

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

Alexander Hahn

Bachelor-Thesis

Studiengang Informatik

Fakultät für Informatik

Hochschule Mannheim

12.05.2015

Betreuer:

Prof. Kirstin Kohler, Hochschule Mannheim

Horst Schneider, B.Sc, 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, 12.05.2015

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 und stationäre Endgeräte der neueren Generation mit dem Betriebssystem „Android“. Der Quellcode wird als Open Source Projekt zur Verfügung gestellt.



---

# Inhalt

---

<b>1</b>	<b>Einleitung.....</b>	<b>8</b>
1.1	Motivation .....	8
1.2	Zielsetzung .....	8
1.3	Inhalt und Struktur .....	9
<b>2</b>	<b>Stand der Technik.....</b>	<b>10</b>
2.1	Verwandte Arbeiten und Applikationen.....	10
2.2	Technische Grundlagen .....	13
2.2.1	Technologien zur Datenübertragung .....	13
2.2.2	Datenkommunikation zwischen mobilen Endgeräten .....	17
2.2.3	Sensoren .....	19
<b>3</b>	<b>Konzept der „Swipe To Give“-Geste .....</b>	<b>20</b>
3.1	Ablauf der Interaktion .....	20
3.2	Aktionen des Benutzers.....	21
3.2.1	Geste zum Verbinden .....	21
3.2.2	Auswahl der Dateien.....	21
3.2.3	Geste zum Senden .....	22
3.3	Reaktion des Systems .....	23
3.3.1	Reaktion des Systems beim Sender .....	23
3.3.2	Reaktion des Systems beim Empfänger .....	23
3.4	Geeignete Endgeräte .....	23

3.5	Umsetzung der Interaktion.....	25
<b>4</b>	<b>Umsetzung der Interaktion .....</b>	<b>26</b>
4.1	Vorbedingungen .....	26
4.2	Ablauf .....	27
4.3	Bestimmung Peer-to-Peer Verbindung.....	29
4.4	Grundlagen zu Android Frameworks .....	30
4.4.1	Network Service Discovery .....	30
4.5	Datenübertragung.....	33
<b>5</b>	<b>Demonstrator Applikation .....</b>	<b>35</b>
5.1	Quellgerät.....	36
5.2	Zielgerät.....	39
<b>6</b>	<b>Interaktionspattern.....</b>	<b>41</b>
6.1	Was .....	41
6.2	Wie .....	42
6.3	Wann .....	43
6.4	Warum .....	44
6.5	Technisches .....	46
6.6	Sonstiges.....	46
<b>7</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>47</b>
	<b>Abkürzungsverzeichnis .....</b>	<b>50</b>
	<b>Tabellenverzeichnis.....</b>	<b>51</b>

<b>Abbildungsverzeichnis .....</b>	<b>52</b>
------------------------------------	-----------

<b>Literaturverzeichnis.....</b>	<b>53</b>
----------------------------------	-----------

---

# 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 sollen auch andere Dateien übertragen werden können, 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 2 werden verwandte Arbeiten und Applikationen zum Thema, sowie technische Grundlagen beschrieben. Kapitel drei analysiert und beschreibt die Swipe-Interaktion. Kapitel vier beschreibt einleitend Vorbedingungen, die für eine Umsetzung zu erfüllen sind. Anschließend werden die Grundlagen zu relevanten Android Frameworks beschrieben, auf deren Basis Konzepte zur Geräteidentifizierung und zum Datenaustausch und verglichen werden. In Kapitel fünf wird die entwickelte Demonstrator Applikation vorgestellt. Kapitel sechs ist die Patterbeschreibung, also eine Zusammenfassung über die erarbeiteten Ergebnisse aus Kapitel drei und vier.

Abschließend bietet Kapitel sieben 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

---

Im folgenden Kapitel werden verwandte Arbeiten und Applikationen zum Thema, sowie technisches Grundlagenwissen, dass zur Umsetzung der Interaktion benötigt wird, beschrieben.

### 2.1 Verwandte Arbeiten und Applikationen

Anhand der Studie „**Gradual Engagement: Facilitating Information Exchange between Digital Devices as a Function of Proximity**“ [10], 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 [6]

Zu Beginn gab es zwei 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 (siehe Abbildung 2.2). Die auf dem Gerät befindlichen Bilder werden direkt angezeigt, sobald die Person sich dem TV nähert (siehe Abbildung 2.3).



Abbildung 2.2: Visuelle Benachrichtigung, dass die Geräte kommunizieren können [6]

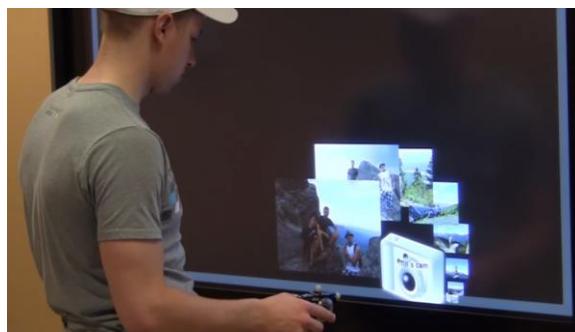


Abbildung 2.3: Darstellung der Fotos auf dem Gerät [6]

**Hoccer** [12] 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** [11] 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. [7] Die App wird nicht im App-Store von Apple angeboten, kann jedoch auf eigenes Risiko unter folgender URL gedownloadet werden: <http://www.appato.com/s-s-royal-ltd/push2send/>

## 2.2 Technische Grundlagen

### 2.2.1 Technologien zur Datenübertragung

#### WLAN

Ein Wireless Local Area Network (WLAN) ist ein lokales Funknetz, das auf der Normenfamilie IEEE 802.11 basiert und sich über eine Reichweite von 30 - 100 Meter aufspannen kann. Ein solches Netzwerk kann entweder im Infrastruktur-Modus oder im Ad-hoc-Modus betrieben werden. Der Infrastruktur-Modus ist topologisch wie ein Stern (Siehe Abbildung 2.4) aufgebaut. Endgeräte melden sich an einer zentralen Basisstation an und können danach über diese miteinander kommunizieren.

Im Ad-hoc-Modus bauen Endgeräte eine direkte Verbindung miteinander, ohne den Umweg über eine gesonderte Basisstation, auf. Jedes Endgerät kann mehrere Verbindungen zu anderen Geräten unterhalten, womit ein vermaschtes Netzwerk (Siehe Abbildung 2.5) gebildet wird. [2]

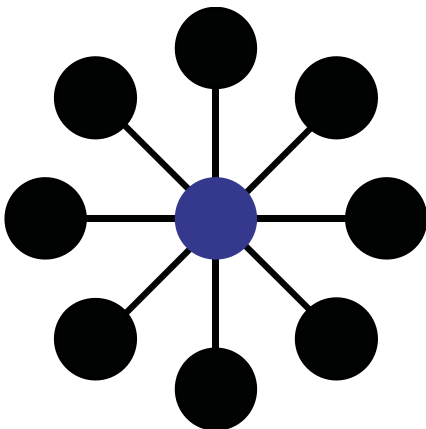


Abbildung 2.5: Sterntopologie

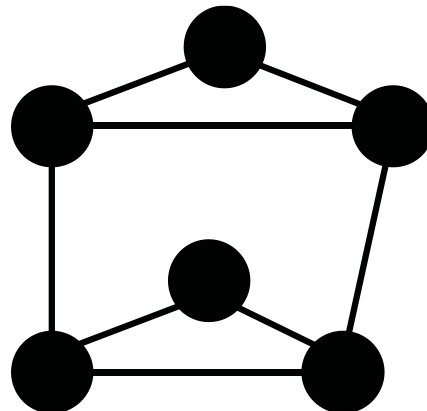


Abbildung 2.4: Maschentopologie

Damit Endgeräte verfügbare WLAN Netze entdecken können, senden Basisstationen oder Ad-hoc Geräte in Intervallen Beacons mit Informationen über

das Netzwerk an alle Endgeräte im Empfangsbereich. Sobald Geräte in Reichweite eines Netzwerkes sind, können sie sich direkt verbinden. Sollte das Netzwerk jedoch durch ein Passwort geschützt sein, muss dieses erst eingegeben und überprüft werden. Es existieren verschiedene WLAN-Standards, die sich vor allem durch ihre Datentransferrate unterscheiden.

