

Konzeption und Implementierung der Swipe- Geste zum Datentransfer im Multi-Screen Kontext

Alexander Hahn

Bachelor-Thesis

Studiengang Informatik

Fakultät für Informatik

Hochschule Mannheim

08.06.2015

Betreuer:

Prof. Kirstin Kohler, Hochschule Mannheim

Horst Schneider, B.Eng, Hochschule Mannheim

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mannheim, 08.06.2015

Alexander Hahn

Abstract

In der vorliegenden Arbeit wird die smartphonebasierte Geste „Swipe To Give“ konzipiert. Als praktischer Teil wird das Konzept prototypisch implementiert.

Ziel dieser Arbeit ist die Ausarbeitung eines detaillierten Interaktionskonzeptes. Hierbei wird auch die technische Umsetzbarkeit betrachtet und prototypisch implementiert. Die Erfahrungen, basierend auf der Konzeption und Implementierung, werden in einem Pattern-Template wohl strukturiert aufbereitet.

Diese Art von Interaktion orientiert sich an unseren Erfahrungen der realweltlichen Interaktion mit Objekten und der sozialen Kommunikation und vermischt diese bzw. reichert diese um digitalweltliche Möglichkeiten an.

Dabei wird auf eine Verständlichkeit geachtet, sodass diese Beschreibung sowohl von Konzeptern als auch von Programmierern bei der Umsetzung einer Swipe-To-Give Geste im Rahmen eines konkreten Anwendungskontextes herangezogen werden kann.

Die Implementierung unterstützt mobile Endgeräte der neueren Generation mit dem Betriebssystem „Android“. Der Quellcode wird als Open Source Projekt zur Verfügung gestellt.

Inhalt

1	Einleitung.....	6
1.1	Motivation	6
1.2	Zielsetzung	6
1.3	Inhalt und Struktur	7
2	Stand der Technik.....	8
2.1	Verwandte Arbeiten und Applikationen.....	8
2.2	Technische Grundlagen	11
2.2.1	Touchscreen.....	11
2.2.2	Peer-to-Peer Verbindung.....	12
2.2.3	Sonstige Verbindungen	14
2.2.4	Datenkommunikation zwischen mobilen Endgeräten	15
3	Konzept der „Swipe To Give“-Geste	18
3.1	Ablauf der Interaktion	18
3.2	Geräte sehen / Awareness	21
3.3	Auswahl der Dateien / Reveal	22
3.4	Geste zum Senden / Transfer.....	23
3.5	Reaktion des Systems.....	25
3.5.1	Reaktion des Systems beim Sender.....	25
3.5.2	Reaktion des Systems beim Empfänger.....	25
3.6	Geeignete Endgeräte	26

3.7	Schlussfolgerung für die Implementierung.....	27
4	Umsetzung der Interaktion	28
4.1	Vorbedingung	28
4.2	Bestimmung der Verbindung	28
4.3	Geräte sehen / Awareness	29
4.4	Auswahl der Dateien / Reveal	33
4.5	Geste zum Senden.....	34
4.6	Datenübertragung.....	36
5	Demonstrator Applikation	38
5.1	Startgerät.....	39
5.2	Zielgerät.....	42
6	Zusammenfassung und Ausblick	44
	Abkürzungsverzeichnis	vi
	Tabellenverzeichnis.....	vii
	Abbildungsverzeichnis	viii
	Literaturverzeichnis.....	ix
	Anhang 1 - Interaktionspattern.....	xi

1 Einleitung

1.1 Motivation

90% aller medialen Interaktionen sind Bildschirm-basiert. Hierfür wird jedoch nicht mehr nur der Desktop-PC genutzt. Es gibt eine Vielzahl verschiedener Endgeräte, um den täglichen Aktivitäten nachzugehen. Abhängig vom Kontext, das heißt von Ziel, Aufenthaltsort und Zeitpunkt, entscheiden Anwender, welche Geräte sich für eine Situation am besten eignen und treffen eine dementsprechende Auswahl. [1]

Bei der Erledigung einer Aufgabe kann es vorkommen, dass mehrere Endgeräte gleichzeitig oder nacheinander genutzt werden. Ein Vorteil der unterschiedlichen Geräte ist unter anderem die Bildschirmgröße oder die Interaktionsmöglichkeit. Die Aktivitäten starten häufig auf einem Gerät und werden später auf einem anderen Gerät fortgeführt. Beispielsweise wechseln 61% beim Online-Shopping vom Smartphone zum PC. [1]

Beim Gerätewechsel müssen meist erst Daten übertragen werden. Der Ablauf des Datenaustauschs ist kompliziert und störend. Mit der Gestensteuerung, insbesondere der „Swipe-To-Give“-Geste, soll dieser Vorgang vereinfacht werden.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Ausarbeitung eines detaillierten Interaktionskonzeptes. Hierbei wird auch die technische Umsetzbarkeit betrachtet und prototypisch implementiert. Die Umsetzung sieht vor, dass Bilder, die sich auf einem Gerät befinden, per Swipe-Geste auf ein anderes Endgerät übertragen werden können. Jedoch können durch das Konzept auch andere Dateien übertragen wer-

den, wie z.B. Videos, Apps, usw. Die Erfahrungen, basierend auf der Konzeption und Implementierung, werden in einem Pattern-Template wohl strukturiert aufbereitet.

1.3 Inhalt und Struktur

In Kapitel zwei werden verwandte Arbeiten und Applikationen zum Thema, sowie technische Grundlagen beschrieben. Kapitel drei analysiert und beschreibt die Swipe-Interaktion. Hierbei werden neben technischen Aspekten auch die geeigneten Endgeräte beschrieben. Kapitel vier beschreibt einleitend die Vorbedingungen, welche für eine erfolgreiche Ausführung der Interaktion gelten müssen. Anschließend werden Peer-to-Peer Verbindungen untersucht die zum Datenaustausch verglichen werden. Des Weiteren wird auf die Geste zum Senden, den Swipe, eingegangen. Die Zusammenfassung aus Kapitel drei und vier wird in einem Interaktionspattern aufbereitet. Dies ist dem Anhang dieser Arbeit beigelegt. In Kapitel fünf wird die entwickelte Demonstrator Applikation vorgestellt. Abschließend bietet Kapitel sechs eine Zusammenfassung über die erarbeiteten Ergebnisse und einen Ausblick auf zukünftige potenzielle Weiterentwicklungen der Arbeit. Die Bedeutungen der verwendeten Abkürzungen befinden sich im Abkürzungsverzeichnis und ein Überblick der dargestellten Abbildungen im Abbildungsverzeichnis sowie der Tabellen im Tabellenverzeichnis. Zudem ist die verwendete Literatur im Literaturverzeichnis angegeben.

2 Stand der Technik

In diesem Kapitel werden verwandte Arbeiten und Applikationen zum Thema, sowie technisches Grundlagenwissen beschrieben, welches zur Umsetzung der Interaktion benötigt wird.

2.1 Verwandte Arbeiten und Applikationen

Anhand der Studie „**Gradual Engagement: Facilitating Information Exchange between Digital Devices as a Function of Proximity**“ [13], die in Kooperation der University of Calgary und der Microsoft Research Abteilung durchgeführt wurde, entstand ein Prototyp, der mehrere Aspekte der Interaktion mobiler Geräte mit fest-angebrachten Geräten berücksichtigt.

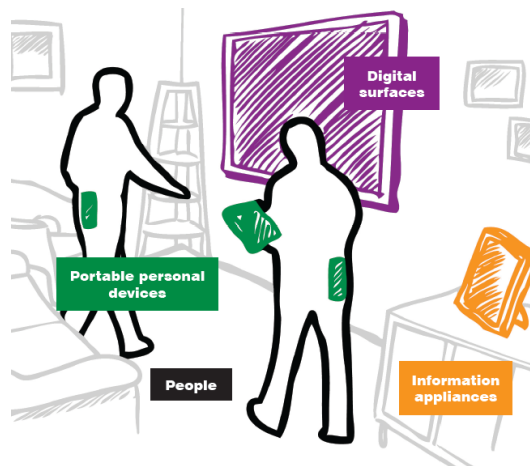


Abbildung 2.1: Mobile und stationary (fest-angebrachte) Geräte [15]

Zu Beginn existierten folgende Fragen die zu klären waren:

1. Wie erkennt der Nutzer, dass sein mobiles Gerät mit einem fest-angebrachten Gerät interagieren kann?
2. Wie können Daten von einem zum anderen Gerät übertragen werden?

Um eine mögliche Interaktion aufzuzeigen, wählten die Entwickler eine visuelle Lösung. Ist ein kompatibles Gerät in der Nähe, so wird eine Repräsentation auf dem Bildschirm angezeigt. Die auf dem Gerät befindlichen Bilder werden direkt angezeigt, sobald die Person sich dem TV nähert.

Hoccer [8] ist eine Applikation, die über zwei Gesten verfügt, mit denen Daten zwischen Endgeräten ausgetauscht werden können. Zum einen gibt es die Swipe Geste und zum anderen eine Wurf und Fang-Geste. Bei der Wurf und Fang-Geste, führt ein Anwender die Wurf-Geste aus und wirft einem zweiten Anwender die Daten, wie mit einem Frisbee, zu.

Mit der Swipe Geste wird über den Touchscreen eine geöffnete Datei von einem Gerät auf das andere geschoben. Die Geräte müssen hierfür nebeneinander liegen. Die Richtung des Swipes ist wichtig. Wird ein Bild in die entgegengesetzte Richtung geschoben, so kommt es auf dem Zielgerät nicht an. Das Swipen ist bei dem ausgewählten Bild nur für die Datenübertragung reserviert. Scrollen ist hierbei nicht möglich und auch nicht nötig, da nur ein Bild übertragen werden kann.

Die Applikation kommuniziert über die GPS-Standortdaten mit einem Server, der Geräte zuordnet und den Datenverkehr steuert. Eine lokale Kommunikation ist nicht möglich. Hoccer ist für Apple- und Android-Geräte verfügbar und kann unter folgender URL gedownloadet werden: <http://hoccer.com/de/#section-download>

Mit der Applikation **Push2Send** [7] ist es ebenfalls möglich, Bilder und Videos per Swipe zu übertragen. Hierbei werden die gewünschten Dateien an das obere Ende des Displays geschoben. Es spielt keine Rolle, wo das Zielgerät liegt; die Dateien kommen trotzdem an.

Die Kopplung und das Versenden erfolgt entweder über Bluetooth oder über WiFi. Die App gibt es bislang nur für iOS. Geräte mit einem anderen Betriebssystem werden nicht unterstützt. [4] Die App wird nicht im App-Store von Apple

angeboten, kann jedoch auf eigenes Risiko unter folgender URL gedownloaded werden: <http://www.appato.com/s-s-royal-ltd/push2send/>

FotoSwipe [17] bietet ebenfalls die Möglichkeit, Bilder per Swipe zu übertragen. Die Bilder werden aus der App heraus nach rechts oder links geschoben (siehe Abbildung 2.2). Hierbei können auch mehrere Bilder übertragen werden. Für die Dateiübertragung wird zwingend eine Internetverbindung und GPS benötigt. Die App kommuniziert mit einem Server um die Dateien auszutauschen. Eine Adhoc-Verbindung ist nicht möglich. FotoSwipe kann auch zwischen unterschiedlichen Betriebssystemen genutzt werden.

Die Installationen für die unterschiedlichen Betriebssysteme, sind auf der Webseite zu finden: <http://fotoswipe.com>

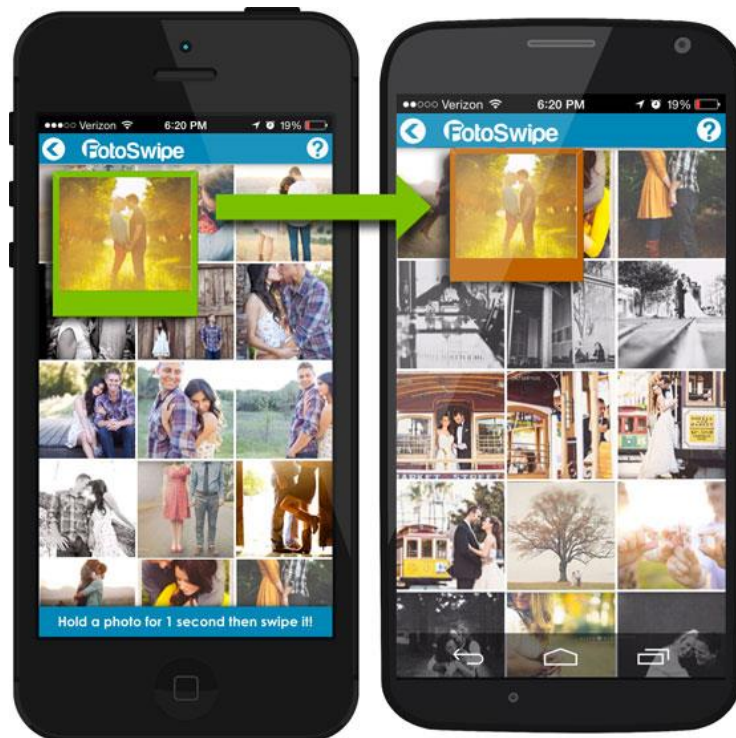


Abbildung 2.2: FotoSwipe Demo [17]

2.2 Technische Grundlagen

2.2.1 Touchscreen

Mobile-, aber auch fest angebrachte Geräte, verfügen über Touchscreen-Displays. Hierbei unterscheidet man zwischen resistiven und kapazitiven Displays. Nachfolgend werden die Vor- und Nachteile betrachtet.

