionUtils* können weitere Methoden wie „onStart()“ oder „onResume()“ aufgerufen werden. Diese legen fest, was beim Start oder der Aktivierung der SmartWatch geschieht.

3.5 Die Klasse WatchRegistrationInformation

Die Datei WatchRegistrationInformation.java erweitert den com.sonyericsson.extras.liveware.aef.util.registration.RegistrationInformation Service. *RegistrationInformation* ist eine abstrakte Klasse aus den *SmartExtensionUtils* und verwaltet die Registrierung der Smart Extension bei dem *Smart Connect registration content provider*.

3.6 Die Klasse ControlSmartWatch

Die Datei ControlSmartWatch.java erweitert die Klasse com.sonyericsson.extras.liveware.extension.util.control.ControlExtension. Hier können beispielsweise „on-Touch()“-Events bearbeitet werden oder auch komplexere Funktionen für die Smart-Watch. Informationen für das Display können als Bitmap mit der Funktion „show-Bitmap()“ gesendet werden. Um die zu sendende Datenmenge so gering wie möglich zu halten und gleichzeitig auch die Batterie der SmartWatch zu schonen, sollten jeweils nur die Daten neu gesendet werden, bei denen sich etwas geändert hat, also mehr ein Update des Displays bei dem nur Teile des Bildes neu gezeichnet werden als ein ganzes, vollkommen neues Bild.

4 Diskussion

Aktuelle AR-Applikationen auf mobilen Endgeräten machen sich die Video-See-Through-Technologie zu Nutze, mit welcher in das durch eine Kamera aufgenommene Bild weitere Informationen eingeblendet werden. Beispielsweise wäre hier die Applikation „Wikitude“ für Smartphones zu nennen, welche einen standortbezogenen Ansatz verfolgt und nach einer Positionsbestimmung per GPS, Kompass und Bewegungssensor entsprechend Informationen zur Umgebung einblendet (Wikitude 2009). Der Nutzer schaut durch die Kamera seines Handys und bekommt Informationen zu bekannten Gebäuden oder Restaurants in der Nähe angezeigt. Vorteile der Nutzung von mobilen Geräten ist eben diese Mobilität und die leichte Bedienung der integrierten Kamera, ein Nachteil ist der Zwang, das Gerät immer auf Augenhöhe vor sich halten zu müssen.

Head-Up-Displays sind schon seit den 1960er und 70er Jahren in der Luftfahrt in Benutzung und auch nicht mehr wegzudenken (Imhof 2012). Den Piloten werden Informationen wie Höhe, Geschwindigkeit und ein virtueller Horizont angezeigt. Seit den späten 80er Jahren wird diese Technologie auch in Autos verwendet, um dem Fahrer sicherheitsrelevante Informationen in sein Sichtfeld anzuzeigen (Pluta 2012a). Mit fortlaufender Verbesserung der verfügbaren Sensoren kann dies immer präziser geschehen. Infrarotsensoren oder Nachtsichtkameras erfassen bei schlechter Sicht wie starkem Nebel oder Dunkelheit Hindernisse oder Straßenschilder schneller und sicherer als der Fahrer selbst. Ein etwas anderer Ansatz wird von der amerikanischen Firma „Making Virtual Solid“ mit ihrem Produkt „Virtual Cable“ verfolgt, welches auf Piktogramme und Texteinblendungen verzichtet und eine schlichte Linie zur Navigation über die Straße projiziert (MVS 2001). Dies soll weniger ablenkend für den Fahrer sein und somit sicherer.

Für die Zukunft bietet die Augmented Reality-Technik vielfältigste Möglichkeiten einer sinnvollen und für den Menschen hilfreichen Nutzung. Die bereits vor ihrem für das Ende des Jahres 2013 bzw. Anfang 2014 angekündigten Verkaufstart für starke Kontroversen sorgenden Google Glass' werden den Anfang machen in einer Reihe von neuen AR-beeinflussten Entwicklungen. Sämtliche großen Technologiekonzerne wie Apple, Samsung, Google, Sony, LG und Motorola haben be-

reits offiziell Produkte aus dem Bereich „wearable computers“ angekündigt oder es wurden zumindest glaubhafte Gerüchte bekannt (ZeitOnline 2013; Ernst 2013; McKenzie 2013).