Der weit verbreitete Standard 802.11n bietet im Infrastruktur Modus Übertragungsraten von brutto 100-120 Mbit/s. [2]

### **WiFi Direct**

WiFi Direct unterstützt den direkten Austausch von Daten zwischen Geräten, ohne sich vorher, wie bei den aktuellen WLAN-Standards, in ein 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 Wi-Fi 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. Wi-Fi Direct soll zu den bisherigen WLAN-Standards kompatibel sein und man kann Wi-Fi 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.

## 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 2.11).

Ein Pico Netz besteht aus maximal 255 Teilnehmern, wovon maximal acht aktiv sein dürfen. Einer der aktiven Teilnehmer nimmt dabei die Rolle des Masters ein, während die anderen sieben Slaves sind. Die restlichen 247 Teilnehmer sind passiv, können jedoch jederzeit vom Master aktiviert werden. Der Master steuert die Kommunikation und den Datenverkehr zwischen den Teilnehmern des Netzes. [2]

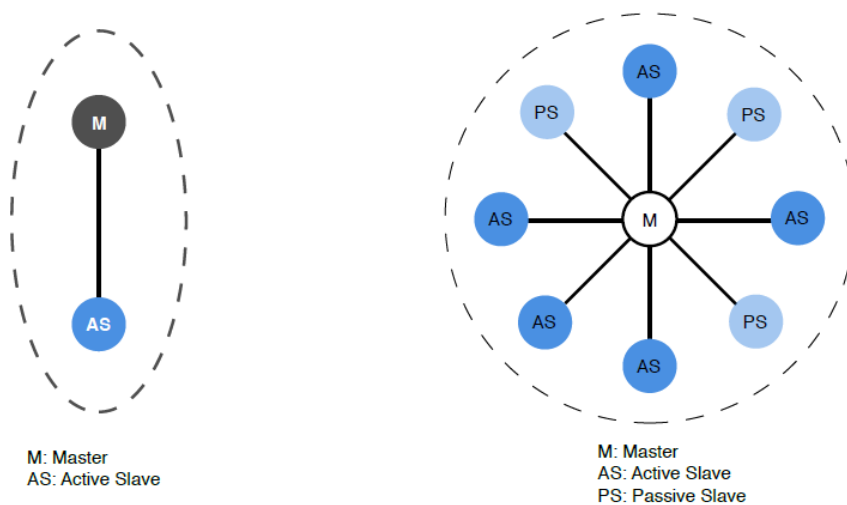


Abbildung 2.6: Piconetz / Bluetooth-Topologie, [2]

Bevor Daten übertragen werden können, müssen sich Geräte erst über den Vorgang des Pairings kennen lernen. Bei diesem Vorgang müssen die Benutzer jeweils einen gemeinsamen Code durch Tastendruck auf ihrem Gerät bestätigen. Nachdem Geräte einmal gepaired wurden, ist ein weiteres Pairing

nicht mehr nötig, da sie sich bereits kennen. Seit Version 3.0 beherrscht Bluetooth den High Speed Modus, welcher eine Kombination aus Bluetooth und WLAN 802.11g ist. Über die Bluetooth Verbindung mit 3 Mbit/s werden dabei nur noch Steuerdaten sowie Sitzungsschlüssel übertragen. Für die Nutzdaten wird eine Ad-hoc Verbindung über WLAN aufgebaut, die eine Datentransfer-rate von ca. 24 Mbit/s erreicht. [2]

### **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 gewährleisten 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. [4]

### **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. [4]

### **2.2.2 Datenkommunikation zwischen mobilen Endgeräten**

Über die Technologien, die im vorherigen Abschnitt beschrieben wurden, kann abhängig von der Technologie, entweder direkt zwischen Geräten im Netzwerk oder über einen Server kommuniziert werden. Folgend wird beschrieben, mit welchen Protokollen die Kommunikation in beiden Fällen umgesetzt werden kann.

#### **Kommunikation in lokalen Netzwerken**

Sollen mobile Applikationen über ein lokales Netzwerk miteinander kommunizieren, kann dies durch die Nutzung von Netzwerkdiensten realisiert werden. Ein Netzwerkdienst ist ein Informationsobjekt, das von einer Person, einem Programm, oder von einem anderen Dienst genutzt werden kann. Jeder Dienst stellt eine Funktionalität zur Verfügung, dabei kann es sich z.B. um Anwendungsdienste oder Kommunikationsdienste handeln [14]. Eine wichtige Komponente zur Nutzung von Diensten ist service discovery, über welches angebotene Dienste in einem Netzwerk gefunden werden können. Für service discovery existieren verschiedene Protokolle.

Apple nutzt in seinen Betriebssystemen Bonjour, Android nutzt Network Service Discovery. Beide Protokolle ermöglichen das automatische Entdecken von Geräten und Diensten in lokalen Netzwerken.

#### **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**

Anstatt direkt in einem lokalen Netzwerk zu kommunizieren, können Applikationen auch über einen Server Daten austauschen. Das Standardmodell für diese Kommunikation ist das Client-Server-Modell (Siehe Abbildung 2.7). In diesem Modell rufen Clients bestimmte Funktionalitäten bzw. Dienstleistungen, die ein Server zur Verfügung stellt, über ein Netzwerk hinweg auf.

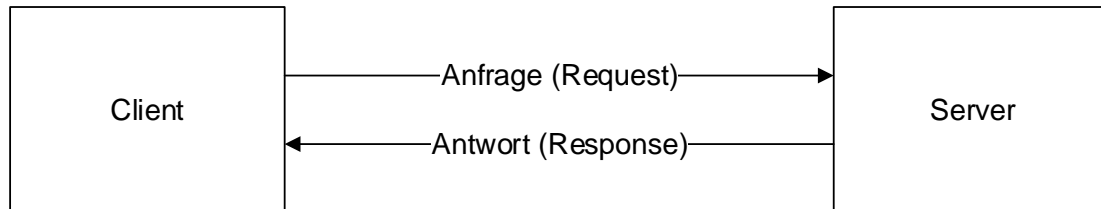


Abbildung 2.7: Client-Server-Modell

### **HTTP Operationen**

Clients stehen HTTP Operationen zur Verfügung, mit denen die Kommunikation mit einem Server erfolgen kann. Die wichtigsten dieser Operationen sind die sogenannten CRUD Operationen: create, read, update and delete. Die entsprechenden HTTP Methoden, mit denen diese Operationen durchgeführt werden können, sind:

- POST um Ressourcen auf dem Server anzulegen.
- GET um Ressourcen aufzurufen.

- PUT um eine Resource zu aktualisieren
- DELETE um eine Resource zu löschen

Endgeräte tauschen über diese Methoden Daten aus, indem eines der Geräte eine Ressource auf dem Server anlegt, während das andere Gerät die entsprechende Ressource aufruft. Der Datenaustausch über diese Methoden hat zur Folge, dass Clients anhaltend überprüfen müssen, ob die Daten bereits auf dem Server verfügbar sind. Für mobile Endgeräte ist dieser Vorgang aber nachteilig, da die ständigen Anfragen an den Server kostbare Batterielaufzeit verschwenden [5]. Es existiert jedoch eine alternative Möglichkeit, mit der Client und Server kommunizieren können ohne dieses Problem, in der Form von Sockets.