Resistive Touchscreens

Ein resistiver Touchscreen reagiert auf Druck. Dabei ist es nicht wichtig, ob mit dem ganzen Finger oder einem Stift aufgedrückt wird. Ein solcher Touchscreen besteht aus zwei Schichten. Die obere Schicht besteht aus Polyester, die darunter liegende meist aus Glas. Eine dieser beiden Schichten wird unter Gleichspannung gesetzt, meist die untere Glasschicht. Werden nun die Schichten zusammengedrückt, kann die Spannung an der oberen Polyesterfolie an den Rändern gemessen werden und so die Position der Druckstelle bestimmen. [16]

Kapazitive Touchscreens

Im Gegensatz zu einem resistiven Touchscreen funktioniert ein kapazitiver Touchscreen auch ohne Druck. Er reagiert bereits auf Berührung. Diese Technik ist in den meisten neuen Smartphones verbaut, da es in der Benutzerinteraktion deutlich komfortabler ist. Der Bildschirm besteht nicht aus mehreren Schichten, sondern nur noch aus einer Glasplatte. Auf diese wird eine durchsichtige leitfähige Folie aufgebracht. Über Wechselstrom entsteht ein elektrisches Feld, welches auf Berührung reagiert. Der entsendete Stromfluss wird an den Ecken gemessen, um die Position des Fingers zu ermitteln. [16]

2.2.2 Peer-to-Peer Verbindung

Mit dem Begriff Peer-to-Peer (P2P) ist die Vorstellung verbunden, dass in einem Verbund Gleichberechtigter („Peers“), die sich wechselseitigen Ressourcen wie Informationen, Speicher und Bandbreite zugänglich machen, kollaborative Prozesse durchgeführt werden. Hierbei wird auf zentrale Koordinationsinstanzen verzichtet. [18] Nachfolgend werden Technologien erläutert, die für eine P2P Verbindung genutzt werden können.

Bluetooth Low Energy

Bluetooth ist ein Funksystem zur Datenübertragung über Distanzen von bis zu 30 Metern. Ab der Version 4.0 spricht man auch von Bluetooth Low Energy, da ab dieser Version die Technologie auf sehr niedrigen Stromverbrauch optimiert wurde. Um miteinander zu kommunizieren bauen Endgeräte eine Peer to Peer Verbindung auf und organisieren sich in sogenannten Pico Netzen (siehe Abbildung 2.3). [2]

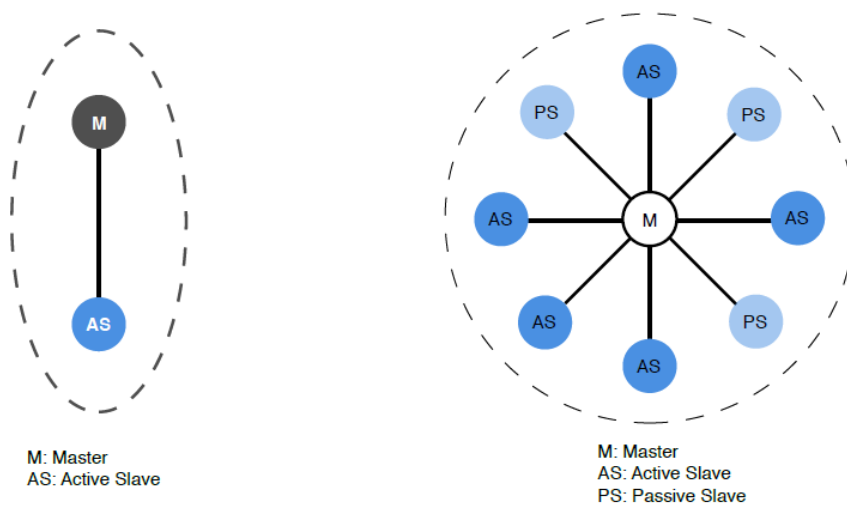


Abbildung 2.3: Piconetz / Bluetooth-Topologie [2]

Eine detaillierte Beschreibung von Bluetooth Low Energy wurde bereits in diversen Arbeiten verfasst, u.a. in „*Implementierung einer gestenbasierten Interaktion zum Datenaustausch zwischen mobilen Endgeräten*“ [6], weshalb hier nicht näher darauf eingegangen wird.

WiFi Direct

WiFi Direct unterstützt den direkten Austausch von Daten zwischen Geräten, ohne sich vorher, wie bei den aktuellen WLAN-Standards, in einem Netzwerk anmelden zu müssen. Die Technik baut dabei aber auf die WLAN-Technik, die es bereits gibt, ist aber nicht zu verwechseln mit dem WLAN-Ad-Hoc-Modus. Es können sich zwei oder auch mehrere Geräte gleichzeitig über WiFi Direct verbinden.

Die Wi-Fi Alliance sieht die Technologie in Zukunft in Geräten wie Mobiltelefonen, Kameras, Drucker, PCs und auch in Spielekonsolen. Die Einsatzzwecke sind dementsprechend vielfältig, z.B. einfacher Datenaustausch, Synchronisation des Smartphones oder Spiele im Multiplayer-Modus spielen. WiFi Direct soll zu den bisherigen WLAN-Standards kompatibel sein und man kann WiFi-Direct-Geräte mit anderen WLAN-Geräten verbinden.

Da Wi Fi Direct eine ähnliche Reichweite hat, wie bisherige WLAN-Geräte (maximal 200 Meter), ist dies neben dem höheren Sicherheitsstandard und der angestrebten leichteren Verbindbarkeit ein weiterer Unterschied gegenüber der Bluetooth-Technologie. [3]

WiFi Direct ist verfügbar auf Geräten mit Android oder Windows Betriebssystemen.

Near-Field-Communication

Near Field Communication, kurz NFC, ist eine drahtlose Übertragungstechnik, die zum kontaktlosen Datenaustausch zwischen Geräten mit einer Distanz von bis zu vier Zentimeter dienen soll. NFC stammt aus dem Jahr 2002 und wurde von der ehemaligen Philips-Tochter NXP und Sony entwickelt. Dabei wurde auf Standards wie Bluetooth und RFID zurückgegriffen. An der Weiterentwicklung von NFC sind Philips, Sony und weitere im NFC-Forum organisierten Firmen beteiligt.

NFC kann bis zu 424 kBit/s übertragen, was für umfangreiche Daten, wie Bilder oder Videos nicht ausreichend ist. Die typischen Anwendungen von NFC ist

deshalb der Austausch von Informationen zwischen zwei nahe aneinander gehaltenen Geräten. Beispielsweise kann man so den Zugriff auf Inhalte gewähren und Dienste, wie bargeldlose Zahlungen oder Ticketing, umsetzen.

Der Funkstandard NFC ist gezielt auf eine geringe Reichweite im Zentimeterbereich entwickelt worden, um das Ausspähen der übertragenen Daten zu erschweren. Durch die extrem kurze Distanz sind unbeabsichtigte Verbindungen nahezu ausgeschlossen. [3]

2.2.3 Sonstige Verbindungen

WLAN

Wireless LAN (Wireless Local Area Network) ist als Oberbegriff für alle auf dem Markt befindlichen drahtlosen lokalen Datennetze zu verstehen. Darunter fällt beispielsweise auch Bluetooth, sowie alle anderen Techniken und Standards mit denen sich drahtlose Funknetzwerke aufbauen lassen. Anstatt Daten über ein Kabel zu übertragen, dient die Luft als Übertragungsmedium und als Schnittstelle, die sich per Funk nutzbar machen lässt. Diese Ungebundenheit erlaubt ungeahnte Möglichkeiten der Mobilität und des Komforts. [11] Ein solches Netzwerk kann entweder im Infrastruktur-Modus oder im Ad-hoc-Modus betrieben werden.

Die Vor- und Nachteile dieser Modi, wurden bereits in diversen Arbeiten betrachtet, u.a. in *„Implementierung einer gestenbasierten Interaktion zum Datenaustausch zwischen mobilen Endgeräten“* [6], weshalb hier nicht näher darauf eingegangen wird.

Mobilfunk

Moderne Mobilfunknetze bieten neben dem Zugang zu Telefondiensten auch Zugang zu Internetdiensten. Die aktuellsten Mobilfunkstandards sind High Speed Packet Access (HSPA+) und Long Term Evolution (LTE). Bei HSPA+ handelt es sich um eine Erweiterung des Mobilfunkstandards UMTS

mit Datenraten von bis zu 42 Mbit/s. LTE dagegen ist ein komplett neu entwickeltes Mobilfunknetz, das in der aktuellen Spezifikation theoretische Datenraten von bis zu 75 Mbit/s im Upload und 300 Mbit/s im Download bietet. [11]

2.2.4 Datenkommunikation zwischen mobilen Endgeräten

Abhängig von der Technologie, können die Geräte entweder direkt im Netzwerk miteinander kommunizieren oder über einen Server. Folgend wird beschrieben, wie die Kommunikation jeweils umgesetzt werden kann.

Kommunikation in lokalen Netzwerken

Um eine Kommunikation in mobilen Applikationen herzustellen, werden Netzwerkdienste benötigt. Ein Netzwerkdienst ist ein Objekt, welches den Anwendern bzw. teilnehmenden Geräten Informationen bereitstellt. Jeder Dienst stellt eine Funktionalität zur Verfügung, dabei kann es sich z.B. um Anwendungsdienste oder Kommunikationsdienste handeln [11].

Um diese angebotene Dienste im Netzwerk finden zu können, wird die service discovery Komponente benötigt, für die verschiedene Protokolle existieren. Apple nutzt in seinen Betriebssystemen Bonjour, Android nutzt Network Service Discovery. Beide Protokolle ermöglichen das automatische Entdecken von Geräten und Diensten in lokalen Netzwerken.

Network Service Discovery

Durch das Hinzufügen des Network Service Discovery (NSD), können Benutzer einer Anwendung andere Geräte, die die Dienste Ihrer App-Anforderungen unterstützen, im lokalen Netzwerk identifizieren [10]. Dies ist nützlich für eine Vielzahl von Peer-to-Peer-Anwendungen wie Filesharing oder Multiplayer-Spiele. Android NSD APIs vereinfachen den Aufwand um solche Features zu implementieren. Unter die Kategorie Peer-to-Peer Verbindung zählen, wie in Kapitel 2.2.2 beschrieben, die Verbindungen via WiFi Direct, Bluetooth und

NFC. Für die jeweilige Verbindungsart existieren unterschiedliche Erweiterungen.

Sockets

Ein Socket dient zur Abstraktion und ist ein Verbindungspunkt in einem TCP/IP-Netzwerk. Werden mehrere Computer verbunden, so implementiert jeder Rechner einen Socket: Derjenige, der die Verbindung initiiert und Daten sendet, einen Client-Socket und derjenige, der auf eingehende Verbindungen horcht, einen Server-Socket. Es lässt sich in der Realität nicht immer ganz trennen, wer Client und wer Server ist, da Server zum Datenaustausch ebenfalls Verbindungen aufbauen können. Doch für den Betrachter von außen ist der Server der Wartende und der Client derjenige, der die Verbindung initiiert.

Socket-Verbindungen bleiben dauerhaft bestehen und werden erst geschlossen, wenn der Client die Verbindung schließt. Des Weiteren können Socket-Verbindungen lokal, jedoch auch über einen Server stattfinden.

Datenkommunikation über einen Server

Wenn die Kommunikation in einem lokalen Netzwerk nicht gewünscht oder umzusetzen ist, können Applikationen auch über einen Server Daten austauschen. Das Standardmodell für diese Kommunikation ist das Client-Server-Modell (siehe Abbildung 2.4). In diesem Modell rufen Clients bestimmte Funktionalitäten bzw. Dienstleistungen, die ein Server zur Verfügung stellt, über ein Netzwerk auf.

