

Stitch-to-Tile

EINE GESTENSTEUERUNG ZUR AUSLÖSUNG VON
BILDSCHIRMERWEITERUNG

1. Stitch, 2. Tile

- Die Stitch-to-Tile Interaktion besteht aus 2 Teilen
- Die beiden Teile stehen in einem Aktion-Reaktion Verhältnis
 - Auslöser: Geste (Stitch)
 - Effekt: Bildschirmverweiterung (Tile)
- Stitch-to-Tile ist eine Synchronous Gesture

Das Spontaneous Device Sharing Problem

“Das ‘Spontaneous Device Sharing Problem’ stellt die folgende Frage: Wie kann ein Benutzer eine zweckvolle Verbindung herstellen zwischen zwei oder mehr Zielgeräten ohne dass sie vorher gegenseitig ihre Netzwerkadressen kennen?”

(Hinkley 2004)

- Lösungen vor allem nützlich für spontane, kurze Aktionen (z.B. Bild versenden)
- Verschiedene Technologien mit jeweiligen Vor- und Nachteilen
- „Synchronous Gestures“ könnten eine Lösung bieten

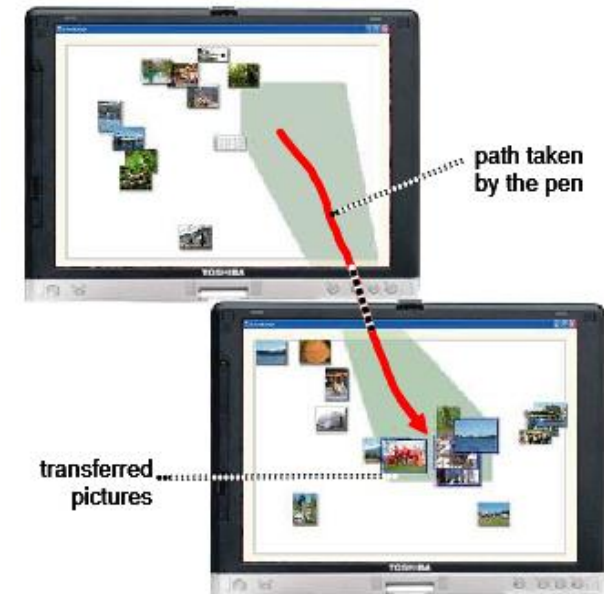
Synchronous Gestures

- Aktivitätsmuster, die eine neue Bedeutung annehmen wenn sie in einer bestimmten zeitlichen Abfolge geschehen
- Sensordaten werden verglichen um S.G. zu erkennen
- Können das Spontaneous Device Sharing Problem lösen:
 - Eine Verbindung wird durch eine S.G. hergestellt

State of the Art: Synchronous Gestures

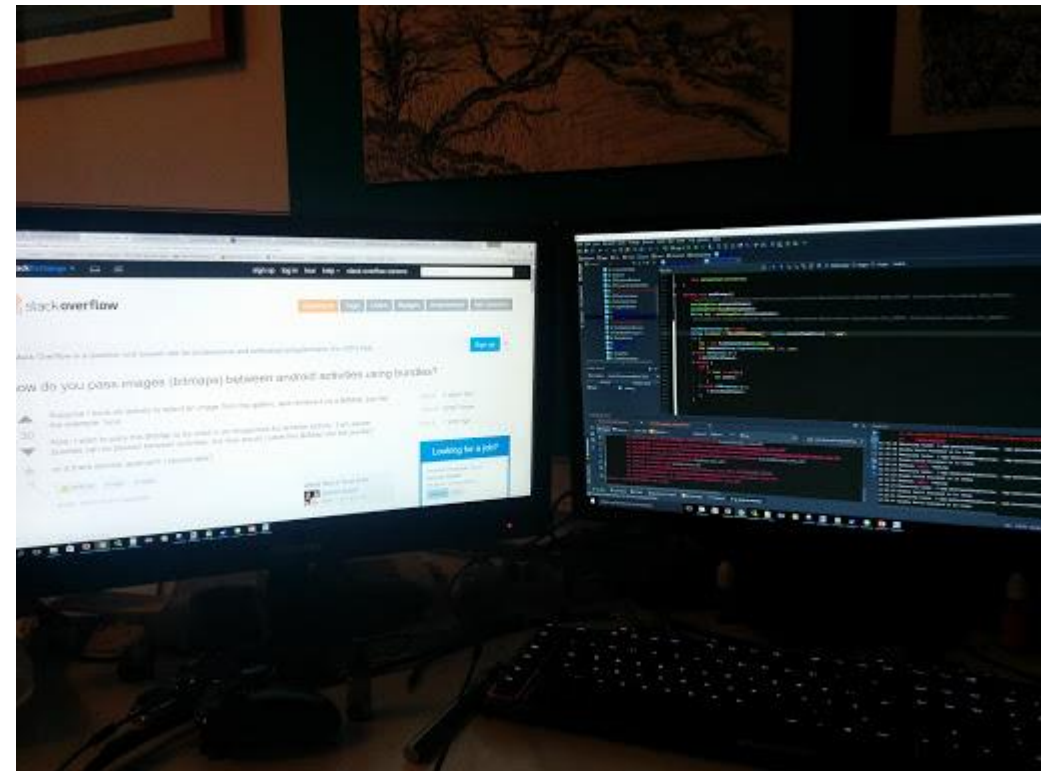
Stitching:

- Von Hinkley und Kollegen entwickelte S.G.
- Prototypische Umsetzung namens „Stitchmaster“
 - Foto-Sharing Anwendung auf Tablet PCs
- Stylus wird von einem Bildschirm auf den nächsten geführt
- Metapher: Zusammennähen zweier Stoffteile
 - Stylus dient als Nadel
 - Bildschirme als Stoffteile



Bildschirmerweiterung

- Weit verbreitet bei Desktop PCs
- Nützlich für Arbeiten die
 1. Viel Bildschirmfläche (z.B. Videobearbeitung)
 2. Häufiges wechseln zwischen Fenstern (z.B. Programmieren)erfordern.
- Weitgehend ungenutzt bei mobilen Endgeräten



State of the Art: Display Tiling auf mobilen Endgeräten

iPodWall:

- 20 iPods an einem Holzkasten
- App läuft auf jedem
- Zeigt Slideshow von Bildern an
- Kann ein Bild erweitert auf allen anzeigen



Tiling auf mobilen Endgeräten

- Es wurden zwei Formen der Bildschirmvergrößerung für mobile Endgeräte erkannt:

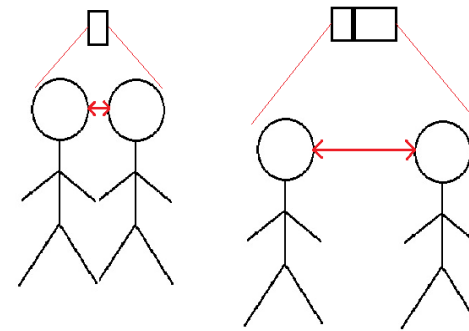
1. Visuelles Tiling
2. Logisches Tiling

Visuelles Tiling

- Reine Erweiterung der Bildschirmfläche
- Ziel: Vergrößerte Darstellung des angezeigten Inhalts
 - Unterschied zu PCs: Größerer Inhalt \leftrightarrow Mehr Inhalt

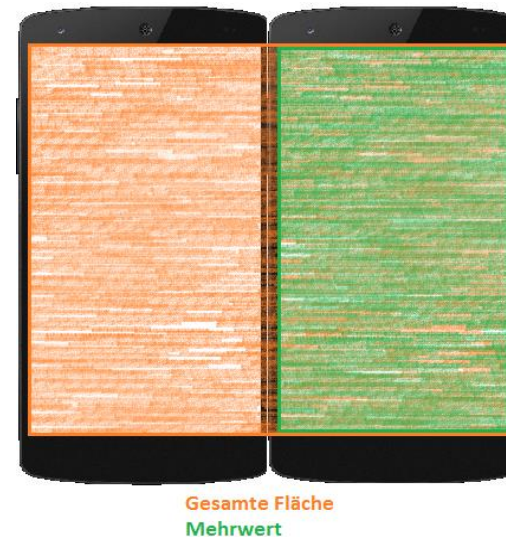
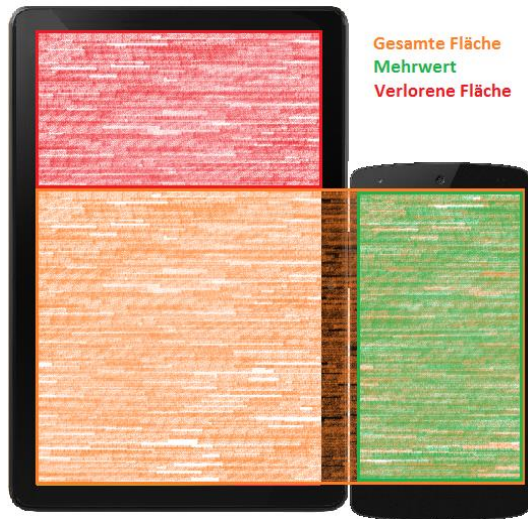
Visuelles Tiling – Anwendungen