### **2.2.3 Sensoren**

Mobile Endgeräte sind mit einer Reihe von Sensoren ausgestattet, mit denen es möglich ist, unentwegt Daten zu sammeln. Applikationen können diese Daten auswerten, um zu sehen, hören und fühlen, was mit dem Gerät und in seiner Umgebung geschieht.

Die vorhandenen Sensoren in einem Gerät variieren zum Teil sehr stark. Android und Apple-Geräte haben nahezu die gleichen Sensoren verbaut, wohingegen Endgeräte mit Windows nur einen kleinen Teil abdecken.

### 3 Konzept der „Swipe To Give“-Geste

#### 3.1 Ablauf der Interaktion

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

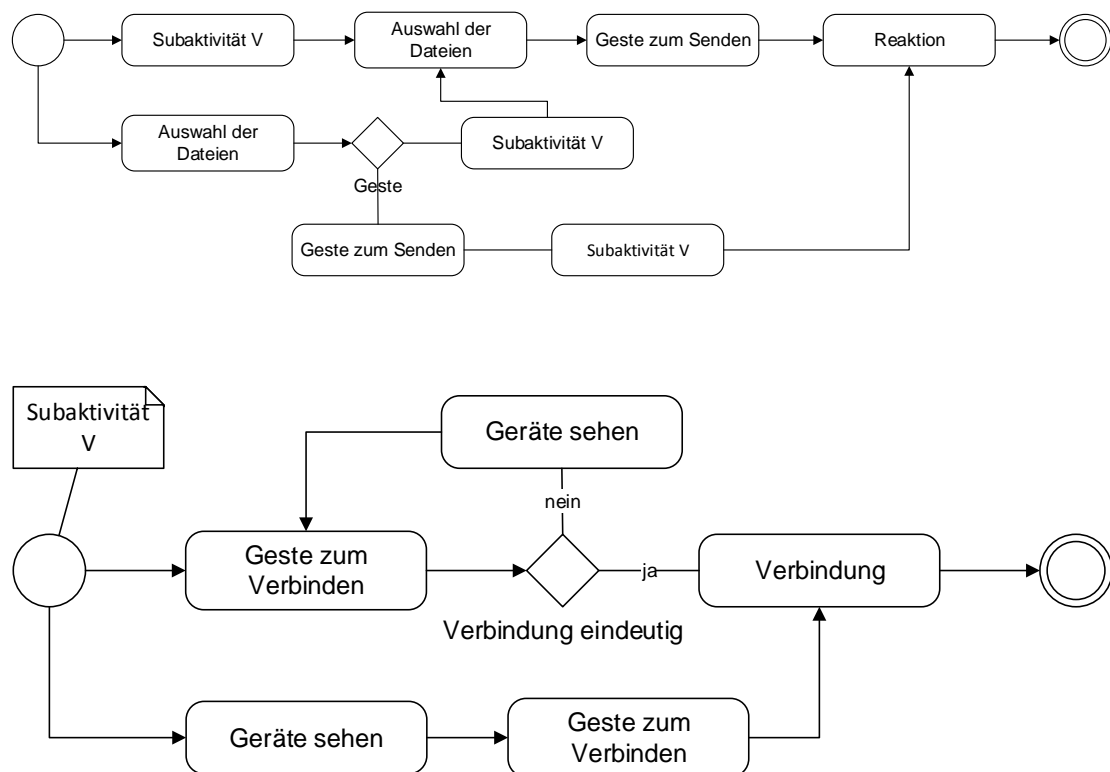


Abbildung 3.1: Aktivitätsdiagramm einer gestenbasierten Interaktion

## 3.2 Aktionen des Benutzers

Bei der Durchführung der Interaktion liegen die beiden Geräte entweder gegenüber oder werden jeweils in der Hand gehalten.

Die Entfernung zwischen Quell- und Zielgerät hängt von der Übertragungstechnologie ab. Dies wird im nachfolgenden Kapitel genauer behandelt.

### 3.2.1 Geste zum Verbinden

Die Geste zum Verbinden kann je nach Interaktionstyp variieren. Man unterscheidet zwischen Interaktionen, bei denen das Zielgerät eindeutig identifiziert werden kann und denen, bei denen der Benutzer anhand einer Liste verfügbarer Geräte das Zielgerät festlegt.

Ersteres kann z.B. ein „Bump“ sein, welches bereits in einer früheren Arbeit betrachtet wurde. Die beiden Geräte werden aneinander gestoßen, wodurch eine Koppelung anhand der Beschleunigungssensoren im Gerät identifiziert werden kann (siehe Abb. 3.2).

Sollte der Benutzer aus einer Liste auswählen, so ist die Geste zum Verbinden meist ein Klick auf einen „Verbinden“-Button.

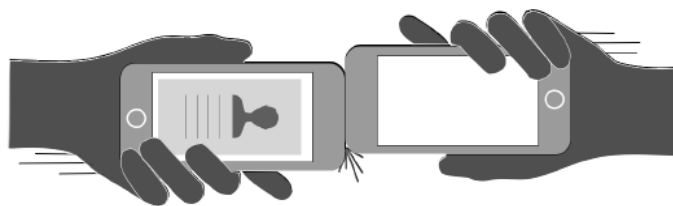


Abbildung 3.2: Bump zweier Geräte zur eindeutigen Identifizierung

### 3.2.2 Auswahl der 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. 3.3). 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. 3.4).



Abbildung 3.4: Auswahl der Dateien

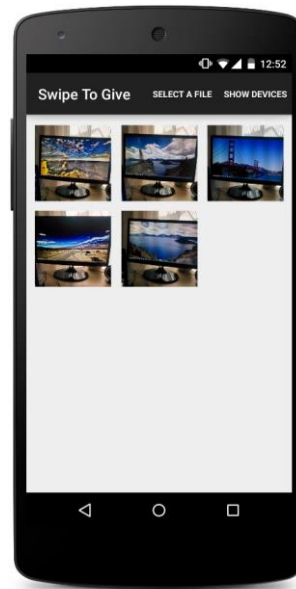


Abbildung 3.3: Darstellung der ausgewählten Dateien

### 3.2.3 Geste zum Senden

Nachdem die Bilder fertig ausgewählt werden und eine Verbindung zu einem anderen Gerät besteht, können die Dateien durch eine Geste versendet werden. In diesem Anwendungsfall ist es ein Swipe vom unteren Ende des Displays zum Oberen. Sollte die Geste ausgeführt werden ohne dass eine Verbindung besteht, springt die App automatisch auf die Liste der verfügbaren Geräte.

### **3.3 Reaktion des Systems**

Die Reaktion des Systems, nach dem Senden der Dateien, wird in Reaktion des Senders und Reaktion des Empfängers unterschieden.

#### **3.3.1 Reaktion des Systems beim Sender**

Die Reaktion beim Sender ist eine Pop-up-Nachricht mit der Nachricht, dass die Dateien übertragen wurden. Des Weiteren wird die Liste der ausgewählten Bilder geleert. Sollte es zu einem Fehler kommen, wird der Fehlertext angezeigt.

#### **3.3.2 Reaktion des Systems beim Empfänger**

Die eigentliche Reaktion ist die übertragene Datei anzuzeigen (visuelle Rückmeldung). 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.

### **3.4 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 nach Terrenghi et al. [8] basierend auf ihrer Bildschirmgröße kategorisiert. Die folgende Tabelle liefert einen Überblick über diese Kategorien.

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	$\geq 20$ m	Mehrere Displays

Tabelle 3.1: Kategorisierung von Endgeräten

Um für die Interaktion geeignete Geräte zu identifizieren gilt es erst zu definieren, welche verschiedenen Rollen ein Gerät in der Interaktion einnehmen kann. Die Endgeräte können aufgeteilt werden in Quell- und Zielgeräte. Ein Quellgerät ist der Informationsträger, von welchem Daten durch das Swipen auf das Zielgerät übertragen werden. Bei einem Quellgerät muss es sich um ein Endgerät handeln, welches von der Klassifizierung unter die Kategorie Inch-, Foot oder Yard size fällt.