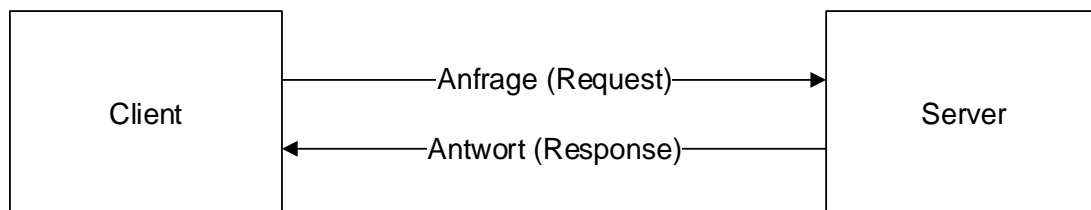


Abbildung 2.4: Client-Server-Modell

Eine detaillierte Beschreibung zum Client-Server-Modell wurde bereits in diversen Arbeiten verfasst, u.a. in „*Implementierung einer gestenbasierten Interaktion zum Datenaustausch zwischen mobilen Endgeräten*“ [6], weshalb hier nicht näher darauf eingegangen wird. Dort wird auch auf die HTTP-Operationen eingegangen.

3 Konzept der „Swipe To Give“-Geste

In diesem Kapitel werden die Komponenten für eine erfolgreiche Konzipierung der Geste identifiziert und beschrieben. Die Swipe-Geste ist eine Interaktion, bei der Dateien von einem Startgerät auf ein Zielgerät übertragen werden. Der Swipe initiiert die Datenübertragung. Um genau zu klären wie dieser Vorgang funktioniert, wird unter anderem der Ablauf bestimmt und untersucht, welche Endgeräte sich für die Geste eignen. Eine Zusammenfassung der Ergebnisse aus diesem Kapitel findet sich im Interaktionspattern, welches dem Anhang beigelegt ist.

3.1 Ablauf der Interaktion

Nachfolgend ist ein allgemeingültiger Ablauf für eine gestenbasierte Interaktion (siehe Abbildung 3.1). Um einen besseren Überblick zu erhalten, wurden Aktivitäten in eine Subaktivität ausgelagert (siehe Subaktivität V).

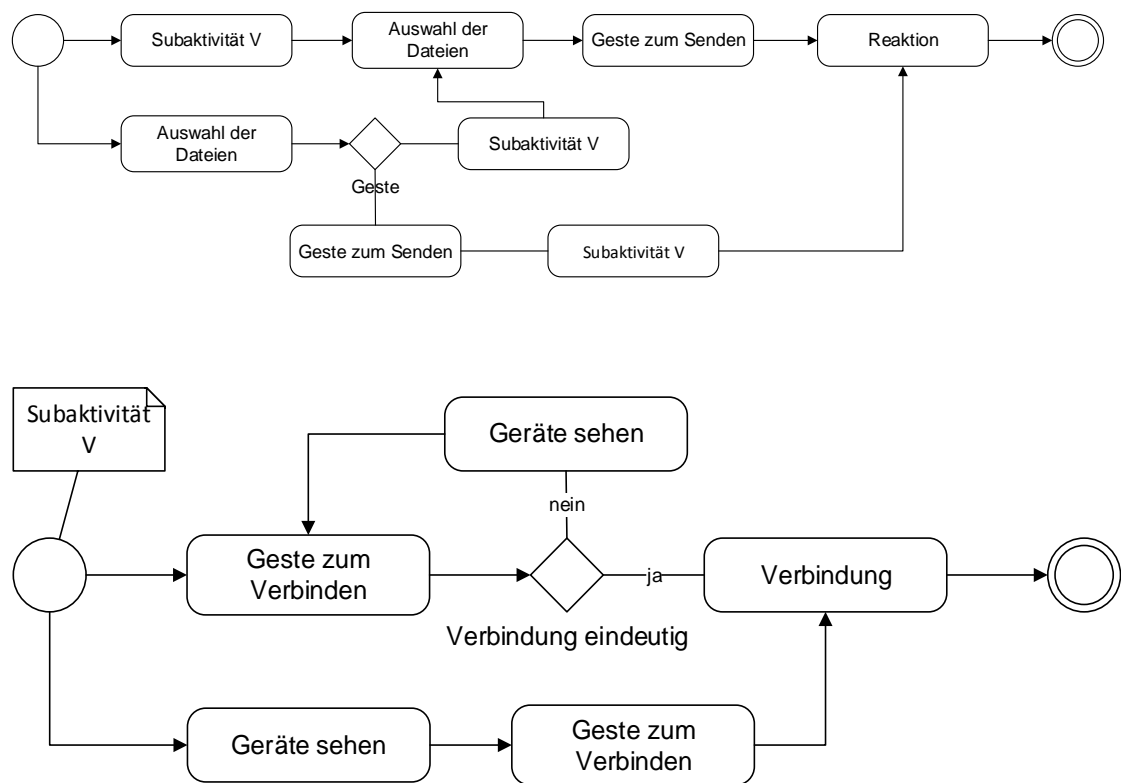


Abbildung 3.1: Aktivitätsdiagramm einer gestenbasierten Interaktion

Dieser Ablauf ist für alle Patternkandidaten gültig, die zuvor bereits in einer Studenarbeit [19] identifiziert wurden. Für die hier zu konzipierende Interaktion „Swipe To Give“, wird der Ablauf wie folgt festgelegt (siehe Abbildung 3.2). Dies ist eine Variation des zuvor gezeigten allgemeingültigen Ablaufs. Diese Variation berücksichtigt u.a. die Besonderheit, dass eine Verbindung nicht eindeutig ist und das Zielgerät aus einer Liste ausgewählt werden muss.

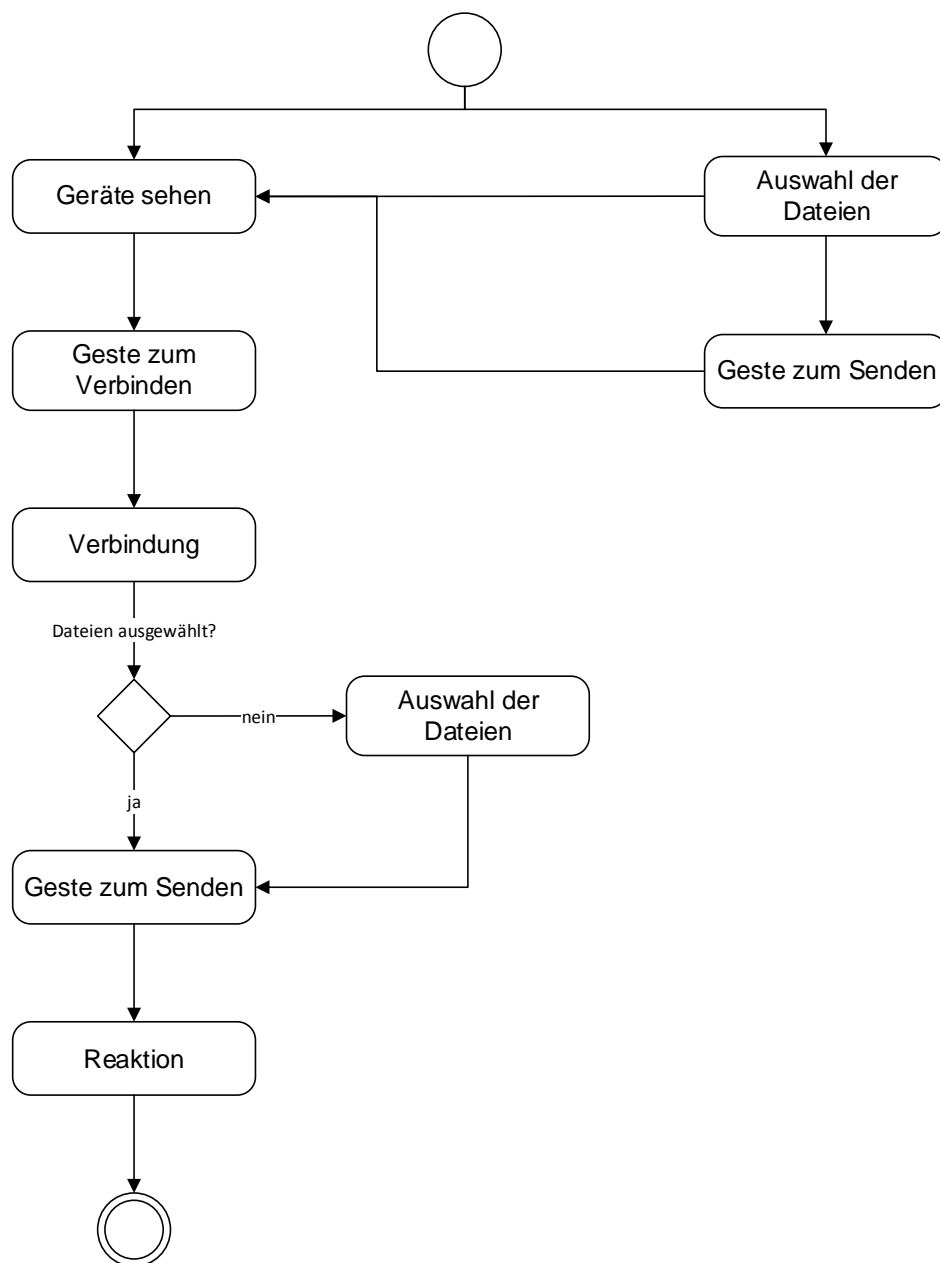


Abbildung 3.2: Aktivitätsdiagramm "Swipe To Give"

Alle Aktionen vor der „Geste zum Senden“ sind Vorbedingungen, die zwingend für eine erfolgreiche Ausführung der Interaktion durchgeführt werden müssen. In den folgenden Abschnitten wird auf die Komponenten und Aktionen eingegangen, die im Ablauf zu finden sind.

3.2 Geräte sehen / Awareness

In der bereits im zweiten Kapitel vorgestellten Arbeit „Gradual Engagement: Facilitating Information Exchange between Digital Devices as a Function of Proximity“ wurden drei Stufen eingeführt, die durch die Interaktion führen sollen. Die erste Stufe wurde Awareness genannt. Hierbei sollen Geräte identifiziert und angezeigt werden, mit denen das eigene Gerät kommunizieren kann. In dem vorgestellten Paper ist die Rede von Proxemik. Hierbei werden Distanzen in gewisse Bereiche unterteilt.

Bezeichnung	Entfernung	Beispiel
Intimate distance	Bis max. 46 cm	Umarmen oder Flüstern
Personal distance	46 – 122 cm	Interaktionen zwischen guten Freunden oder Familienmitgliedern
Social distance	1,2 – 3,7 m	Interaktionen zwischen Bekannten
Public distance	Ab 3,7 m	Reden in der Öffentlichkeit

Tabelle 3.1: Edward T. Hall's Definition des persönlichen Bereichs [12]

Für die Swipe to Give Geste können alle beschriebenen Kategorien zutreffen. Es muss jedoch unterschieden werden, wie das Zielgerät identifiziert wird. Man unterscheidet zwischen Interaktionen, bei denen das Zielgerät eindeutig identifiziert werden kann und solchen, bei denen der Benutzer anhand einer Liste verfügbarer Geräte das Zielgerät auswählt.

Ersteres kann z.B. ein „Bump“ sein, welches bereits in einer früheren Arbeit [6] betrachtet wurde. Die beiden Geräte werden aneinander gestoßen, wodurch eine Koppelung anhand der Beschleunigungssensoren im Gerät identifiziert

werden kann (siehe Abb. 3.3). Diese Interaktion liegt im Bereich der Intimate distance oder der Personal distance.

Sollte der Benutzer aus einer Liste auswählen, so ist die Geste zum Verbinden meist ein Klick auf einen „Verbinden“-Button. Hierbei können zusätzlich die Bereiche Social distance und Public distance zutreffen. Eine Verbindung wird vorausgesetzt, damit die Interaktion, der Swipe, erfolgreich durchgeführt werden kann.

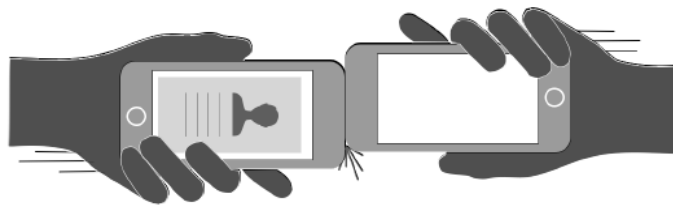


Abbildung 3.3: Bump zweier Geräte zur eindeutigen Identifizierung [6]

3.3 Auswahl der Dateien / Reveal

Die zweite Stufe beschäftigt sich mit der Auswahl der zu übertragenen Dateien. Nachdem die Geräte verbunden sind, können Dateien, welche sich auf dem Gerät befinden, zum Versenden ausgewählt werden. Dieser Vorgang kann jedoch auch im Voraus durchgeführt werden, benötigt also nicht zwingend eine Verbindung.

Für die Auswahl der Dateien kann die vom Betriebssystem bereitgestellte Bibliothek genutzt werden (s. Abb. 3.4). Der Benutzer ist durch die Nutzung seines Geräts bestens mit dem Gebrauch dieser Komponente vertraut und benötigt keine weitere Hilfe. Auch eine Mehrfachauswahl ist hierbei möglich. Die ausgewählten Dateien, in diesem Anschauungsbeispiel sind es Bilder, werden in einer Raster-Ansicht dargestellt (s. Abb. 3.5). Auch dieser Schritt ist eine Vorbedingung, um den Swipe für die Dateiübertragung nutzen zu können.