- Videos, Foto-Slideshows etc. erscheinen größer
=> Größere Entfernung zu Geräten möglich
=> Größere Entfernung zwischen Zuschauern möglich
- Panorama Format profitiert besonders stark



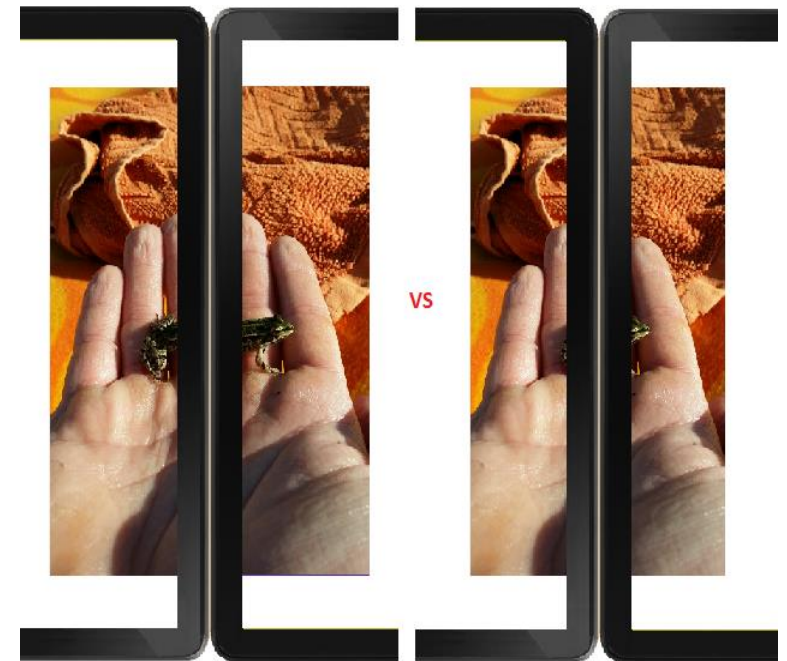
Visuelles Tiling – Anforderungen

- Bildschirmgrößen sollten ähnlich zueinander sein
- Geräte sollten mit 0mm Abstand zueinander liegen



Visuelles Tiling – Probleme

- „Verschwendete“ Bildschirmfläche bei zu unterschiedlichen Bildschirmgrößen
- Lücke zwischen den Geräten relativ zu Bildschirmflächen groß
- 2 Möglichkeiten mit der Lücke umzugehen:
 1. Lücke ignorieren
 2. Lücke miteinbeziehen



Logisches Tiling

- Ähnlichkeiten zu Second Screen
- Second Screen Apps: Begleitende Applikationen zum laufenden Fernsehprogramm
- Logisches Tiling gibt einen Teil der Hauptapp an ein Zweitgerät ab
- Unterschied zu Second Screen: Beide Geräte sind interaktionsfähig



Logisches Tiling - Anwendungen

- Ermöglicht bessere Bildschirmflächennutzung bei Applikationen mit mehreren, gleichzeitig angezeigten Funktionen

- Beispiel 1: Livestreaming App:

- Livestream auf größerem Gerät
- Chat auf kleinerem Gerät

- Beispiel 2: Spaceteam:

- Asymetrisches Multiplayer Spiel
- Ein „Kapitänsgerät“
- Mehrere „Crewgeräte“

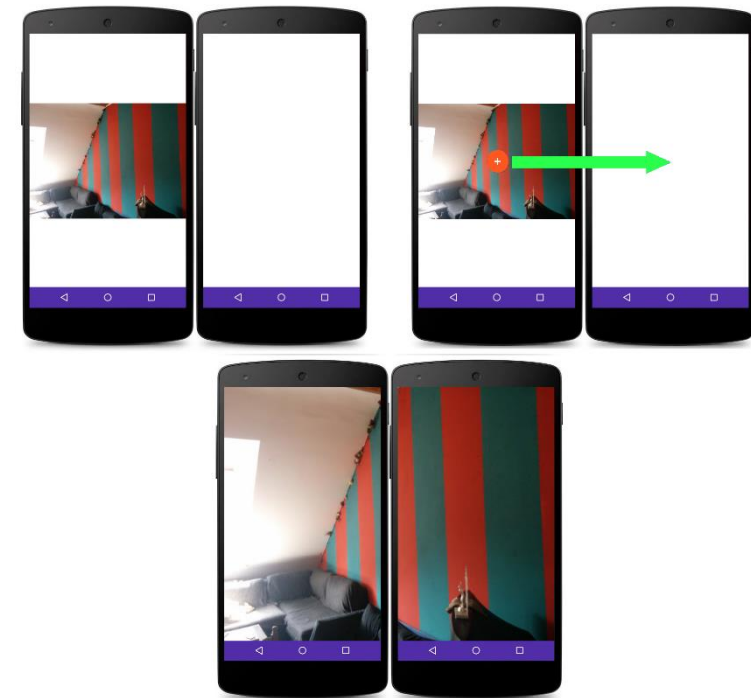


Logisches Tiling - Anforderungen

- Jegliche Kombination von Bildschirmgrößen denkbar
 - Sinnvoll: Das größte Gerät für das Anzeigen der Medien nutzen
- Geräte müssen nicht nebeneinander liegen
- Applikation muss modularisierte Anwendungslogik haben

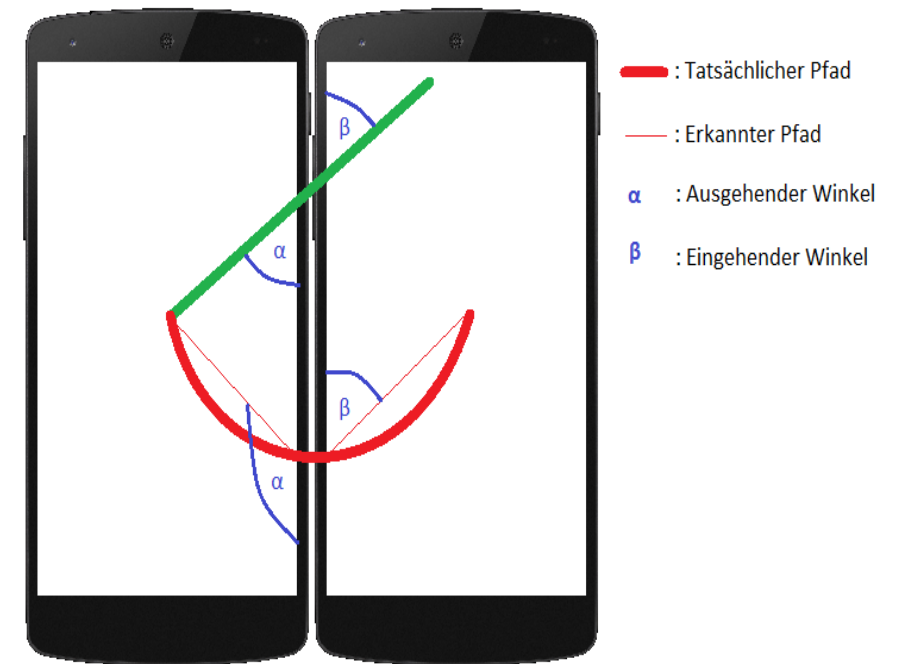
Stitching

- Löst Bildschirmenerweiterung aus
 - Nutzer führt ein Symbol von einem Bildschirm auf den nächsten
-
- Stitch besteht aus 2 Teilen:
 1. Ausgehender Swipe vom Hauptgerät
 2. Eingehender Swipe auf Zweitgerät