Bei Zielgeräten hingegen kann es sich sowohl um ein mobiles als auch um ein stationäres Endgerät handeln. Stationäre Endgeräte zeichnen sich dadurch aus, dass Sie vom Anwender bei der Nutzung nicht in der Hand gehalten werden.

Die definierten Gerätekategorien können jetzt auf die jeweiligen Rollen aufgeteilt werden. Quellgeräte können nur Geräte aus den Kategorien Inch size, Foot size und Yard size sein. Zielgeräte können zusätzlich auch aus den Kategorien Yard-, Perch- und Chain size kommen (s. Tabelle 3.1).

Für die Quellgeräte 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. Da hier keine Interaktion stattfindet, eignen sich alle Geräte.



### **3.5 Umsetzung der Interaktion**

Um die Swipe-Interaktion umzusetzen, müssen zwei Apps entwickelt werden, eine für das Quell- und eine für das Zielgerät.

Die Apps müssen eine Verbindung zwischen Quell- und Zielgerät herstellen und einen Informationskanal öffnen, damit Dateien übertragen werden können. Hierbei wird das Quellgerät als Client agieren und das Zielgerät als Server.

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 identifizierten System.

### 4.1 Vorbedingungen

#### **App**

Um eine Dateiübertragung per Swipe zu realisieren, muss die Swipe-Geste für das Senden reserviert werden. Dies ist in der normalen Android-Umgebung nicht leicht zu realisieren; hierfür müsste das Betriebssystem manipuliert werden. Aufgrund dessen muss eine App erstellt werden, in der das Swipen für die Dateiübertragung genutzt wird. Die Auswahl der Dateien muss zwingend in der App geschehen. Das Senden eines geöffneten Bildes funktioniert demnach nicht.

#### **Swipe**

Im umgesetzten Prototyp wird die Datenübertragung erst ausgeführt, wenn der Swipe ausreichend lang ist. Dies verhindert das Senden, wenn der Finger das Display berührt. Außerdem muss der Swipe nach oben ausgeführt werden. Dies wird in der Abbildung 4.1 verbildlicht.

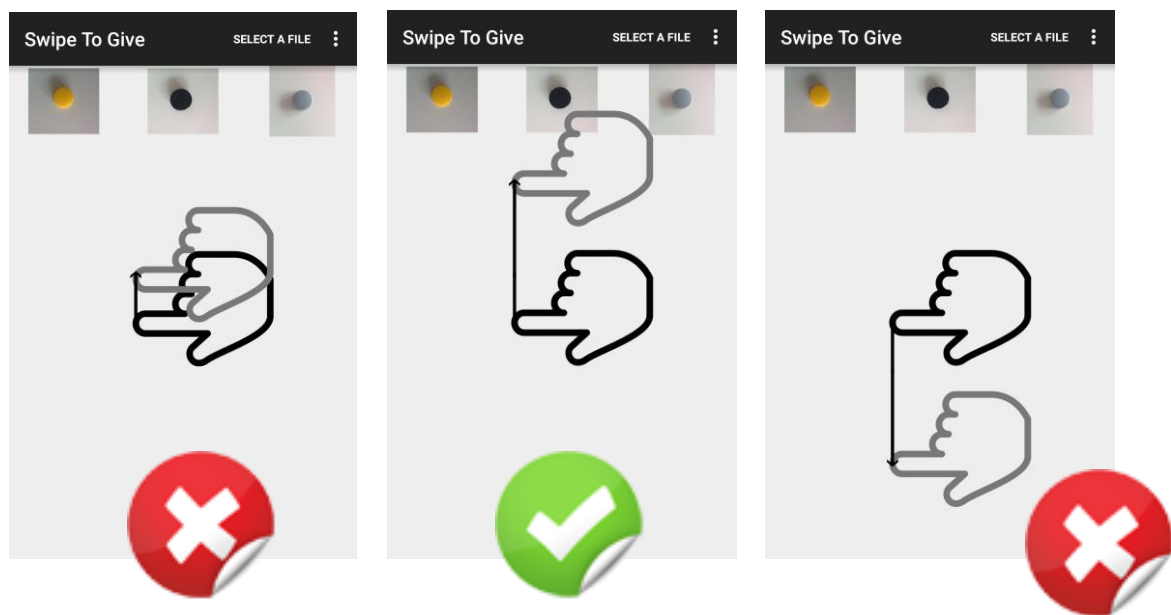


Abbildung 4.1: Swipe-Parameter

## 4.2 Ablauf

Für die hier zu konzipierende Interaktion „Swipe To Give“, wird der Ablauf wie folgt festgelegt (siehe Abbildung 4.2). Dies ist eine Variation des im Kapitel 3.1 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.

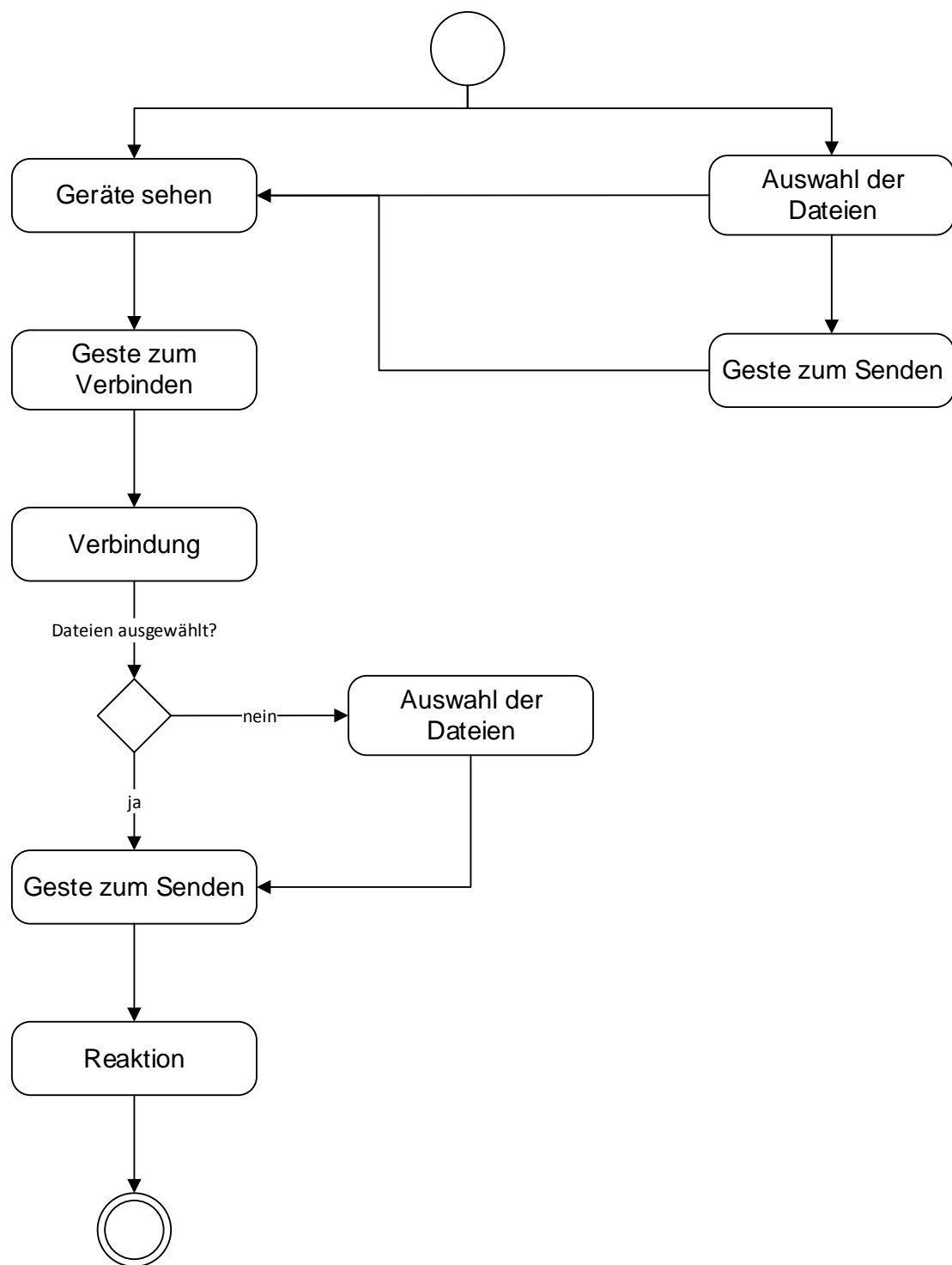


Abbildung 4.2: Aktivitätsdiagramm "Swipe To Give"

### **4.3 Bestimmung Peer-to-Peer Verbindung**

Wie zuvor beschrieben, stehen mehrere Peer-to-Peer Technologien zur Verfügung.