Abbildung 3.4: Auswahl der Dateien

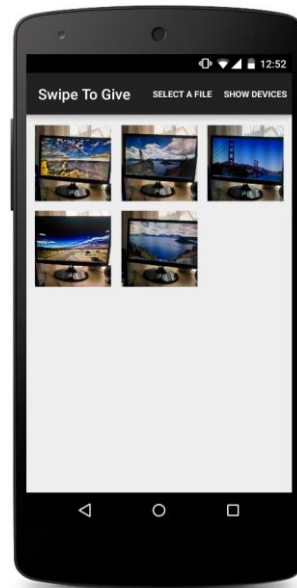


Abbildung 3.5: Darstellung der ausgewählten Dateien

3.4 Geste zum Senden / Transfer

Die dritte und letzte Stufe beschäftigt sich mit der Übertragung der ausgewählten Dateien vom Start- zum Zielgerät. Der Transfer wird durch einen Swipe ausgelöst. Diese Geste wird in der Regel zum Scrollen in Listen verwendet, sobald diese größer als der Bildschirm werden. Um den Swipe für das Senden zu nutzen, dürfen die ausgewählten Dateien nicht die Bildschirmgröße überschreiten. Des Weiteren muss der Swipe gewisse Parameter erfüllen. Ein Parameter ist die Länge des Swipes. Dies soll verhindern, dass die ausgewählten Dateien gesendet werden, sobald der Finger das Display berührt. Ein weiterer Parameter ist die Richtung. Da Dateien vom Gerät versendet werden sollen, sollte eine Dateiübertragung nur gestartet werden, wenn der Swipe nach oben gerichtet ist. Sollte er nach unten gehen, ist dies nicht im Sinne der Blended

Interaction¹. Ein weiterer Aspekt im Sinne der Blended Interaction ist die Ausrichtung des Start- und Zielgeräts. Diese sollten aufeinander zeigen, damit eine Übertragung angestoßen wird. Es macht semantisch wenig Sinn, wenn die Geräte voneinander weg zeigen. Weiterhin sollte dem Benutzer signalisiert werden, dass durch den Swipe die ausgewählten Dateien versendet werden. Hierfür können gestalterische Mittel gewählt werden, wie z.B. ein ausfaden (verblassen). Dieser Vorgang wird in der Abbildung 3.6 verdeutlicht.

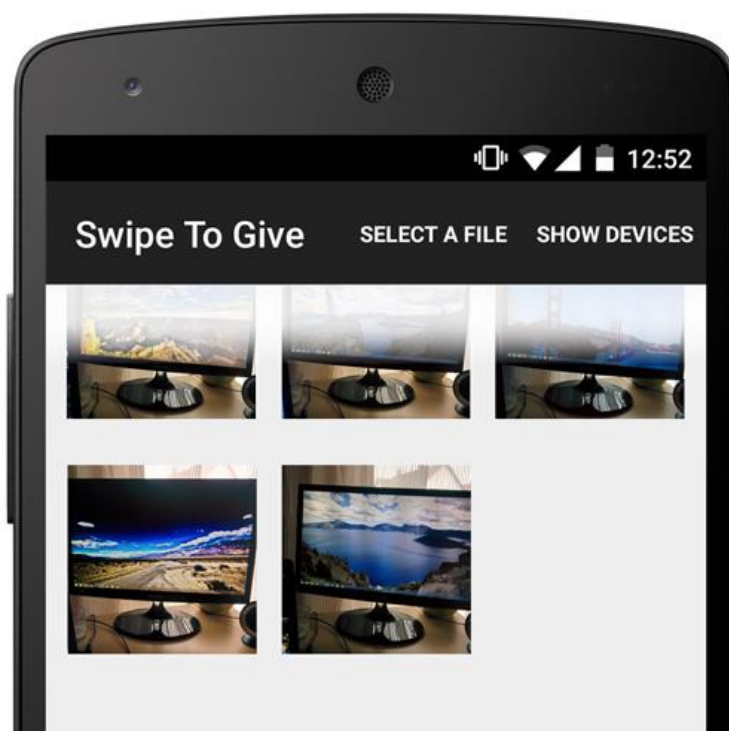


Abbildung 3.6: Verblassen der ausgehenden Dateien

¹ Unter einer Blended Interaction versteht man die Kopplung von digitalen Vorgehensweisen, um die reale Welt mit der digitalen Welt zu vermischen.

3.5 Reaktion des Systems

Die Reaktion des Systems, nach dem Senden der Dateien, wird in Reaktion des Start- (Sender) und Reaktion des Zielgeräts (Empfänger) unterschieden.

3.5.1 Reaktion des Systems beim Sender

Die Reaktion beim Sender ist eine Popupnachricht, welche signalisiert, dass die Dateien übertragen wurden. Des Weiteren wird die Liste der ausgewählten Bilder geleert. Sollte es zu einem Fehler kommen, wird der Fehlertext angezeigt.

3.5.2 Reaktion des Systems beim Empfänger

Die eigentliche Reaktion ist die übertragenen Dateien anzuzeigen (visuelle Rückmeldung). Dies könnte bspw. durch Hineingleiten vom oberen Rand des Geräts sein, wie es auch beim Senden beschrieben wurde (siehe Abbildung 3.6). Der Benutzer erhält durch diese gestalterische Lösung ein besseres Verständnis für den Vorgang. Das Gerät soll jedoch zusätzlich eine akustische Rückmeldung an den Benutzer geben, dass Daten übertragen wurden. In diesem Fall ist die akustische Rückmeldung ein Vibrationsalarm.

3.6 Geeignete Endgeräte

Die Swipe-Interaktion wurde bislang meist mit Smartphones verbildlicht. Dies ist jedoch auch mit anderen Geräten möglich. Es gilt an dieser Stelle allgemein zu klären, zwischen welchen Geräteklassen die Swipe-Geste eingesetzt werden kann. Die Geräte werden, basierend auf ihrer Bildschirmgröße, kategorisiert. In der folgenden Tabelle 3.2 werden die Kategorien dargestellt, welche nach Terrenghi et al. [5] definiert wurden.

Kategorie	Bildschirmgröße	Beispiel
Inch size	$\geq 2,54$ cm	Smartphone
Foot size	$\geq 30,48$ cm	Tablet
Yard size	$\geq 91,44$ cm	Tabletop
Perch size	$\geq 3,05$ m	TV-Bildschirm
Chain size	≥ 20 m	Mehrere Displays

Tabelle 3.2: Kategorisierung von Endgeräten

Die Identifikation der geeigneten Geräte hängt von den Rollen ab, die in der Interaktion eingenommen werden können. Die Endgeräte können in Start- und Zielgeräte aufgeteilt werden. Ein Startgerät ist der Träger der Informationen, von welchem Daten durch ein Swipe auf das Zielgerät übertragen werden. Es kann sich hierbei um mobile und stationäre Endgeräte handeln. Ein stationäres Endgerät zeichnet sich durch die Eigenschaft aus, dass es bei der Nutzung vom Anwender nicht in der Hand gehalten wird. Bei beiden Geräten (Start- und Endgeräte) muss die Klassifizierung unter die Kategorie Inch-, Foot oder Yard size fallen. Für die Startgeräte ist es wichtig, dass auf diesen ein Swipen möglich ist. Umso größer der Bildschirm wird, desto schwieriger wird dies zu realisieren. Bei den Zielgeräten spielt dies keine Rolle. Diese Zuweisung wird in der

angehängten Patterbeschreibung nochmal aufgegriffen und in einer Tabelle verdeutlicht.

3.7 Schlussfolgerung für die Implementierung

Um die Interaktion umzusetzen, müssen zwei Apps entwickelt werden, eine für das Start- und eine für das Zielgerät. Die App ist notwendig, damit das Swipen zur Datenübertragung genutzt werden kann. Außerdem müssen die Auswahl der Dateien und das Herstellen der Verbindung von Start- zu Zielgerät speziell für diese Interaktion angepasst werden. Dies ist nur in einer eigens entwickelten App möglich. Bei der Verbindung der beiden Endgeräte muss ein Informationskanal geöffnet werden, damit Dateien übertragen werden können. Des Weiteren muss mindestens ein Gerät erkennen, welche Geräte zur Interaktion zu Verfügung stehen.

4 Umsetzung der Interaktion

Dieses Kapitel beschreibt Konzepte für das im vorangegangenen Kapitel identifizierte System. Hierbei wird auf den in Kapitel 3.1 dargestellten Ablauf Bezug genommen. Eine Zusammenfassung der Ergebnisse aus diesem Kapitel findet sich ebenfalls im Interaktionspattern, welches dem Anhang beigelegt ist.

4.1 Vorbedingung

Schon zu Beginn der Umsetzung wurde festgestellt, dass die Interaktion nur im Rahmen einer eigenen App genutzt werden kann. Eine Verbindung herzustellen ist mit den im Betriebssystem verankerten Funktionen zwar möglich, jedoch scheitert die Umsetzung bei der Auswahl der Dateien und der Anregung der Datenübertragung per Swipe.

4.2 Bestimmung der Verbindung

Wie in Kapitel zwei beschrieben, stehen mehrere Peer-to-Peer Technologien zur Verfügung. Nachfolgend werden diese auf ihre Vor- und Nachteile, im Bezug zur App, untersucht.

NFC

Bei NFC sind die Vorteile, die einfache Verbindung zwischen den Geräten und eine eindeutige Identifizierung des Zielgeräts. Der Nachteil wiederum ist die sehr langsame Übertragungsgeschwindigkeit. Des Weiteren müssen die beiden zu verbindenden Geräte extrem nah beieinander liegen um eine Verbindung aufzubauen und aufrecht zu erhalten.

Bluetooth

Bluetooth bietet im Vergleich zu NFC eine höhere Übertragungsgeschwindigkeit, jedoch ist die Identifizierung des Zielgeräts nur durch die manuelle Auswahl des Anwenders möglich. Ein weiterer Nachteil ist die Reichweite von ca. 10m, die in Gebäuden nochmal abnehmen kann.

WiFi Direct

WiFi Direct bietet die Übertragungsgeschwindigkeit von WiFi, welches bis zu 300 MB/s ermöglicht. Außerdem ist die Reichweite von WiFi Direct mit 90m deutlich stärker als die zuvor betrachteten Technologien.

Ein Nachteil von WiFi Direct ist die, wie bei Bluetooth, die manuelle Auswahl des Zielgeräts.

Entscheidung

Aufgrund der Erhebung der technischen Vor- und Nachteile, fiel die Auswahl der Übertragungstechnologie für den Prototyp auf WiFi Direct. Die hohe Reichweite und Übertragungsgeschwindigkeit ist für diese Entscheidung ausschlaggebend. Eine manuelle Auswahl des Zielgeräts ist für den Prototyp von Nachteil.

4.3 Geräte sehen / Awareness

Aufgrund der Bestimmung der Verbindung, kann nun genauer auf die einzelnen Aktivitäten eingegangen werden. Um Geräte anzeigen zu können, wird ein Teil des Android Frameworks benötigt. Wie bereits in den technischen Grundlagen erwähnt, nutzt Android NSD um Geräte im Netzwerk zu finden und eine Verbindung herzustellen. Da für den Prototyp WiFi Direct als Übertragungstechnologie gewählt wurde, wird nachfolgend auf die Besonderheiten des Frameworks eingegangen.

Intent Filter

Um WiFi P2P zu verwenden, muss die Applikation auf Broadcast-Aktionen lauschen. Hierfür wird ein Intent Filter mit folgenden Aktionen angelegt:

```
WIFI_P2P_STATE_CHANGED_ACTION  
WIFI_P2P_PEERS_CHANGED_ACTION  
WIFI_P2P_CONNECTION_CHANGED_ACTION  
WIFI_P2P_THIS_DEVICE_CHANGED_ACTION
```

Quellcode 1: Intent Filter

Die Aktion `WIFI_P2P_STATE_CHANGED_ACTION`, siehe Quellcode 1, wird aufgerufen, wenn WiFi an oder ausgeschaltet wird. `WIFI_P2P_PEERS_CHANGED_ACTION` wird aufgerufen, wenn sich die Liste der verfügbaren WiFi Geräte geändert hat. Ändert sich die Verbindung mit anderen Geräten, so wird die Aktion `WIFI_P2P_CONNECTION_CHANGED_ACTION` aufgerufen. Ändert sich die Konfiguration des eigenen Geräts, so lautet die Aktion `WIFI_P2P_THIS_DEVICE_CHANGED_ACTION`. Sobald einer dieser Fälle eintritt, wird der sog. Broadcast Receiver aufgerufen, der das weitere Handling übernimmt.