Von Smart Watches als weiteren „wearable computers“ versprechen sich die Unternehmen eine größere Akzeptanz der Nutzer im Gegensatz zu Datenbrillen wie z.B. Google Glass (Stangel 2013). Solche Uhren ermöglichen es, ständig auf dem Laufenden zu sein und keinerlei Nachrichten (SMS, Tweets, Mails) zu verpassen, ohne sein Smartphone aus der Tasche nehmen zu müssen. Und auch wenn es noch erforderlich ist, auf seinen Arm zu schauen, so ist diese Kopfbewegung den meisten Personen als Trägern von regulären Armbanduhren längst keine Unannehmlichkeit mehr. Smart Watches fügen sich somit nahtlos in das Erscheinungsbild des Nutzers ein und erfordern im Gegensatz zu AR-Brillen keine Änderung etablierter Verhaltensweisen. Abbildung 9 zeigt ein Umfrageergebnis von Juni 2013 über die Akzeptanz von „wearable computers“ in den USA.

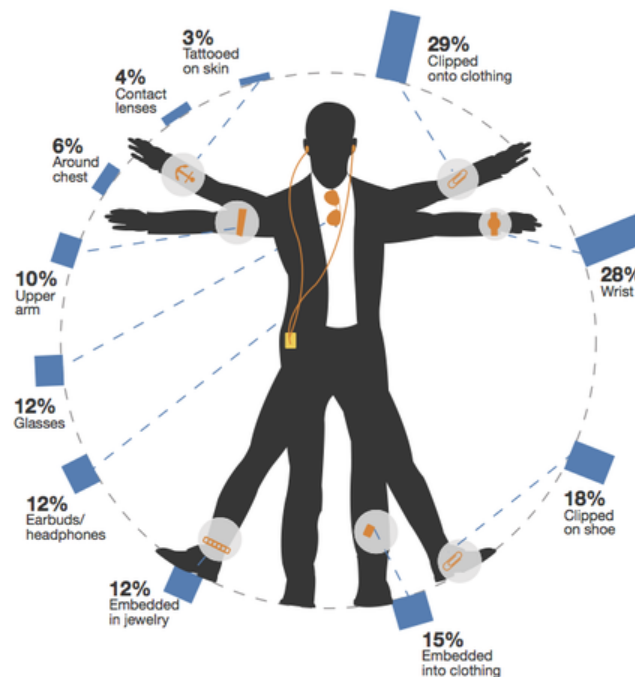


Abbildung 9: Akzeptanz von „wearable computers“ in den USA. Gegenüber Datenbrillen wie Google Glass herrscht noch eine allgemeine Skepsis, Smart Watches sind beliebter (Stangel 2013).

Head-Mounted-Displays wurden ab den 1970er Jahren erfolgreich zuerst im militärischen Bereich eingesetzt und seit den 1990er Jahren widmen sich ihnen auch immer mehr kommerzielle Unternehmen (Birt 1974). Sie erlauben einen freihändigen Einsatz in zahlreichen Bereichen:

- dem Mediziner können Patientendaten wie z.B. Röntgen- oder MRT-Bilder eingeblendet werden (Medgadget 2012; Suthau 2003; Donath 2013a; Grossmann 2013)
- dem Museumsbesucher werden Hintergrundinformationen zu Ausstellungsstücken angezeigt (Miyashita u. a. 2002)
- Mechaniker können sich verschiedene Bauteile von Maschinen optisch hervorheben lassen (Arvika 2001)
- Bedienungsanleitungen jeglicher Art können direkt visuell umgesetzt werden (Donath 2013d)
- Fernwartung durch Experten, welche einer hilfeschenden Person vor Ort durch Hinzufügen optischer Informationen weiterhelfen können (Arvika 2001)
- Navigation unabhängig vom Transportmittel, zu Fuß, auf dem Fahrrad oder beim Skifahren (Google 2013; UMass 2009; Donath 2013b; Oakley 2013)
- Architekten können ihre Modelle am erwünschten Bauort entwerfen und Archäologen sehen historische Stätten aus Ruinen wiederauferstehen (Dähne und Karigiannis 2002; Miyashita u. a. 2002)