#### **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.

## 4.4 Grundlagen zu Android Frameworks

Das Android-Betriebssystem bietet durch seine Frameworks Zugang zu diversen Technologien. Im Folgenden wird die grundlegende Funktionsweise von dem Framework beschrieben, das zur Umsetzung der Swipe-Interaktion genutzt wurde.

### 4.4.1 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.

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 zählt die Verbindung via WiFi Direct, Bluetooth und NFC. Für die jeweilige Verbindungsart existieren unterschiedliche Erweiterungen.

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`

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.

`WIFI_P2P_CONNECTION_CHANGED_ACTION`

Wird aufgerufen, wenn sich die Verbindung mit anderen Geräten geändert hat.

WIFI\_P2P\_THIS\_DEVICE\_CHANGED\_ACTION

Wird aufgerufen, wenn sich die Konfiguration des eigenen Geräts geändert hat.

Sobald einer dieser Fälle eintritt, wird der 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. 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;
        }
    }
}
```

Der Verbindungsaufbau lässt sich auch über ein Aktivitätsdiagramm verbildlichen (siehe Abbildung 4.3).

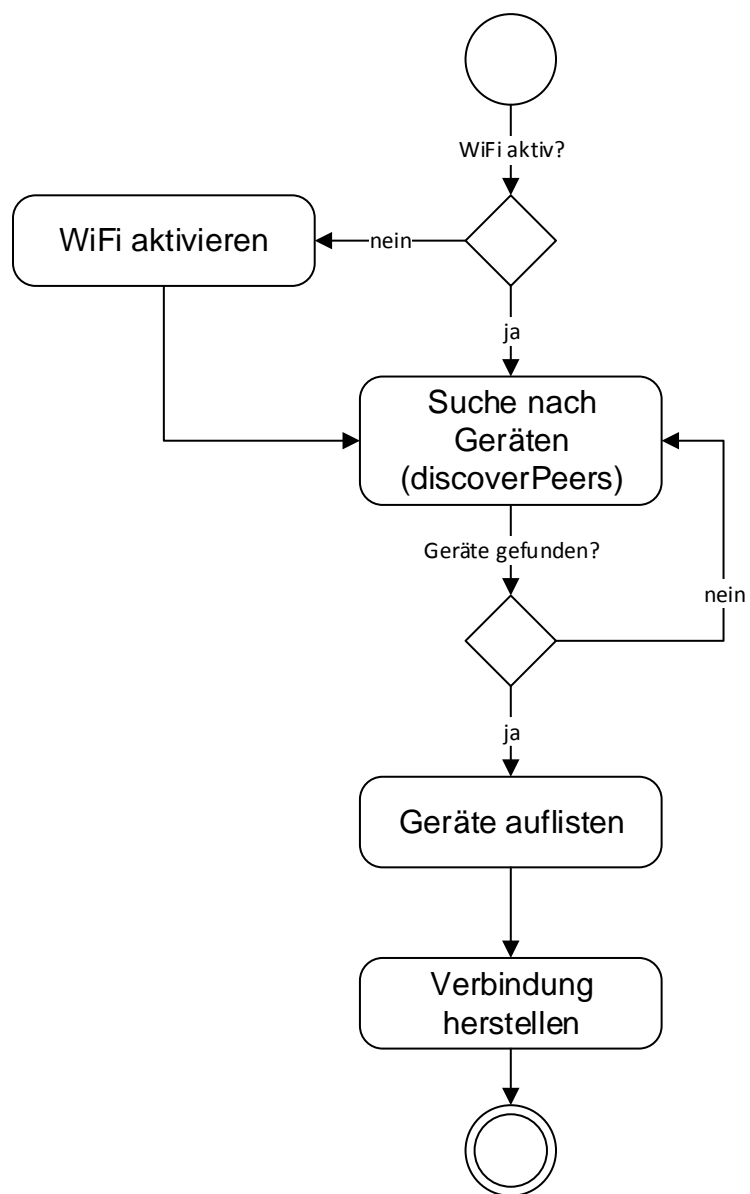


Abbildung 4.3: Aktivitätsdiagramm des Verbindungsaufbaus



## 4.5 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.

Direkte Kommunikation bietet den Vorteil, ohne zusätzliche Server-Infrastruktur zu operieren und es wird, falls der Server nur über das Internet erreichbar ist, auch keine Internetverbindung benötigt. Nachteilig ist jedoch der höhere Implementierungsaufwand, um für jede Plattform ein System anzubieten. Die von den Plattformen genutzten Technologien und Frameworks sind teilweise nicht kompatibel, was es erforderlich macht, für jede Plattform Einzellösungen zu implementieren.

Wird dagegen eine Serverarchitektur gewählt, ist der Implementierungsaufwand wesentlich geringer, da die meiste Implementierung nur einmal für den Server erfolgen muss. Nachteilig ist der erforderliche Zugang zu entweder einem lokalen WLAN oder zu mobilem Internet. Weiterhin handelt es sich bei einem Server um einen Single-Point-Of-Failure. Fällt der Server aus, funktioniert das komplette System nicht mehr. Hier bieten lokale Lösungen einen klaren Vorteil, da diese ausfallsicher sind. Weiterhin ist der Betrieb eines Servers auch mit Kosten verbunden die in einem lokalen System nicht anfallen. [9]

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 4.4).

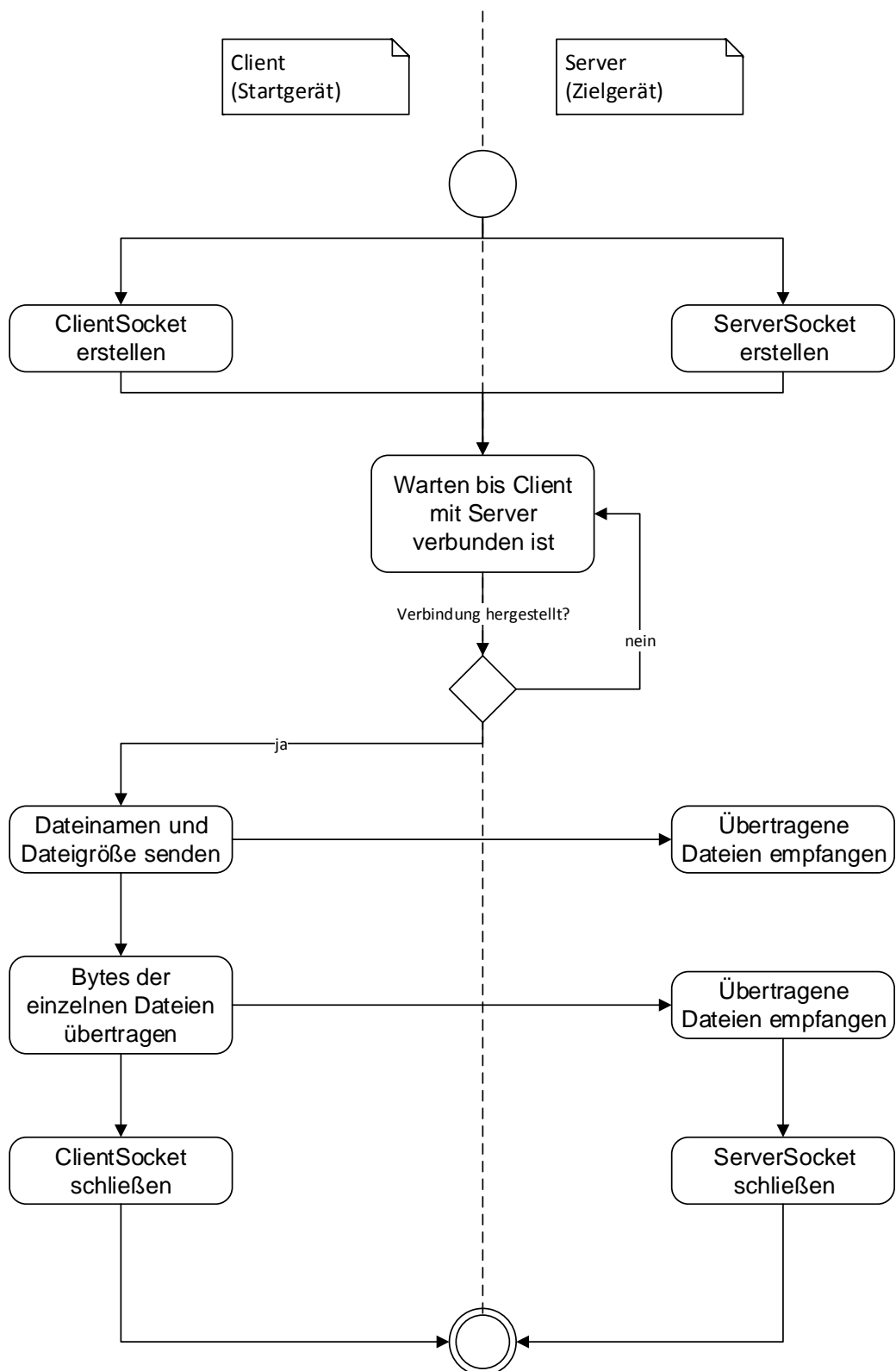


Abbildung 4.4: Aktivitätsdiagramm Dateiübertragung

---

## 5 Demonstrator Applikation

---

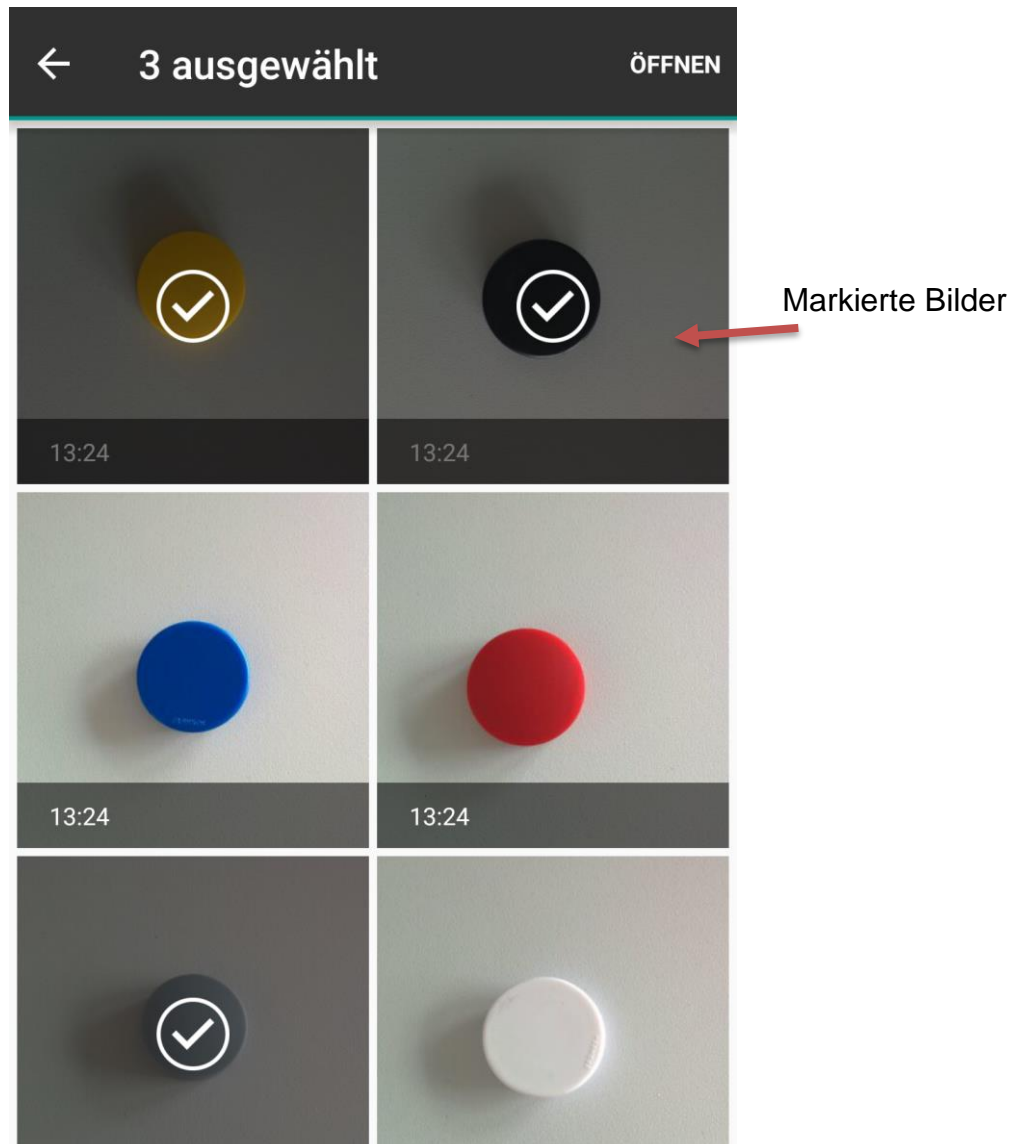
Als praktischer Teil dieser Arbeit wurde die netzwerkbasierte App implementiert. Mit der Applikation kann der Anwender Fotos, welche auf seinem Endgerät gespeichert sind, auswählen und per Swipe übertragen.

Der Prototyp wurde für das Betriebssystem Android implementiert, welches als Programmiersprache auf Java basiert. Das frei verfügbare Git Repository ist unter folgendem Link erreichbar:

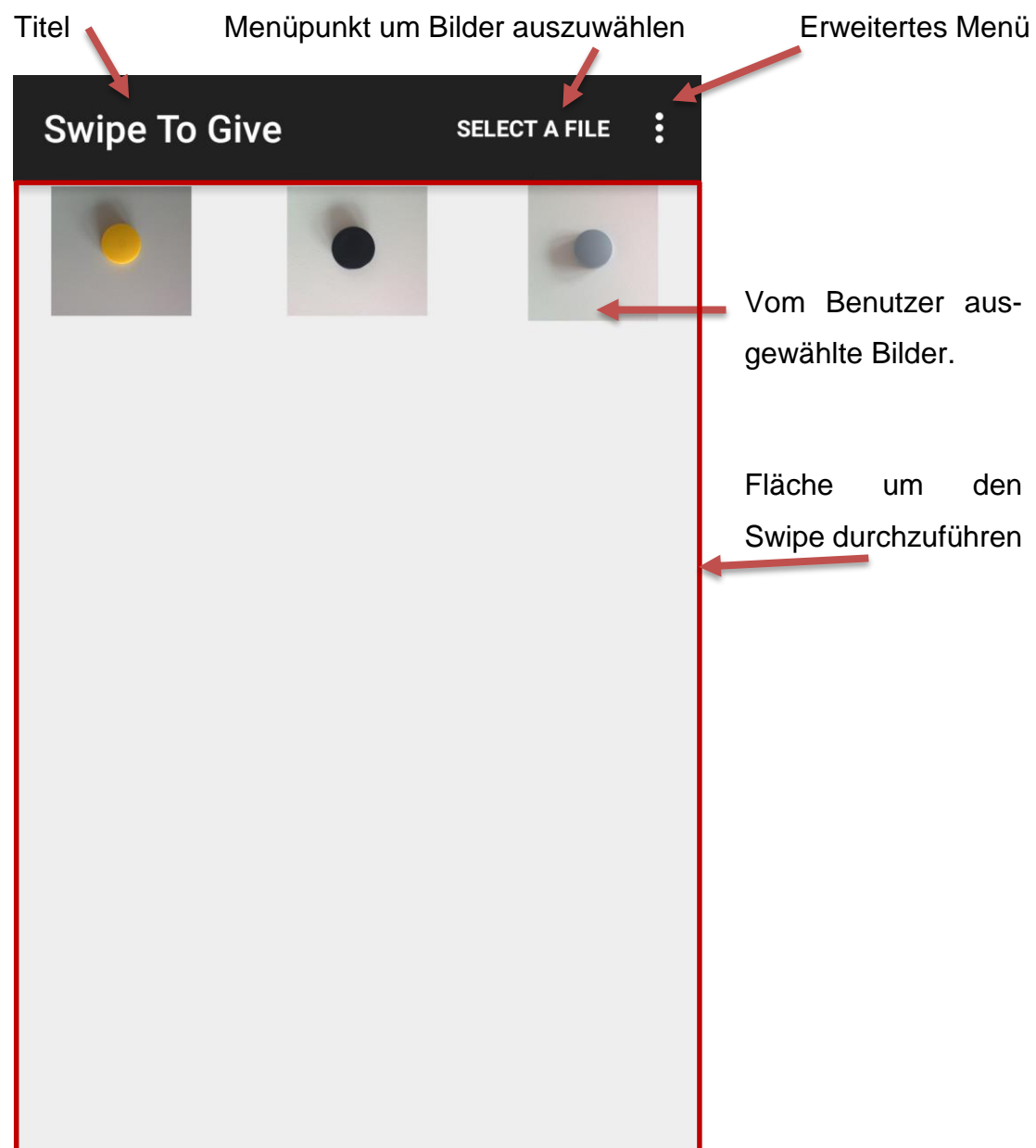
<https://github.com/informatik-mannheim/thesis-swipe-to-give>

Folgend wird das Wireframe in Auszügen dargestellt um einen Einstieg in die Nutzung der Applikation zu bieten. Auf den Screenshots ist sichtbar welche Interaktionsmöglichkeiten dem Anwender, vor und nach der Durchführung der Interaktion, geboten werden.

## 5.1 Quellgerät



Die Auswahl der Bilder erfolgt über den Standard-Android Dateimanager. Dies ist für den Benutzer intuitiv.



# WiFi Direct Devices



BRAVIA KDL-50W656A

Available

CONNECT

DISCONNECT

Device address: 3e:77:e6:33:7c:60

Liste der verfügbaren Geräte

Möglichkeit sich mit dem ausgewählten Gerät zu verbinden

## 5.2 Zielgerät

The screenshot shows the 'Swipe To Give Receiver' app interface. At the top, there is a header bar with the title 'Swipe To Give Receiver' and a menu icon. Below the header, there is a list of available devices. The first device is 'Nexus 5' with a status of 'Connected'. The second device is 'BRAVIA KDL-50W656A' with a status of 'Available'. Below the list, there is a section for the selected device, which includes a 'DISCONNECT' button and the following information: '3e:77:e6:33:7c:60', 'Group Owner IP - 192.168.49.1', and 'group owner: yes'. At the bottom of the screen, there is a black bar with the text 'Give me your files!'.

Swipe To Give Receiver

Nexus 5  
Connected

BRAVIA KDL-50W656A  
Available

DISCONNECT

3e:77:e6:33:7c:60  
Group Owner IP - 192.168.49.1  
group owner: yes

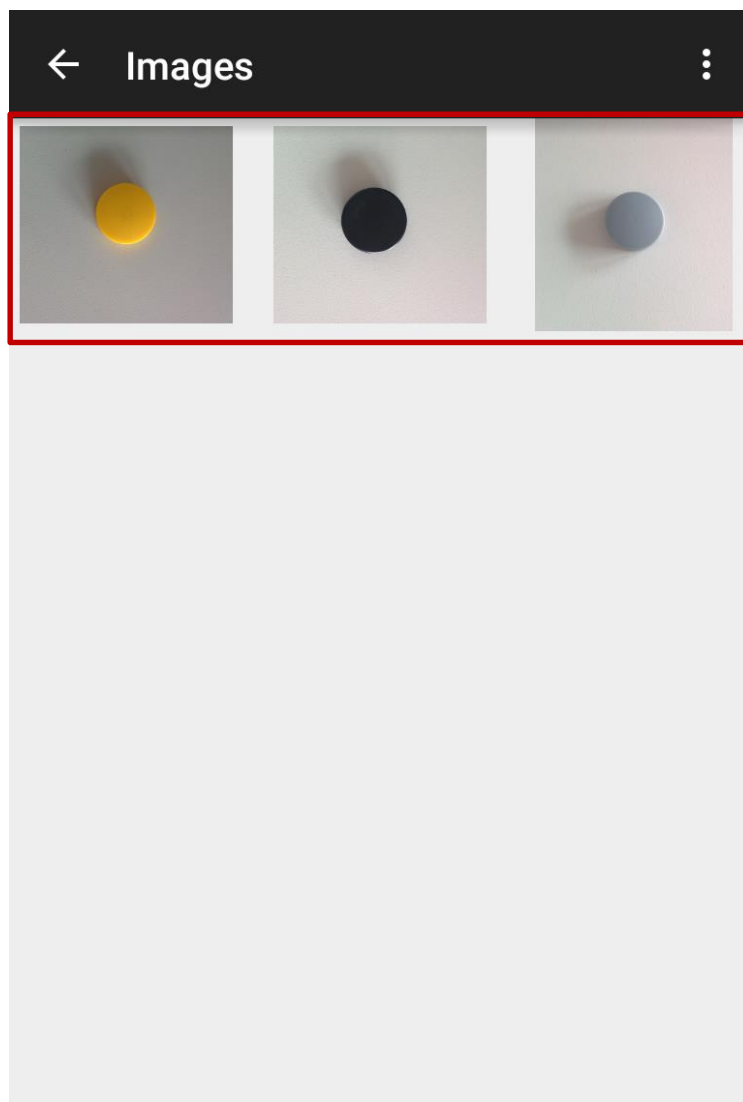
Give me your files!

Liste der verfügbaren Geräte

Status des Geräts

Informationen über das ausgewählte Gerät

Information, dass Verbindung erfolgreich aufgebaut wurde



Übertragung war erfolgreich und die empfangenen Bilder werden angezeigt.





---

## 6 Interaktionspattern

---

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

### 6.1 Was

#### Allgemeine Beschreibung

Der Benutzer hat ein Gerät A in der Hand oder vor sich stehen. Mit einer Wischbewegung vom unteren zum oberen Bildschirmrand auf A, in Richtung von Gerät B, werden Daten von A nach B übertragen.

#### Kategorie

- ☒ Geben
- ☐ Holen
- ☐ Verbinden
- ☐ Klonen
- ☐ Sonstige

#### Tags

wischen, swipe, Finger, von unten nach oben, Daten senden

## 6.2 Wie

### Aktion des Benutzers

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

Der Benutzer führt eine Wischbewegung mit einem Finger vom unteren zum oberen Bildschirmrand aus. Die Bewegung erfolgt in Richtung von Gerät B, welches die Daten empfängt.

### Reaktion des Gerätes/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.

Gerät B empfängt den Datensatz und zeigt diesen auf dem Bildschirm an. Außerdem wird ein kurzer Vibrationsalarm ausgelöst.

### Hinweise zur Gestaltung der Interaktion

Es ist gefordert, entsprechend der Geräte zu berücksichtigen, mit wie viel Fingern die Swipe-Geste ausgeführt werden soll. Beispiel: bei iOS (8)-Geräte öffnet sich bei der Swipe-Geste von unten nach oben ein Kontrollzentrum bei der Nutzung von einem Finger.