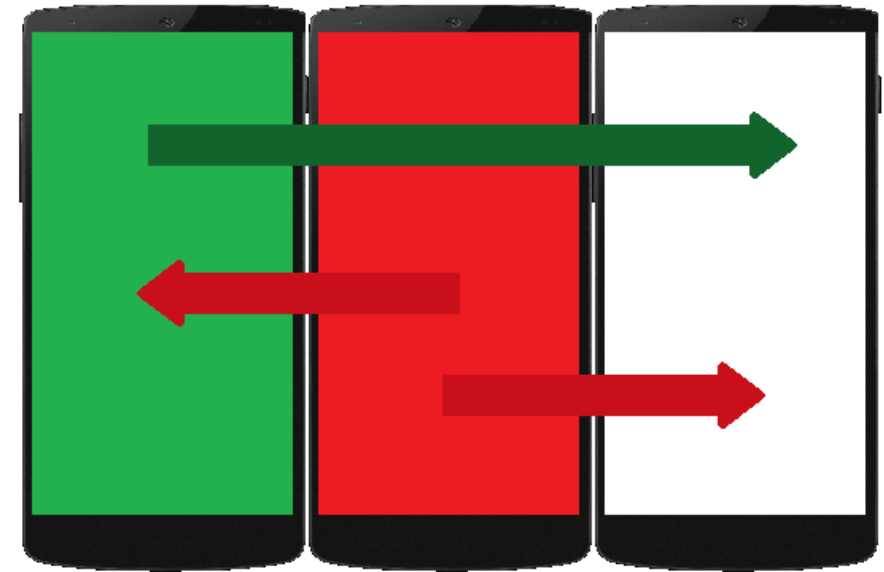
Erkennung eines Stitches

- Die beiden Swipes werden verglichen
- 3 Bedingungen:
 1. Grundrichtung muss übereinstimmen
 2. Winkel darf nicht zu unterschiedlich sein
 3. Zeit zwischen Swipes darf nicht zu groß sein



Räumliche Orientierung der Geräte

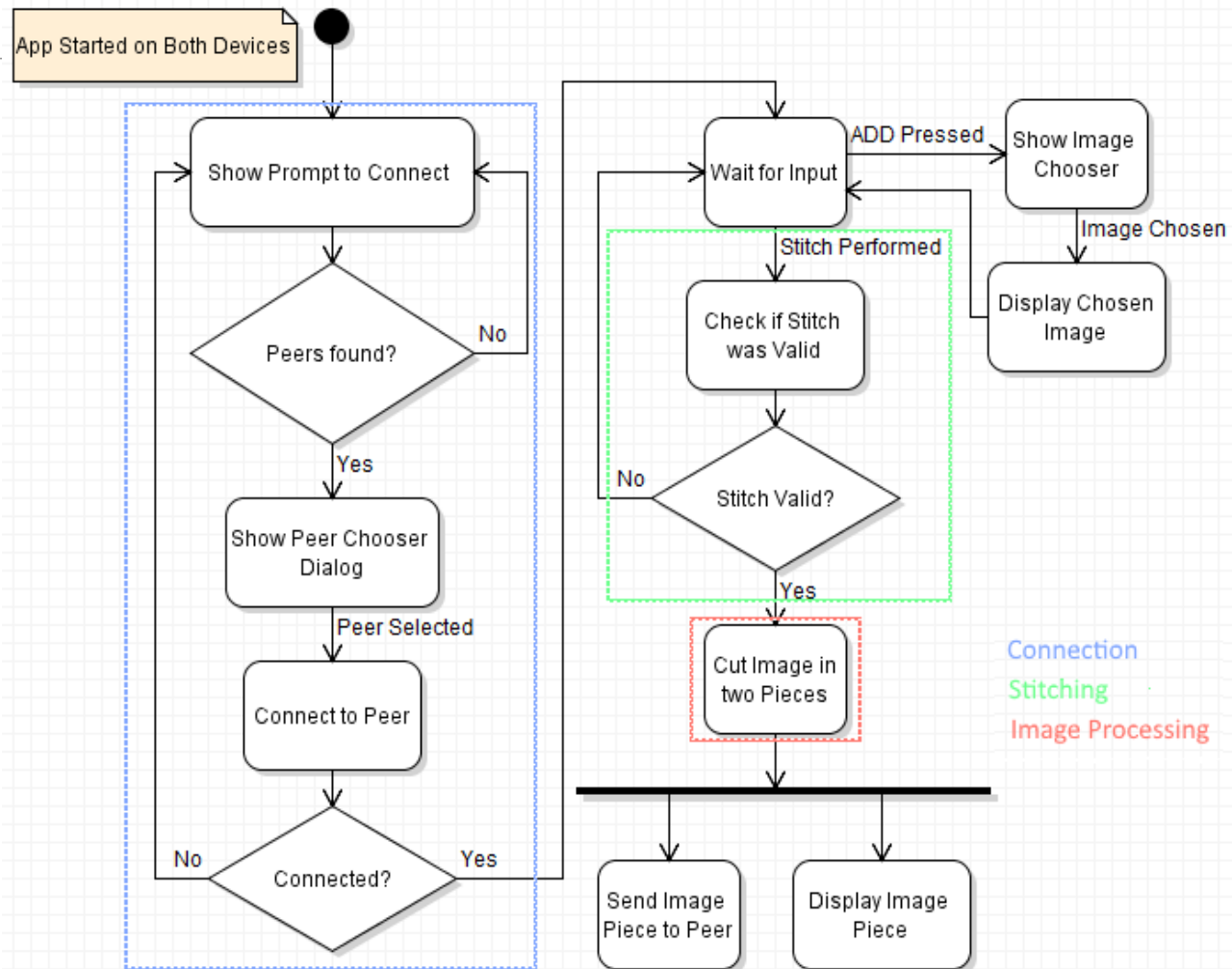
- 0mm voneinander entfernt
- Möglichst große Überschneidung der Bildschirme
- Hauptgerät liegt außen