Die gedankliche Weiterentwicklung von Datenbrillen ist, das Display entweder in Kontaktlinsen zu realisieren oder die Anzeige direkt auf die Netzhaut zu projizieren. Die Nutzung von Kontaktlinsen als Display, wenn auch bisher nur mit einem Pixel, ist bereits 2011 gelungen und verspricht für die Zukunft ähnliche Anwendungen wie bei Datenbrillen (Donath 2013c; Imec 2012; Pluta 2012b). Eine weitere Möglichkeit sind „Virtual Retinal Displays“, eine Anzeigetechnik, bei der ein Projektor ein Rasterbild, ähnlich wie ein Fernseher, direkt auf die Netzhaut abbildet und somit den Anschein einer vor dem Auge schwebenden Leinwand erweckt. Mit

Hilfe dieser Technologie könnten individuelle Sehschwächen wie Kurz- oder Weit-sichtigkeit oder nachlassende Netzhautsensitivität umgangen werden, da nicht der gesamte Sehapparat (Pupille, Linse, Cornea) in Anspruch genommen wird. Auch dies ist schon realisiert (Brother 2011).

Ein wichtiger Aspekt in Hinblick auf die Nutzung von AR-Techniken sind Fragen der Sicherheit, des Datenschutzes und damit einhergehend der Akzeptanz solcher Produkte durch die privaten Nutzer. Dadurch, dass die aktuell angekündigten AR-Brillen eine eingebaute Kamera beinhalten, regen sich schon vor Veröffentlichung Bedenken in Hinblick auf die Privatsphäre der Gefilmten. In den USA, in denen seit März 2013 bereits die Entwicklerversion von Google Glass an erste Testkandidaten ausgeliefert wurden, haben Casinos oder andere Lokalitäten bereits ein Verbot des Tragens von Google Glass ausgesprochen, mit Verweis auf potentielle Verletzungen der Privatsphäre (Clark 2013; Williams 2013). In Großbritannien ist das Tragen von Google Glass beim Autofahren verboten (Millward 2013). Dies basiert auf der Befürchtung, durch Einblendungen in das Sichtfeld des Nutzers diesen von der realen Welt abzulenken. Andererseits gibt es laut einem Bericht von Businessinsider einen potentiellen Markt im Jahr 2018 in der Größe von 21 Millionen Glass-Exemplaren im Wert von bis zu 10,5 Milliarden Dollar jährlich (BusinessInsider 2013). Dies hängt jedoch davon ab, wie sich der Preis der einzelnen Geräte entwickelt, wie schnell sie im Alltag von der Allgemeinheit akzeptiert werden und ob sich eine „Killer-App“, wie „Angry Birds“ für das Smartphone, finden lässt. Auch die Zahlen für Deutschland deuten nicht auf einen völligen Flop hin, zumindest 20% der Bevölkerung können sich vorstellen, in der Zukunft eine Datenbrille zu nutzen (Sawall 2013). Aufgrund dieser Aussichten halte ich die Erweiterung der eKaay-Applikation für den Einsatz mit AR-Geräten für eine Möglichkeit, neue Anwender zu finden. Die ersten Nutzer werden Datenbrillen vermutlich eher als ein Spielzeug sehen und neugierig die möglichen Apps ausprobieren. Wenn nun eine Tätigkeit, wie der Login in Online-Accounts, gleichzeitig schneller, bequemer und praktischer ausgeführt werden kann, so würde dies, meiner Meinung nach, positiv angenommen werden. Inwieweit die Idee, das Zahlenfeld der eKaay-Applikation auf einem Display als „Overlay“-Effekt über das Zahlenfeld auf einem Monitor darzustellen gelingt, muss erst noch herausgefunden werden.

Tabellenverzeichnis

- 1 Auszug aus der capabilities-Datenbank aus Smart Connect 17

Abbildungsverzeichnis

- 1 Augmented Reality bei Fernsehübertragungen am Beispiel von Football 6
- 2 Das eKaay-Login-Verfahren mit PIN-Eingabe 8
- 3 Produktabbildung der Sony SmartWatch 10
- 4 Überblick der Architektur der Smart Extensions 13
- 5 Übersicht der Kommunikation zwischen unterschiedlichen APIs 14
- 6 Die capabilities-Datenbank in Smart Connect 15
- 7 Registrierungsprozess bei Smart Connect 16
- 8 Lebensdauer einer Control Extension 17
- 9 Akzeptanz von „wearable computers“ in den USA 24