### Grafische Darstellung der Interaktion

*Daten übertragen (Quellgerät zu Zielgerät)*



Abbildung 6.1: Übertragung der ausgewählten Dateien per Swipe (re: Quelle, li: Ziel)

*Daten empfangen (auf dem Zielgerät)*



Abbildung 6.2: Visuelle Darstellung nach erfolgreicher Übertragung (re: Quelle, li: Ziel)

## 6.3 Wann

### Geeignete Verwendung/en (Kontext)

Datenaustausch: Bilder, Videos, Dateien, Social Network IDs

### Interaktionsnutzer

- ☒ Einzelnutzung
- ☒ kollaborative Nutzung

### Fähigkeiten des Nutzers

- ☐ Alltagsverständnis des Nutzers über physikalische Gegebenheiten beachtet
- ☐ Körperliches Bewusstsein und körperlichen Fähigkeiten des Nutzers beachtet
- ☐ Räumliches Bewusstsein und räumliche Fähigkeiten des Nutzers beachtet
- ☐ Soziales Verhalten und soziale Fähigkeiten des Nutzers beachtet

## Displaytypen

(Beziehung zwischen 2 verschiedene Gerätetypen)

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

Tabelle 6.1: Displaytypen

## 6.4 Warum

☐ Pattern Kandidat

☐ realisierbar

☐ futuristisch

☒ Bewährtes Interaction Pattern

---

<sup>1</sup> Inch size  $\geq 2,54$  cm  
Beispiel: Smartphone, Smartwatch

<sup>2</sup> Foot size  $\geq 30,48$  cm  
Beispiel: Tablets

<sup>3</sup> Yard size  $\geq 91,44$  cm  
Beispiel: Tabletop

<sup>4</sup> Perch size  $\geq 3,05$  m  
Beispiel: TV-Screen

<sup>5</sup> Chain size  $\geq 20$  m  
Beispiel: Mehrere Displays

## **Analoge Patterns**

Swipe To Take: selbe 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](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.

## **Metapher**

Mit der Hand Blätter (z.B. Aufgabenblätter) verteilen.

## **6.5 Technisches**

### **Benötigte Technologie/n**

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

## **6.6 Sonstiges**

### **Autoren**

Valentina Burjan, Alexander Hahn // Hochschule Mannheim

---

## 7 Zusammenfassung und Ausblick

---

Das zu Beginn dieser Arbeit formulierte Ziel wurde durch die folgenden Teilschritte erreicht: Zu Beginn wurde beschrieben, welche Kategorien von Endgeräten für die Geste geeignet sind und welche Benutzeraktionen nötig sind. 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. Im Zuge des Swipes ist es interessant zu erforschen, ob die Geräte aufeinander zeigen müssen um Daten

übertragen zu können, oder die Ausrichtung der Geräte keine Rolle spielt. Hierbei kann die Abbildung 7.1 mit der Abbildung 7.2 verglichen werden. Es wäre auch möglich, dass diese Eigenschaft die Nutzungsqualität einschränkt, deshalb sollte dies in einer Studie betrachtet werden.



Abbildung 7.1: Geräte sind gegenüberliegend ausgerichtet



Abbildung 7.2: Geräte sind nicht gegenüberliegend ausgerichtet

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 [13] 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. Vielleicht 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)

---

## Tabellenverzeichnis

---