Suche nach verfügbaren Geräten

Durch die Methode `discoverPeers()` wird der Applikation mitgeteilt, dass nach verfügbaren Geräten gesucht werden soll. Sobald Geräte gefunden werden, wird der `peerListListener` aufgerufen. Dieser muss zuvor überschrieben werden, damit auf die Liste zugegriffen werden kann (siehe Quellcode 2). Sobald neue Geräte gefunden werden, wird die alte Liste geleert und mit den Neuen gefüllt. Daraufhin wird die GUI geupdated.

```

private PeerListListener peerListListener = new PeerListListener() {
    @Override
    public void onPeersAvailable(WifiP2pDeviceList peerList) {

        // Out with the old, in with the new.
        peers.clear();
        peers.addAll(peerList.getDeviceList());

        // If an AdapterView is backed by this data, notify it
        // of the change. For instance, if you have a ListView of
        // available peers, trigger an update.
        ((WifiPeerListAdapter) getListAdapter())
            .notifyDataSetChanged();

        if (peers.size() == 0) {
            Log.d(WiFiDirectActivity.TAG, "No devices found");
            return;
        }
    }
}

```

Quellcode 2: PeerListListener

Der Verbindungsaufbau lässt sich auch über ein Aktivitätsdiagramm verbildlichen (siehe Abbildung 4.1).

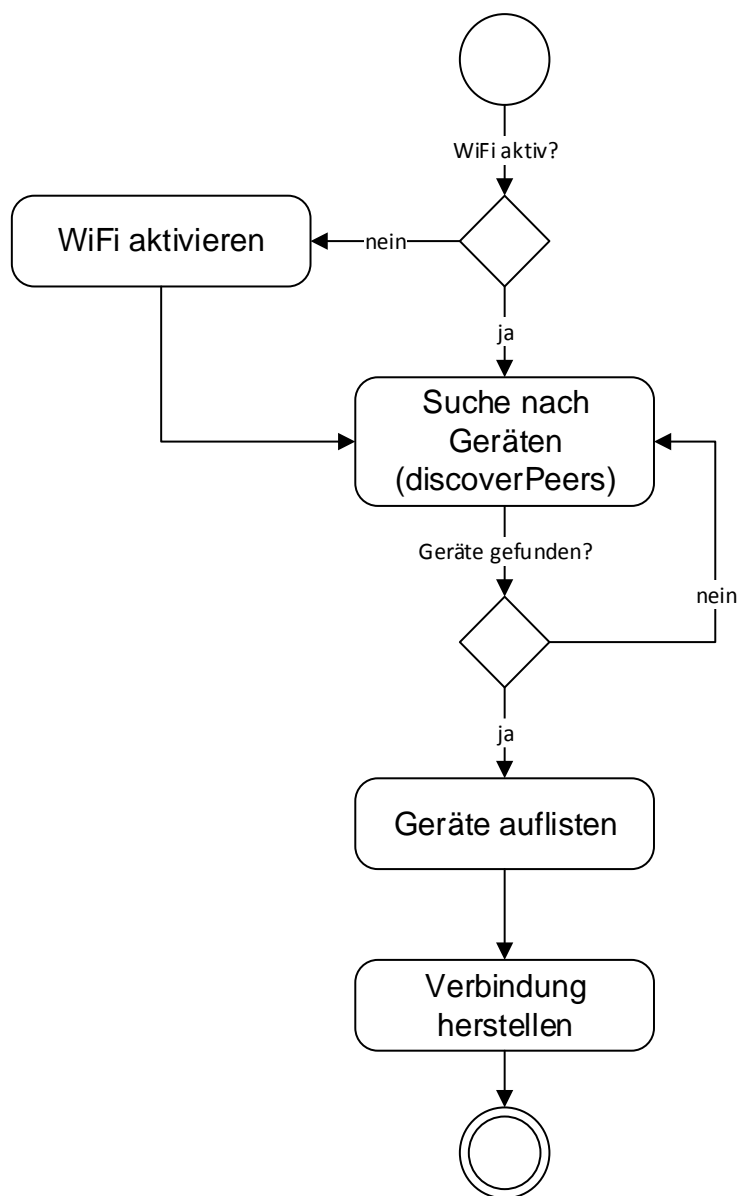


Abbildung 4.1: Aktivitätsdiagramm des Verbindungsaufbaus

4.4 Auswahl der Dateien / Reveal

Um Dateien in Android auswählen zu können, wird das Storage Access Framework (SAF) benötigt. Hierdurch können Nutzer ihre Bibliothek durchsuchen und Dokumente, Bilder, sowie andere Dateien öffnen. Android bietet eine standardisierte und einfach zu bedienende Benutzeroberfläche, auf die alle Apps zugreifen können. Dies hat zur Folge, dass Nutzer sich nur einmal an die Dateiauswahl gewöhnen müssen. Des Weiteren kann über das SAF auch auf die Online-Bibliothek und auf Google Drive zugegriffen werden. [14]

Um auf die Medienbibliothek zuzugreifen, muss eine neue Aktivität, in Android „Intent“ genannt, angelegt werden. Dem Intent müssen nun, wie in Quellcode 3 zu sehen, Parameter zugewiesen werden.

```
Intent buttonLoadPictureIntent = new Intent();
buttonLoadPictureIntent.putExtra(Intent.EXTRA_LOCAL_ONLY, true);
buttonLoadPictureIntent.setType("image/*");
buttonLoadPictureIntent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, true);
buttonLoadPictureIntent.setAction(Intent.ACTION_GET_CONTENT);
startActivityForResult(Intent.createChooser(buttonLoadPictureIntent,
"Select Picture"), RESULT_LOAD_IMAGE);
```

Quellcode 3: Parameterzuweisung für einen Intent

Durch den Parameter „Intent.EXTRA_LOCAL_ONLY,true“, welcher in der zweiten Zeile angegeben ist, wird die Auswahl der Dateien auf die lokal gespeicherten Dateien begrenzt. Dies vereinfacht das Handling der Dateien in diesem Prototyp. Durch das angeben des Typs (setType("image/*")), werden nur Bilder in der Bibliothek angezeigt. Möchte man bspw. alle verfügbaren Dateien ohne Einschränkung auswählen, so muss der Typ wie folgt geändert werden: setType("*/"). Da im Prototyp mehrere Dateien ausgewählt werden sollen, muss der Parameter „Intent.EXTRA_ALLOW_MULTIPLE, true“ angegeben werden. Noch weiß der Intent nicht, was genau er öffnen soll. Dies geschieht durch den Aufruf „Intent.ACTION_GET_CONTENT“ in der fünften Zeile. Hier wird festgelegt, dass die Medienbibliothek geöffnet werden soll. In der letzten

Zeile wird die Aktion letztendlich gestartet und der Nutzer kann Dateien auswählen.

4.5 Geste zum Senden

Wie bereits in Kapitel 3.4 beschrieben, besitzt ein Swipe verschiedene Parameter. In Abbildung 4.2 wird verdeutlicht, wann ein Swipe akzeptiert wird und wann nicht.

In Android werden Gesten vom Betriebssystem erkannt und können weiter verarbeitet werden. Beispielsweise wird erkannt, wann und wo ein Finger das Display berührt (Action Down). Das Gleiche gilt auch für das wieder Anheben des Fingers (Action Up). Durch diese beiden Gesten kann ein Swipe erkannt werden. Für eine Implementierung (siehe Quellcode 4) werden zwei Variablen benötigt. Variable `y1` steht für das Berühren des Displays, Variable `y2` für das wieder loslassen. Nun können die beiden Punkte und deren Entfernung gemessen werden.

```
switch (event.getAction()) {
    // when user first touches the screen we get y coordinate
    case MotionEvent.ACTION_DOWN: {
        y1 = event.getY();
        break;
    }
    case MotionEvent.ACTION_UP: {
        y2 = event.getY();
        float length = Math.abs(y2 - y1);

        if (detectSwipe(length)) return true;
        break;
    }
}
```

Quellcode 4: Swipe-Erkennung

Ein Swipe hat eine Richtung, sowie eine Länge. In diesem Fall muss der Swipe von unten nach oben erfolgen und die Länge muss mindestens 450 Pixel lang sein. Dies hat folgende Abfrage zur Folge:

```
// Geht der Swipe nach oben und ist er lange genug?  
y1 > y2 && length >= SWIPE_MIN_DISTANCE
```

Quellcode 5: Abfrage, ob der Swipe zulässig ist

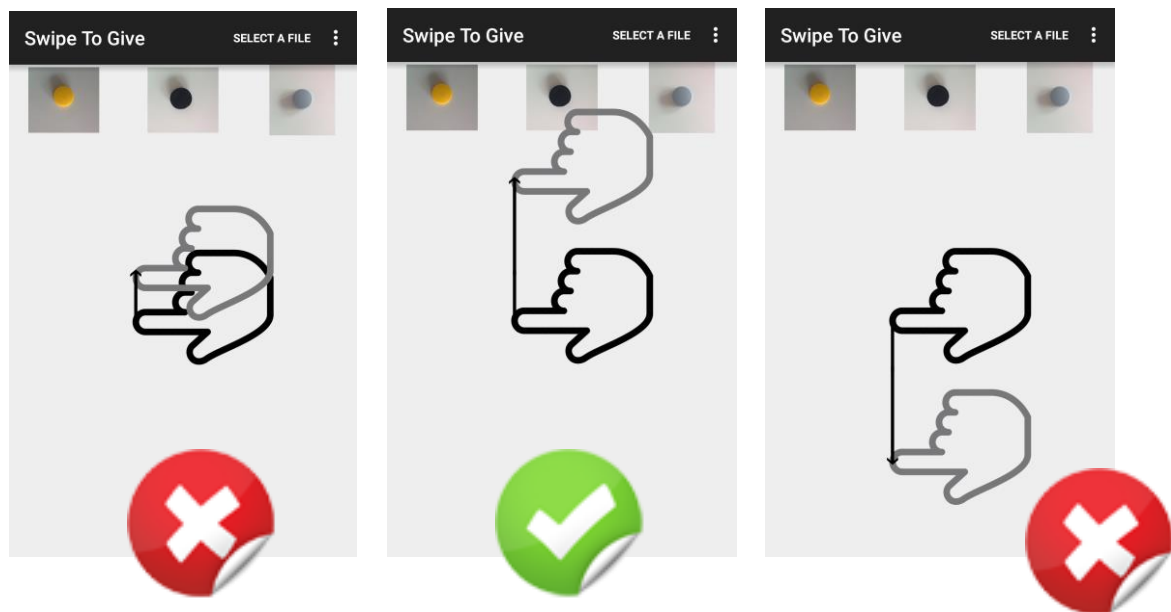


Abbildung 4.2: Swipe-Parameter

4.6 Datenübertragung

Wie in Kapitel 2.2.2 beschrieben, stehen zwei Arten der Datenkommunikation zwischen mobilen Endgeräten zur Verfügung. Zum einen die direkte Kommunikation in einem Adhoc Netzwerk und zum anderen über einen Server.

Die direkte Kommunikation kommt ohne eine Server-Infrastruktur aus und benötigt keine Internetverbindung. Dies ist besonders für mobile Geräte von Vorteil, da eine schnelle und stabile Verbindung nicht zwangsläufig gewährleistet werden kann. Ein Nachteil besteht jedoch im höheren Implementierungsaufwand, wenn für jede Plattform eine Lösung angeboten werden möchte. Hierfür müssten für jede Plattform Einzellösungen geschaffen werden, da Technologien und Frameworks nicht in der Gesamtheit kompatibel sind. Eine Serverarchitektur dagegen muss nur einmal aufgesetzt werden und ist somit schneller implementiert. Nachteilig ist jedoch, wie bereits erwähnt, der zwingende Zugang zum Internet.

Weiterhin handelt es sich bei einem Server um einen Single-Point-Of-Failure. Fällt der Server aus, funktioniert das komplette System nicht mehr [6]. Ein weiterer Vorteil der lokalen Lösung ist die Ausfallsicherheit, die hierbei stärker gegeben ist. Außerdem fallen die Kosten für einen Server weg. In diesem Fall wird für die Übertragung ein Socket verwendet. Dieser kann sowohl online, also auch offline Daten austauschen. Im Folgenden wird die Datenübertragung anhand eines Aktivitätsdiagramms verbildlicht (siehe Abbildung 4.3).