Literatur

- Arvika. 2001. Arvika. Verfügbar unter: <http://www.arvika.de/www/d/topic1/service.htm> ([27. Aug. 2013]).
- Birt, J. A. 1974. Visually coupled systems. Verfügbar unter: <http://www.best-of-flightgear.dk/vtasaur.htm> ([29. Aug. 2013]).
- Borchert, B. 2009. Ekaay - smart login. Verfügbar unter: <http://www.ekaay.com> ([29. Aug. 2013]).
- Borchert, B., und K. Reinhardt. 2007. Vorrichtung und verfahren zur abhör- und manipulationssicheren verschlüsselung für online-accounts. Patent DE 10 2007 052734B4 (DE), **patentfiled**6. Nov. 2007.
- Brother. 2011. Brother industries. Verfügbar unter: <http://www.brother.com/en/news/2011/airscouter/index.htm> ([29. Aug. 2013]).
- BusinessInsider. 2013. Bi intelligence forecast: google glass will be an 11 billion market by 2018. Verfügbar unter: <http://www.businessinsider.com/google-glass-11-billion-market-by-2018-2013-5> ([29. Aug. 2013]).
- Chip.de. 2010. Video: auto-frontscheibe mit augmented reality. Verfügbar unter: http://business.chip.de/news/Video-Auto-Frontscheibe-mit-Augmented-Reality_42023811.html ([11. Juli 2013]).
- Clark, M. 2013. Google glass violates nevada law, says caesars palace. Verfügbar unter: <http://www.ign.com/articles/2013/05/08/google-glass-violates-nevada-law-says-caesars-palace> ([29. Aug. 2013]).
- Dähne, P., und J. N. Karigiannis. 2002. Archeoguide: system architecture of a mobile outdoor augmented reality system. Verfügbar unter: <http://dl.acm.org/citation.cfm?id=854948> ([28. Aug. 2013]).
- Donath, A. 2013a. 3d-datenbrille für chirurgen. Verfügbar unter: <http://www.golem.de/news/sony-3d-datenbrille-fuer-chirurgen-1307-100603.html> ([24. Aug. 2013]).

- Donath, A. 2013b. Computerbrille für radfahrer. Verfügbar unter: <http://www.golem.de/news/recon-jet-computerbrille-fuer-radfahrer-1307-100109.html> ([1. Aug. 2013]).
- . 2013 c. Erster schritt zu google glass in kontaktlinsen. Verfügbar unter: <http://www.golem.de/news/graphen-und-silberdraht-erster-schritt-zu-google-glass-in-kontaktlinsen-1306-99700.html> ([20. Aug. 2013]).
- . 2013 d. Iphone-app erkennt das auto und erklärt es. Verfügbar unter: <http://www.golem.de/news/augmented-reality-iphone-app-erkennt-das-auto-und-erklaert-es-1308-100934.html> ([20. Aug. 2013]).
- Ernst, N. 2013. 5 millionen us-dollar für die alternative zu google glass. Verfügbar unter: <http://www.golem.de/news/telepathy-one-5-millionen-us-dollar-fuer-die-alternative-zu-google-glass-1308-101147.html> ([27. Aug. 2013]).
- Google. 2013. Google glass. Verfügbar unter: <http://www.google.com/glass/start/> ([29. Aug. 2013]).
- Grossmann, R. J. 2013. "ok glass: hand me the scalpel, please..." googleglass during surgery! Verfügbar unter: <http://rgrosssz.wordpress.com/2013/06/20/ok-glass-pass-me-the-scalpel-please-googleglass-during-surgery/> ([15. Aug. 2013]).
- Imec. 2012. Imec and ugent unveil breakthrough in augmented reality contact lens - curved lcd display holds widespread potential for medical and cosmetic applications. Verfügbar unter: http://www2.imec.be/be_en/press/imec-news/imecugentcontactlensdisplay.html ([23. Juni 2013]).
- Imhof, T. 2012. Mit dem tiefflug-display über die deutsche autobahn. Verfügbar unter: <http://www.welt.de/motor/news/article108347262/Mit-dem-Tiefflug-Display-ueber-die-deutsche-Autobahn.html> ([22. Aug. 2013]).