Konflikte mit anderen Funktionen

- Fast jede Interaktion mit Smartgeräten nutzt Touch-Events
 - Risiko für Missverständnisse hoch
 - Viele Applikationen nutzen Swipes vom Bildschirmrand (z.B. zum Aufrufen von Menüs)
- Mögliche Lösung: Aktivierung von Stitch-to-Tile über Share-Funktionalität
 - Stitch-to-Tile Ebene liegt über der Applikationsebene

Prototyp – Gesamtablauf

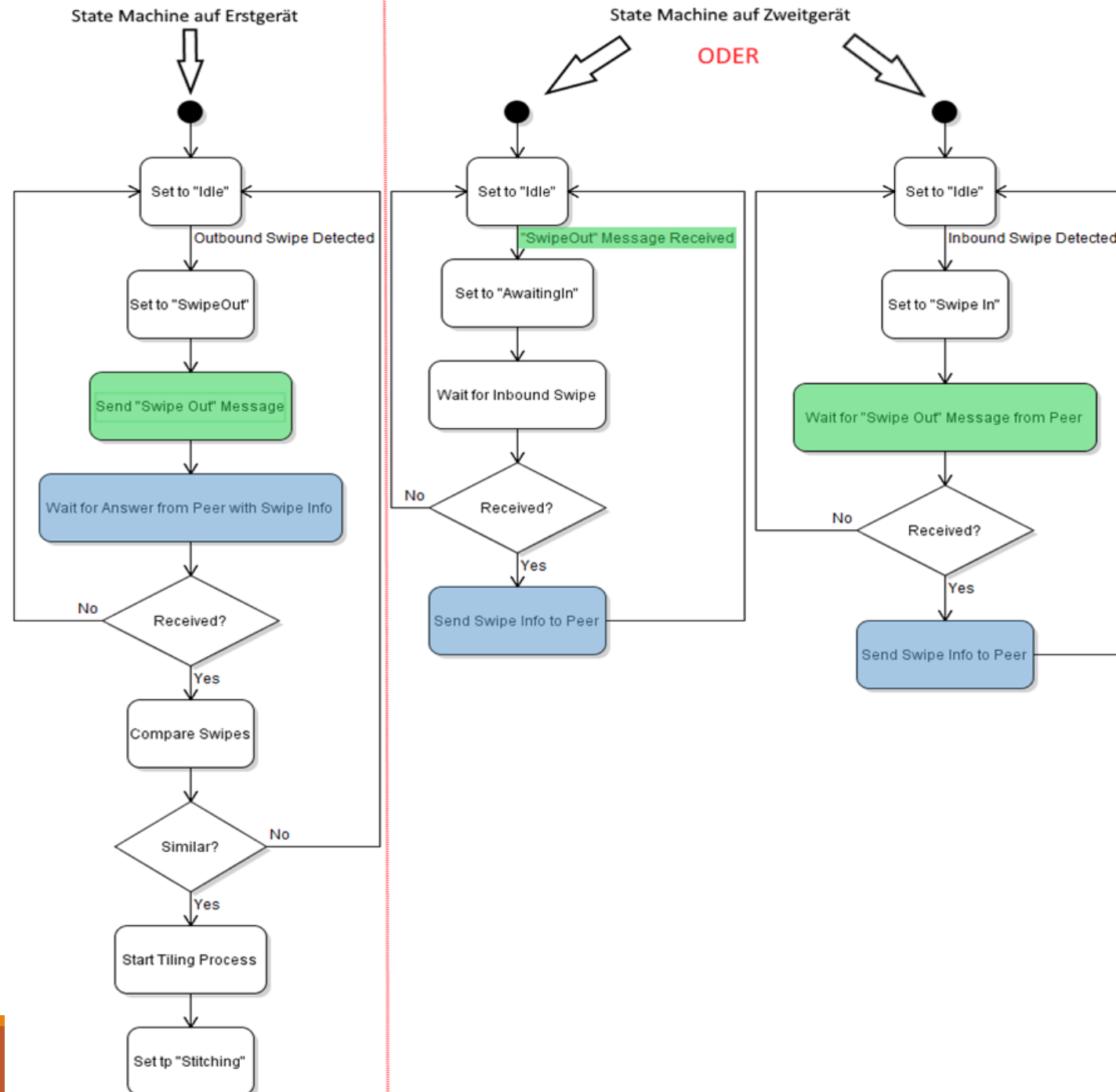


Prototyp – Demo Video



Prototyp – Erkennung eines Stitches

- Vergleich von aus- und eingehendem Swipe nötig
=> Kommunikation zwischen den Geräten nötig
- Kommunikation läuft über Services
- Services geben empfangene Daten an State Machines weiter
- Abhängig von ihrem Zustand entscheiden S.M. was mit den Daten passiert



Tiling – Bildverarbeitung

- Zur Darstellung eines Bildes auf 2 Geräten ist Bildverarbeitung nötig
- Läuft in 2 Schritten ab:
 1. Zerschneiden des zu tilenden Bildes in 2 Teile
 2. Separates Skalieren der Bildteile

Zerschneiden des Bildes

- Realweltliche Maße der Displays werden genutzt
- Jeweilige Bildanteile werden bestimmt
 - Breite des eigenen Displays / Summe aus beiden Breiten = Eigener Prozentsatz des Bildes (x)