Tabelle 3.1: Kategorisierung von Endgeräten.....	24
Tabelle 6.1: Displaytypen .....	44

---

## Abbildungsverzeichnis

---

Abbildung 2.1: Mobile und stationary (fest-angebrachte) Geräte [6].....	10
Abbildung 2.2: Visuelle Benachrichtigung, dass die Geräte kommunizieren können [6].....	11
Abbildung 2.3: Darstellung der Fotos auf dem Gerät [6].....	11
Abbildung 2.4: Maschentopologie.....	13
Abbildung 2.5: Sterntopologie.....	13
Abbildung 2.6: Piconetz / Bluetooth-Topologie, [2] .....	15
Abbildung 2.7: Client-Server-Modell .....	18
Abbildung 3.1: Aktivitätsdiagramm einer gestenbasierten Interaktion.....	20
Abbildung 3.2: Bump zweier Geräte zur eindeutigen Identifizierung .....	21
Abbildung 3.3: Darstellung der ausgewählten Dateien .....	22
Abbildung 3.4: Auswahl der Dateien.....	22
Abbildung 4.1: Swipe-Parameter .....	27
Abbildung 4.2: Aktivitätsdiagramm "Swipe To Give" .....	28
Abbildung 4.3: Aktivitätsdiagramm des Verbindungsaufbaus .....	32
Abbildung 4.4: Aktivitätsdiagramm Dateiübertragung .....	34
Abbildung 6.1: Übertragung der ausgewählten Dateien per Swipe (re: Quelle, li: Ziel) .....	42
Abbildung 6.2: Visuelle Darstellung nach erfolgreicher Übertragung (re: Quelle, li: Ziel) .....	43
Abbildung 7.1: Geräte sind gegenüberliegend ausgerichtet .....	48
Abbildung 7.2: Geräte sind nicht gegenüberliegend ausgerichtet.....	48

---

## 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](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] „wi-fi.org,“ [Online]. Available: <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>. [Zugriff am 07 04 2015].
- [4] P. Schnabel, Kommunikationstechnik-Fibel, Ludwigsburg: Books on Demand, 2003.
- [5] A. Haller und N. Lando, „Efficient server polling system and method“. Patent WO2006103680 A3, 31 03 2006.
- [6] *Gradual Engagement between Digital Devices as a Function of Proximity*. [Film]. 2013.
- [7] „[www.appato.com](http://www.appato.com),“ [Online]. Available: <http://www.appato.com/s-s-royal-ltd/push2send/>. [Zugriff am 10 04 2015].
- [8] L. Terrenghi, A. Quigley und A. Dix, A taxonomy for and analysis of multi-person-display ecosystems, Springer-Verlag London, 2009.
- [9] B. Grab, *Implementierung einer gestenbasierten Interaktion zum Datenaustausch zwischen mobilen Endgeräten*, Mannheim, 2015.
- [10] N. Marquardt, T. Ballendat, S. Boring, S. Greenberg und K. Hinckley, „[research.microsoft.com](http://research.microsoft.com),“ 2012. [Online]. Available: <http://research.microsoft.com/en-us/um/people/kenh/All-Published-Papers/Gradual-Engagement-ITS-2012.pdf>. [Zugriff am 29 04 2015].
- [11] S. Sokolov, „[prmac.com](http://prmac.com),“ 12 05 2012. [Online]. Available: <http://prmac.com/release-id-43191.htm>. [Zugriff am 29 04 2015].

- [12] „hoccer.com,“ [Online]. Available: <http://hoccer.com/de/>. [Zugriff am 29 04 2015].
- [13] „android.com,“ [Online]. Available: <http://www.android.com/tv/>. [Zugriff am 07 05 2015].
- [14] L. Cheng, „Service Advertisement and Discovery in Mobile Ad hoc,“ *Laboratory Of Networking Group*, 2002.