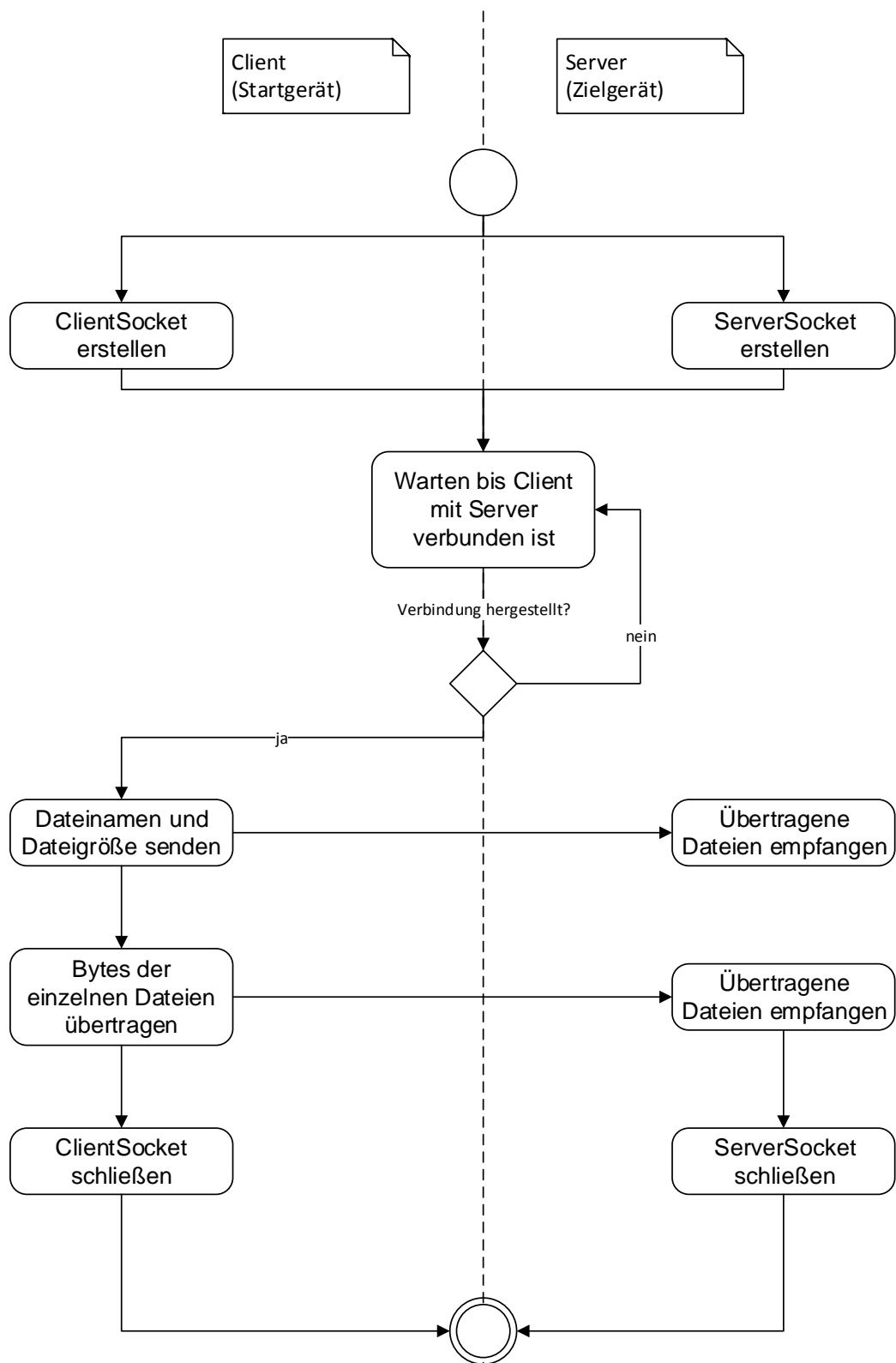


Abbildung 4.3: Aktivitätsdiagramm Dateiübertragung

5 Demonstrator Applikation

Um die Machbarkeit der zuvor konzipierten App zu gewährleisten, wird diese als praktischer Teil dieser Arbeit umgesetzt. Durch die prototypisch umgesetzte Applikation kann der Anwender Bilder, welche auf seinem Endgerät gespeichert sind, auswählen und per Swipe an ein zweites Gerät übertragen. Die Implementierung basiert auf dem Android Betriebssystem, welches als Programmiersprache Java nutzt. Das frei verfügbare Git Repository ist unter folgendem Link erreichbar:

<https://github.com/informatik-mannheim/thesis-swipe-to-give>

Der Code ist frei verfügbar und kann unter folgendem Link erreicht werden:

<https://github.com/informatik-mannheim/thesis-swipe-to-give/tree/master/sources/Code>. Des Weiteren gibt es ein Demo-Video, welches unter folgender Adresse gedownloadet werden kann: <https://github.com/informatik-mannheim/thesis-swipe-to-give/tree/master/sources/Video>.

Nachfolgend werden Screenshots der Applikation dargestellt, um einen Einblick in die Nutzung zu bieten. Hierbei wird das Kapitel in Start- und Zielgerät unterteilt.

5.1 Startgerät

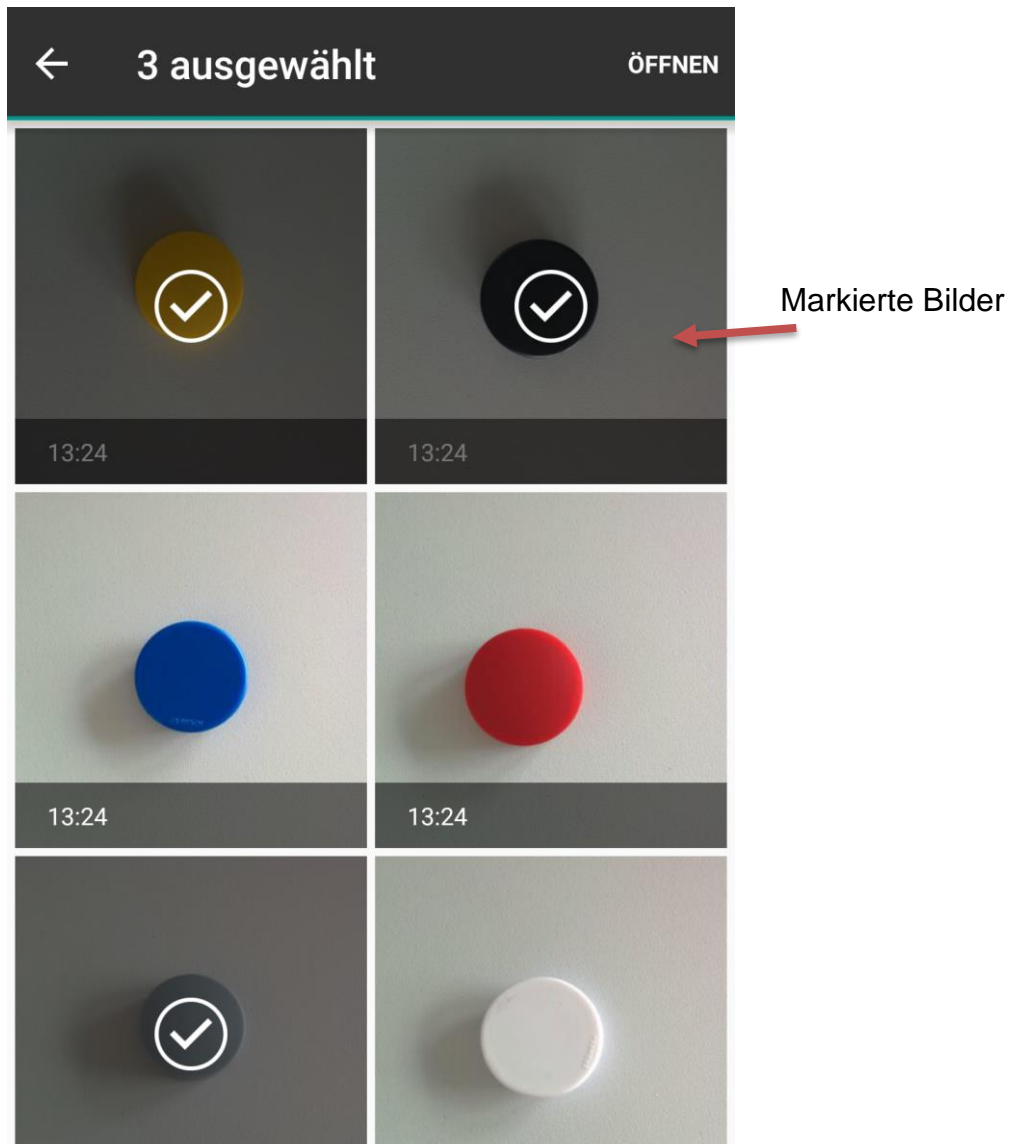


Abbildung 5.1: Auswahl der Dateien

Auf dem Startgerät erfolgt die Auswahl der Bilder, wie in Kapitel 4.4, über den Standard-Android-Dateimanager. Durch das Gedrückthalten einer Datei, wird die Mehrfachauswahl aktiviert und es können, wie in Abbildung 5.1 zu sehen, beliebig viele Dateien ausgewählt werden. Sobald die Dateien durch einen Klick auf ÖFFNEN final ausgewählt wurden, werden sie, wie in Abbildung 5.2, angezeigt.

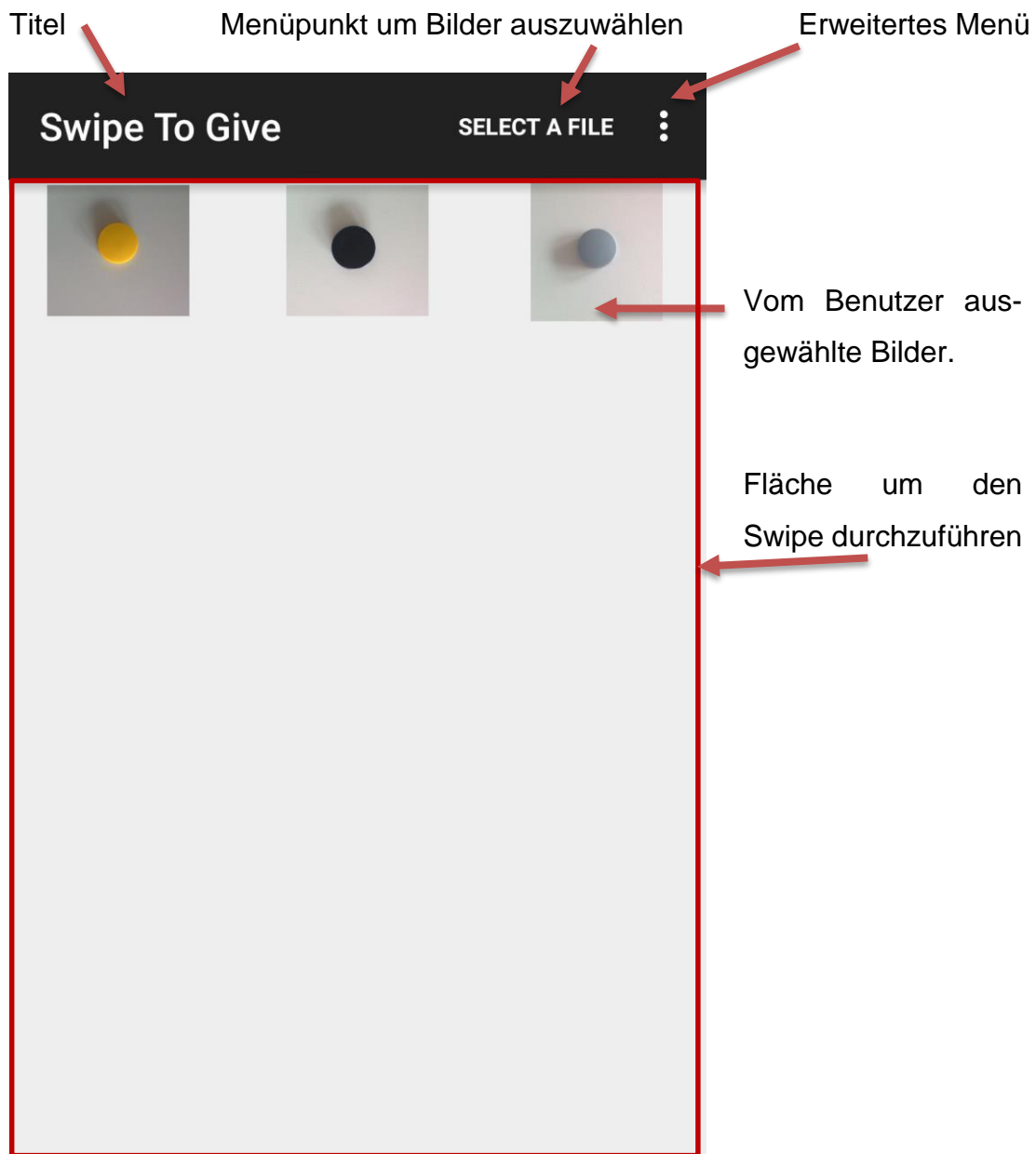


Abbildung 5.2: Darstellung der ausgewählten Dateien

In Abbildung 5.2 ist zu erkennen, dass die zuvor ausgewählten Bilder nun versendet werden können. Dies ist durch ein Swipe von unten nach oben möglich. Es spielt dabei keine Rolle ob die Wischbewegung auf einem Bild stattfindet oder auf einer leeren Fläche. Durch den Menüpunkt oben rechts können die verfügbaren Endgeräte angezeigt und eine Verbindung hergestellt werden. Dies ist im nächsten Bild, Abbildung 5.3, dargestellt.