- Zwei Bitmaps werden aus der originalen erstellt
 - Rand des Bildes bis (x * Breite des Bildes in Pixeln)
 - (x * Breite des Bildes in Pixeln) bis anderer Rand des Bildes

Skalieren der Bilder

- Gesamte Breite der Bildschirme wird genutzt
- Skalierung nötig wegen potentiell unterschiedlichen Pixeldichten der Bildschirme
- Auf kleinerem Gerät trivial
 - Das Bildteil wird auf die gesamte Pixelanzahl des Displays skaliert
- Auf größerem Gerät nicht
 - Realweltliche Höhe des Bildes muss gleich der des Display des kleineren Gerätes sein
 - Formel: Eigene Pixeldichte * Höhe des kleineren Geräts = Höhe in Pixel auf größerem Gerät
- Problem: Manche Geräte geben falsche Werte an!

Zusammenfassung & Ausblick

- Valide Geste

- Lässt sich gut implementieren aufgrund von Modularität (Aktion-Reaktion-Verhältnis)
- Anwendungsfälle für die Geste sinnvoll (Vor allem bei Logischem Tiling)

- Einige Probleme

- Fluss der Interaktion durch Verbindungsdialog gestört
- „Lügende“ Geräte

- In Zukunft

- Verbesserung der UI (Animationen, Verständlichere Symbole)
- Unterstützung für mehr als 2 Geräte
- Für dritte Apps nutzbar machen