- McKenzie, H. 2013. Before google glass, there was terminator vision. now its maker focuses on enterprise. Verfügbar unter: <http://pandodaily.com/2013/07/12/before-google-glass-there-was-terminator-vision-now-its-maker-focuses-on-enterprise/> ([28. Aug. 2013]).
- Medgadget. 2012. Diagnosing medical problems in space using augmented reality. Verfügbar unter: <http://www.theatlantic.com/health/archive/2012/02/diagnosing-medical-problems-in-space-using-augmented-reality/252880/> ([24. Aug. 2013]).
- Migicovsky, E. 2013. Eric migicovsky: pebble watch interview - building a product that people want. Verfügbar unter: <http://drt.fm/eric-migicovsky-pebble-watch-interview-building-a-product-that-people-want/> ([28. Aug. 2013]).
- Millward, D. 2013. Drivers to be banned from wearing google glass. Verfügbar unter: <http://www.telegraph.co.uk/technology/news/10214822/Drivers-to-be-banned-from-wearing-Google-Glass.html> ([29. Aug. 2013]).
- Miyashita, T., P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov und S. Lieberknecht. 2002. An augmented reality museum guide. Verfügbar unter: <http://dl.acm.org/citation.cfm?id=1605355> ([28. Aug. 2013]).
- MVS. 2001. Virtual cable car navigation. Verfügbar unter: <http://mvs.net/index.html> ([28. Aug. 2013]).
- Nielsen. 2013. Deutsche legen ihr smartphone nicht mehr aus der hand. Verfügbar unter: <http://www.nielsen.com/de/de/insights/presseseite/2013/deutsche-legen-ihr-smartphone-nicht-mehr-aus-der-hand.html> ([28. Aug. 2013]).
- Oakley. 2013. Oakley airwave. Verfügbar unter: <http://de.oakley.com/airwave> ([29. Aug. 2013]).

- Pluta, W. 2012a. Augmented reality in der windschutzscheibe. Verfügbar unter: <http://www.golem.de/1201/89033.html> ([23. Aug. 2013]).
- . 2012 b. Us-wissenschaftler testen kontaktlinse mit bildschirm. Verfügbar unter: <http://www.golem.de/1111/87923.html> ([16. Aug. 2013]).
- Sawall, A. 2013. Jeder fünfte in deutschland würde google glass nutzen. Verfügbar unter: <http://www.golem.de/news/datenbrillen-jeder-fuenfte-in-deutschland-wuerde-google-glass-nutzen-1305-99269.html> ([28. Aug. 2013]).
- SonyMobile. 2013a. Smartwatch - sony smartphones (germany). Verfügbar unter: <http://www.sonymobile.com/de/products/accessories/smartwatch/> ([27. Aug. 2013]).
- . 2013 b. Sony add-on sdk. Verfügbar unter: <http://developer.sonymobile.com/knowledge-base/sony-add-on-sdk/> ([27. Aug. 2013]).
- Stangel, L. 2013. 21.6 million geeky americans want google glass right now. Verfügbar unter: <http://www.bizjournals.com/sanjose/news/2013/06/21/216-million-geeky-americans-want.html?page=all> ([29. Aug. 2013]).
- Sung, D. 2011. What is augmented reality? Verfügbar unter: <http://www.pocketlint.com/news/108880-what-is-augmented-reality-ar> ([12. Aug. 2013]).
- Suthau, T. 2003. Augmented reality techniken für den einsatz in der leberchirurgie. Verfügbar unter: http://www.cv.tu-berlin.de/fileadmin/fg140/Augmented_Reality_Techniken.pdf ([25. Aug. 2013]).
- UMass. 2009. Personal head-up display. Verfügbar unter: <http://www.ecs.umass.edu/ece/sdp/sdp09/wolf/index.html> ([14. Aug. 2013]).
- Wikitude. 2009. Wikitude - the app. Verfügbar unter: <http://www.wikitude.com/app/> ([28. Aug. 2013]).

Williams, R. 2013. Google glass will make 'privacy impossible' warn 'stop the cyborgs' campaigners. Verfügbar unter: <http://www.independent.co.uk/life-style/gadgets-and-tech/news/google-glass-will-make-privacy-impossible-warn-stop-the-cyborgs-campaigners-8550499.html> ([29. Aug. 2013]).

ZeitOnline. 2013. Google glass bekommt konkurrenz. Verfügbar unter: <http://www.zeit.de/digital/mobil/2013-08/datenbrillen-meta-spaceglases-oculus-rift/komplettansicht> ([17. Aug. 2013]).