Abbildung 5.3: Liste der verfügbaren Endgeräte

5.2 Zielgerät

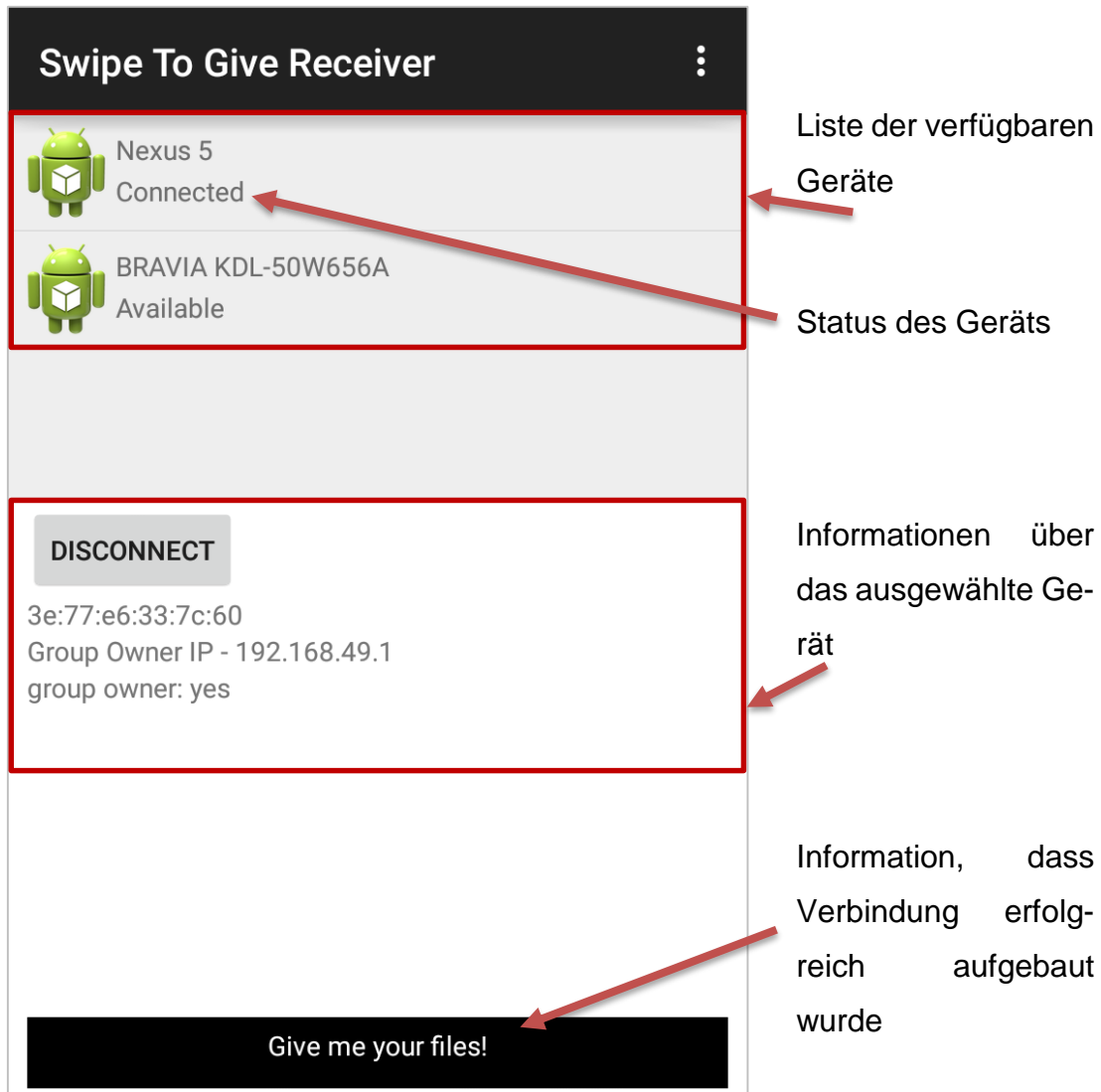
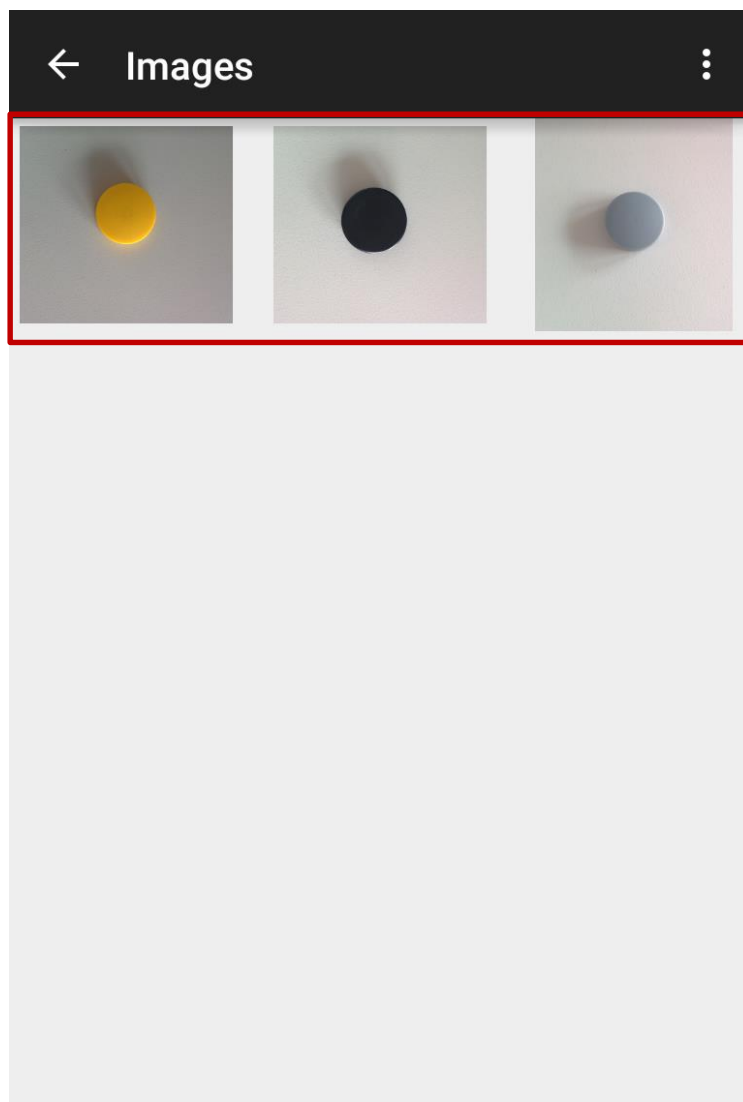


Abbildung 5.4: Liste der verfügbaren Endgeräte

Da das Zielgerät keine Dateien o.ä. auswählen muss, wird nur eine Liste der verfügbaren Endgeräte angezeigt (siehe Abbildung 5.4). Sobald eine Verbindung mit einem Endgerät besteht, können Dateien gesendet werden. Eine erfolgreiche Datenübertragung hat zur Folge, dass die empfangenen Dateien, wie im nächsten Bild (Abbildung 5.5), angezeigt werden und ein Vibrationsalarm ausgelöst wird.



Übertragung war erfolgreich und die empfangenen Bilder werden angezeigt.

Abbildung 5.5: Darstellung der empfangenen Dateien

6 Zusammenfassung und Ausblick

Das zu Beginn dieser Arbeit formulierte Ziel wurde durch die folgenden Teilschritte erreicht: Zu Beginn wurde beschrieben, wie die Interaktion abläuft und welche Kategorien von Endgeräten für die Geste geeignet sind. Außerdem werden die Benutzeraktionen beschrieben. Anschließend wurden Technologien identifiziert, die eine Umsetzung der Geste auf mobilen Endgeräten ermöglicht.

Für die Datenübertragung wurde ein Konzept entwickelt, mit dem Daten durch die Bildung von Ad-Hoc Netzen ausgetauscht werden. Vorher wurden die möglichen Peer-to-Peer Verbindungen untersucht und die Passendste ausgewählt.

Außerdem wurde untersucht, welche Vorbedingungen gelten müssen, damit eine Swipe-Geste funktionieren kann. Die Wichtigste ist, dass die Auswahl der Dateien und das Swipen in einer eigens dafür erstellten App durchgeführt werden muss. Ein Eingriff in das Betriebssystem ist zum aktuellen Zeitpunkt nicht denkbar.

Als praktischer Teil der Arbeit wurde eine Applikation auf der Android-Plattform implementiert mit der sich Endgeräte über WiFi Direct identifizieren und über lokale WLAN Netzwerke oder AD-Hoc Netzwerke Bilddateien austauschen können.

Aufbauend auf den Ergebnissen dieser Arbeit sind einige weiterführende Arbeiten denkbar. Es wäre interessant, auch eine Applikation auf der iOS Plattform zu implementieren und dort zu prüfen, ob auch iBeacons zur Identifizierung der Partnergeräte genutzt werden können. Des Weiteren wäre es von Vorteil, ein Usability-Test anzustellen. In diesem müsste untersucht werden, ab welcher Länge der Interaktion ein Swipe erkannt wird, damit der Anwender ein angenehmes Gefühl bei der Nutzung besitzt.

Aufbauend auf der Implementierung der Swipe-To-Give Geste, kann ebenfalls die Geste Swipe-To-Take implementiert werden. Hierfür sind kaum Änderungen nötig. Lediglich der Swipe muss in die andere Richtung erfolgen und daraufhin eine Übertragung anfordern.

Zu den bereits festgestellten Endgeräten, könnten bald noch Fernseher hinzukommen. Google möchte in Zukunft Android-TVs [9] anbieten. Somit könnte eine Datenübertragung zum Fernseher möglich werden. Auch aus diesem Gedanken können Arbeiten entstehen.

Abschließend ist durch diese Arbeit nicht festzustellen, ob die App den gewünschten interaktiven Nutzen mit sich bringt. Eine Abwandlung des Betriebssystems wäre wohl eine bessere Lösung, jedoch ein enormer Aufwand, der erst durch eine Machbarkeitsstudie untersucht werden muss. Sollte diese positiv ausfallen, kann auch hierzu eine eigenständige Arbeit verfasst werden.

Abkürzungsverzeichnis

GPS Global Positioning System

URL Uniform Resource Locator

App Mobile Applikation

WLAN Wireless Local Area Network

IEEE Institute of Electrical and Electronics Engineers

NFC Near Field Communication

RFID radio-frequency identification

HSPA+ High Speed Packet Access

LTE Long Term Evolution

TCP/IP Transmission Control Protocol / Internet Protocol

HTTP Hypertext Transfer Protocol

CRUD Create, Read, Update, Delete

NSD Android Network Service Discovery

P2P Peer-to-Peer

iOS iPhone Operating System

GUI Graphical user interface (Grafische Benutzeroberfläche)

SAF Storage Access Framework

Tabellenverzeichnis

Tabelle 3.1: Edward T. Hall's Definition des persönlichen Bereichs [12]	21
Tabelle 3.2: Kategorisierung von Endgeräten.....	26

Abbildungsverzeichnis

Abbildung 2.1: Mobile und stationary (fest-angebrachte) Geräte [15].....	8
Abbildung 2.2: FotoSwipe Demo [17]	10
Abbildung 2.3: Piconetz / Bluetooth-Topologie [2]	12
Abbildung 2.4: Client-Server-Modell	16
Abbildung 3.1: Aktivitätsdiagramm einer gestenbasierten Interaktion.....	19
Abbildung 3.2: Aktivitätsdiagramm "Swipe To Give"	20
Abbildung 3.3: Bump zweier Geräte zur eindeutigen Identifizierung [6].....	22
Abbildung 3.4: Auswahl der Dateien.....	23
Abbildung 3.5: Darstellung der ausgewählten Dateien	23
Abbildung 3.6: Verlassen der ausgehenden Dateien.....	24
Abbildung 4.1: Aktivitätsdiagramm des Verbindungsaufbaus	32
Abbildung 4.2: Swipe-Parameter	35
Abbildung 4.3: Aktivitätsdiagramm Dateiübertragung	37
Abbildung 5.1: Auswahl der Dateien.....	39
Abbildung 5.2: Darstellung der ausgewählten Dateien	40
Abbildung 5.3: Liste der verfügbaren Endgeräte	41
Abbildung 5.4: Liste der verfügbaren Endgeräte	42
Abbildung 5.5: Darstellung der empfangenen Dateien	43

Literaturverzeichnis

- [1] Google, „<https://www.thinkwithgoogle.com>,“ 08 2012. [Online]. Available: https://ssl.gstatic.com/think/docs/the-new-multi-screen-world-study_research-studies.pdf. [Zugriff am 23 03 2015].
- [2] C. Baun, Computernetze kompakt (IT kompakt), Heidelberg: Springer Berlin, 2012.
- [3] P. Schnabel, Kommunikationstechnik-Fibel, Ludwigsburg: Books on Demand, 2003.
- [4] „www.appato.com,“ [Online]. Available: <http://www.appato.com/s-s-royal-ltd/push2send/>. [Zugriff am 10 04 2015].
- [5] L. Terrenghi, A. Quigley und A. Dix, A taxonomy for and analysis of multi-person-display ecosystems, Springer-Verlag London, 2009.
- [6] B. Grab, *Implementierung einer gestenbasierten Interaktion zum Datenaustausch zwischen mobilen Endgeräten*, Mannheim, 2015.
- [7] S. Sokolov, „prmac.com,“ 12 05 2012. [Online]. Available: <http://prmac.com/release-id-43191.htm>. [Zugriff am 29 04 2015].
- [8] „hoccer.com,“ [Online]. Available: <http://hoccer.com/de/>. [Zugriff am 29 04 2015].
- [9] „[android.com](http://www.android.com),“ [Online]. Available: <http://www.android.com/tv/>. [Zugriff am 07 05 2015].
- [10] L. Cheng, „Service Advertisement and Discovery in Mobile Ad hoc,“ *Laboratory Of Networking Group*, 2002.
- [11] P. Schnabel, Netzwerktechnik-Fibel: Grundlagen, Übertragungstechnik und Protokolle, Anwendungen und Dienste, Sicherheit, Ludwigsburg: Books on Demand, 2004.

- [12] E. T. Hall, *The Hidden Dimension*, The Bodley Head Ltd., 1969.
- [13] N. Marquardt, T. Ballendat, S. Boring, S. Greenberg und K. Hinckley, „ITS,“ in *Gradual Engagement: Facilitating Information Exchange*, Cambridge, Massachusetts, USA, 2012.
- [14] „developer.android.com,“ [Online]. Available: <https://developer.android.com/guide/topics/providers/document-provider.html>. [Zugriff am 18 05 2015].
- [15] S. e. a. Greenberg, „Proxemic interactions: the new ubicomp?,“ interactions 18.1, 2011.
- [16] U. Laufs und C. Ruff., *Aufbau von Multi-Touch-Systemen*, Berlin Heidelberg: Springer, 2013.
- [17] „fotoswipe.com,“ [Online]. Available: <http://fotoswipe.com>. [Zugriff am 05 06 2015].
- [18] D. Schoder und K. Fischbach, *Peer to Peer*, Springer Berlin Heidelberg, 2002.
- [19] V. Burjan, „Eine Sammlung von Blended Interaction Patterns,“ Mannheim, 2014.

Anhang 1 - Interaktionspattern

Das Interaktionspattern liefert eine Zusammenfassung der in dieser Arbeit ermittelten Konzepten und Lösungen.

Was

Allgemeine Beschreibung

Der Benutzer hat ein Gerät A in der Hand oder vor sich stehen. Durch eine Wischbewegung (Swipe) von unten nach oben, werden Daten auf ein zweites Gerät, Gerät B, übertragen. Die Übertragung hängt mit der Ausrichtung der Geräte zusammen.

Kategorie

- ☒ Give
- ☐ Take
- ☐ Exchange
- ☐ Cut
- ☐ Connect
- ☐ Disconnect
- ☐ Sonstige

Tags

wischen, swipe, Finger, von unten nach oben, Daten senden, übertragen, swipe to give

Wie

Aktion des Benutzers

Der Benutzer hält Gerät A (Startgerät) in der Hand oder hat das Gerät vor sich stehen (z.B. in Form eines Tabletops).

Der Benutzer führt einen Swipe mit einem Finger vom unteren zum oberen Bildschirmrand aus. Die Bewegung erfolgt in Richtung von Gerät B, welches die Daten empfängt. Der Swipe hat eine Länge, sowie eine Richtung. Dies ist zwingend erforderlich, um eine Datenübertragung anzuregen.

Reaktion der Geräte

Ist der Datensatz erfolgreich versendet worden, so reagiert Gerät A mit einer visuellen Nachricht an den Benutzer. Er bekommt eine kleine Nachricht angezeigt, dass die Übertragung erfolgreich verlief. Bereits während des Swipes soll der Benutzer verdeutlicht bekommen, dass die ausgewählten Dateien übertragen werden. Hierfür verblassen die ausgewählten Dateien nach oben hin (siehe nachfolgendes Bild).

Gerät B empfängt den Datensatz und zeigt diesen auf dem Bildschirm an. Außerdem wird ein kurzer Vibrationsalarm ausgelöst. Auch hier könnten die Dateien vom oberen Rand hineingleiten.



Hinweise zur Gestaltung der Interaktion

Es ist gefordert, entsprechend der Geräte zu berücksichtigen, mit wie viel Fingern die Swipe-Geste ausgeführt werden soll. In diesem Fall wird der Swipe mit einem Finger ausgeführt und findet in einer eigens dafür gestalteten App statt. Um die Interaktion, den Swipe, durchführen zu können, wird eine Verbindung zu einem Gerät benötigt.

Grafische Darstellung der Interaktion

Daten übertragen (Startgerät zu Zielgerät)



Übertragung der ausgewählten Dateien per Swipe (li: Start, re: Ziel)

Daten empfangen (auf dem Zielgerät)



Visuelle Darstellung nach erfolgreicher Übertragung (li: Start, re: Ziel)

Wann

Geeignete Verwendung/en (Kontext)

Datenaustausch von Bildern, Videos, Dateien (z.B. PDFs), Social Network IDs

Interaktionsnutzer

- ☒ Einzelnutzung
- ☒ kollaborative Nutzung

Berücksichtigung der Fähigkeiten und Kenntnisse des Anwenders über

- ☒ physikalische Gegebenheiten
- ☒ das eigene Körperbewusstsein
- ☒ Interaktion mit der Umwelt
- ☒ die soziale Interaktion

Displaytypen

(Beziehung zwischen 2 verschiedene Gerätetypen)

nach von	Klein ¹	Mittel ²	Groß ³	Sehr groß ⁴	Riesig ⁵
Klein ¹	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mittel ²	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Groß ³	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sehr groß ⁴	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Riesig ⁵	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Warum

☒ Pattern Kandidat

☒ realisierbar

☐ futuristisch

☐ Bewährtes Interaction Pattern

¹ Inch size $\geq 2,54$ cm
Beispiel: Smartphone, Smartwatch

² Foot size $\geq 30,48$ cm
Beispiel: Tablets

³ Yard size $\geq 91,44$ cm
Beispiel: Tabletop

⁴ Perch size $\geq 3,05$ m
Beispiel: TV-Screen

⁵ Chain size ≥ 20 m
Beispiel: Mehrere Displays

Analoge Patterns

Swipe To Take: gleiche Interaktion, nur wird bei Swipe To Take ein Datensatz von einem entfernten Gerät angefordert

State of the Art / Gebrauchshistorie

1. <https://www.youtube.com/watch?v=GDdPN6mVLPM>
von Minute 0:56 bis 0:59

Mit der Swipe-Geste auf vom Gerät, wird das konfigurierte Auto auf die Leinwand gegenüber den Kunden projiziert.

2. <http://vimeo.com/53606494>
von Minute 2:14 bis 2:25

Mit der Swipe-Geste auf dem Tablet simuliert der Verkäufer (rotes Hemd) dem Kunden (blaues Hemd) wie diverse Krawatten an ihm aussehen.

3. <http://www.microsoft.com/office/vision/>
von Minute 3:41 bis 3:45

4. <https://www.youtube.com/watch?v=ho00x7ZvDLw>
von Minute 0:35 bis 0:37

5. https://www.youtube.com/watch?v=o_hKFOQollg
von Minute 1:04 bis 1:09 und 1:30 bis 1:45 in kollaborativer Nutzung

6. <http://vimeo.com/110928116>
von Minute 1:09 bis 1:15

7. <https://www.youtube.com/watch?v=yw564p8oF1M>
von Minute 0:14 bis 0:20

Chromecast: Technologie mit der Daten z.B. vom Smartphone auf den Fernseher übertragen werden.

8. Gradual Engagement: Facilitating Information Exchange between Digital Devices as a Function of Proximity [13]
9. Hoccer [8]
10. Push2Send [7]

11. FotoSwipe [17]

Metapher

Jemandem etwas aushändigen / übergeben

Technisches

Benötigte Technologie/n

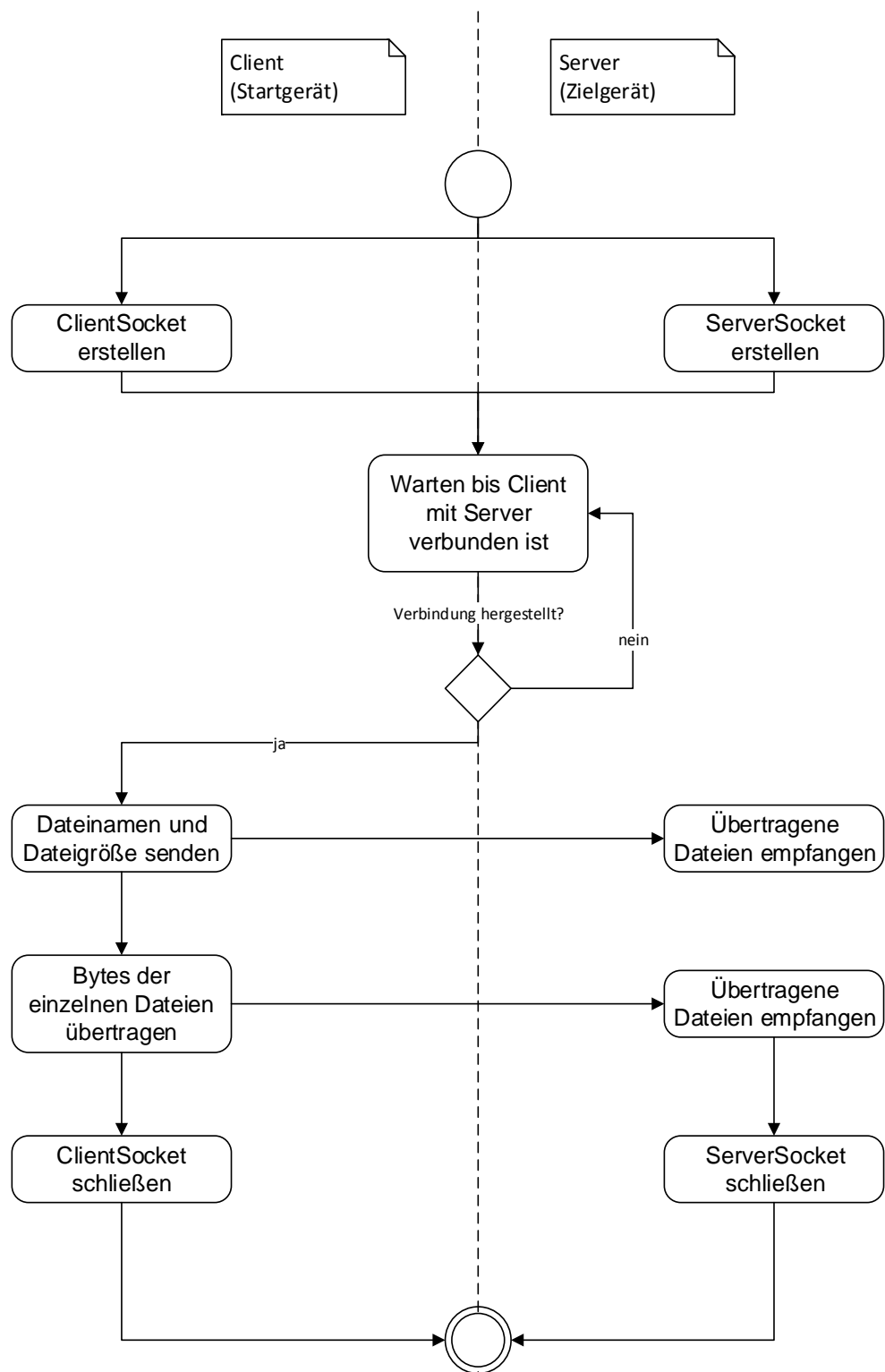
- ☒ Bluetooth
- ☒ Wi-Fi
- ☒ NFC
- ☒ GPRS/3G/4G (LTE)
- ☐ Infrarot
- ☐ GPS

Eine der gekennzeichneten Technologien muss vorhanden sein, damit Daten übertragen werden können.

Prototyp / Lösungsansatz / Code-Snippet / UML-Diagramm

```
switch (event.getAction()) {  
    case MotionEvent.ACTION_DOWN: {  
        y1 = event.getY();  
        break;  
    }  
    case MotionEvent.ACTION_UP: {  
        y2 = event.getY();  
        float length = Math.abs(y2 - y1);  
        if (detectSwipe(length)) return true;  
        break;  
    }  
}
```

Code-Beispiel für den Swipe, um die Datenübertragung anzuregen



UML-Diagramm zum Datenaustausch

Sonstiges

Autoren

Alexander Hahn, Valentina Burjan // Hochschule Mannheim

