

Macoun

Mit WatchKit Apps entwickeln

@claushoefele

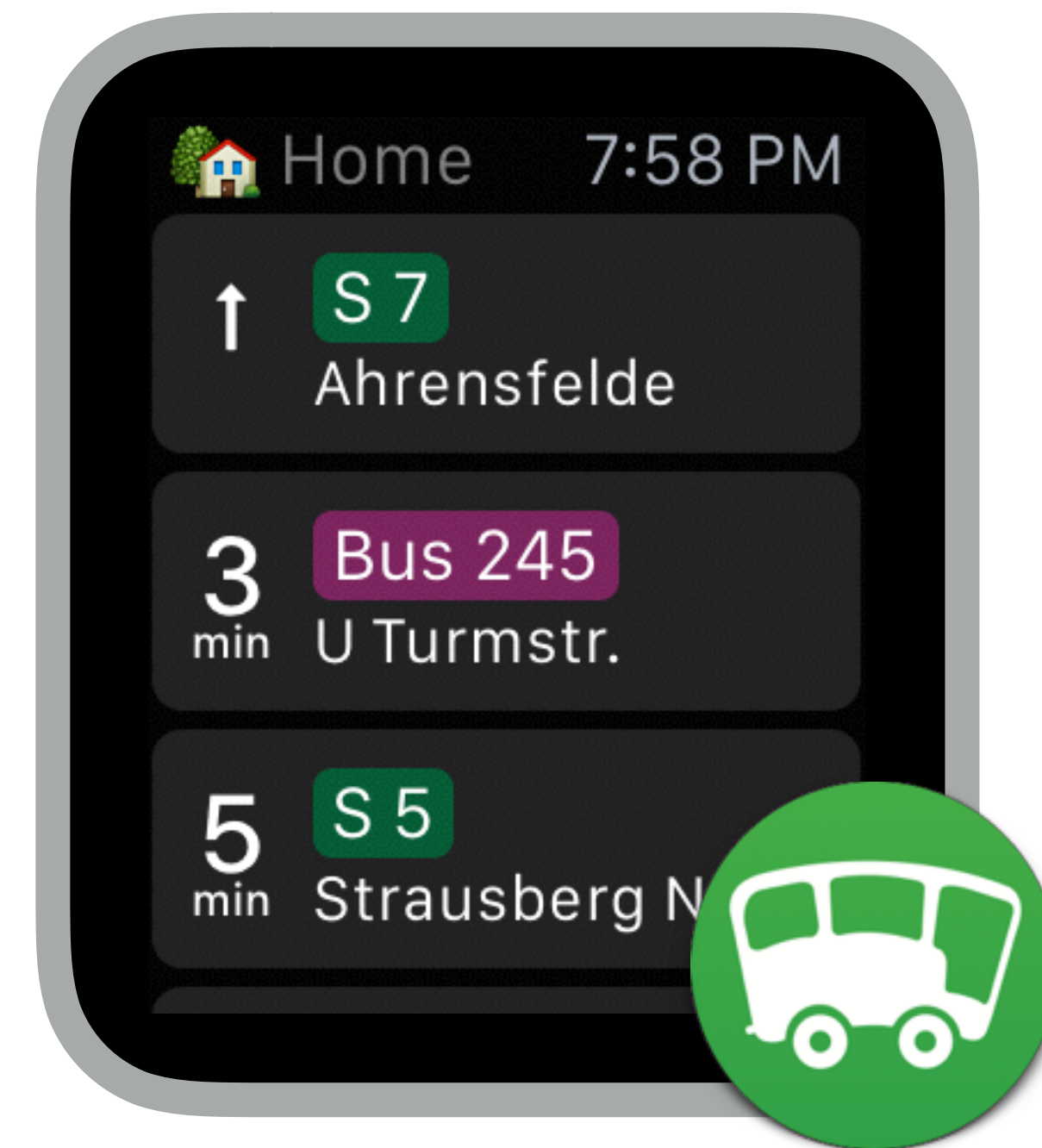


Pimp my App: Core Animation
im Großen Saal





Kleinanzeigen



TransitHopper

Nächste Besprechung



Zurückfolgen



Aufgabe erstellen mit Siri/Things



“[...] the watch is a
useful luxury [...]”

—*Benedict Evans*

Ablauf

- Interaktionsmodelle der Apple Watch
- Von watchOS 1 zu watchOS 2
- Überblick Implementierung
- Netzwerkzugriff & Datensynchronisation

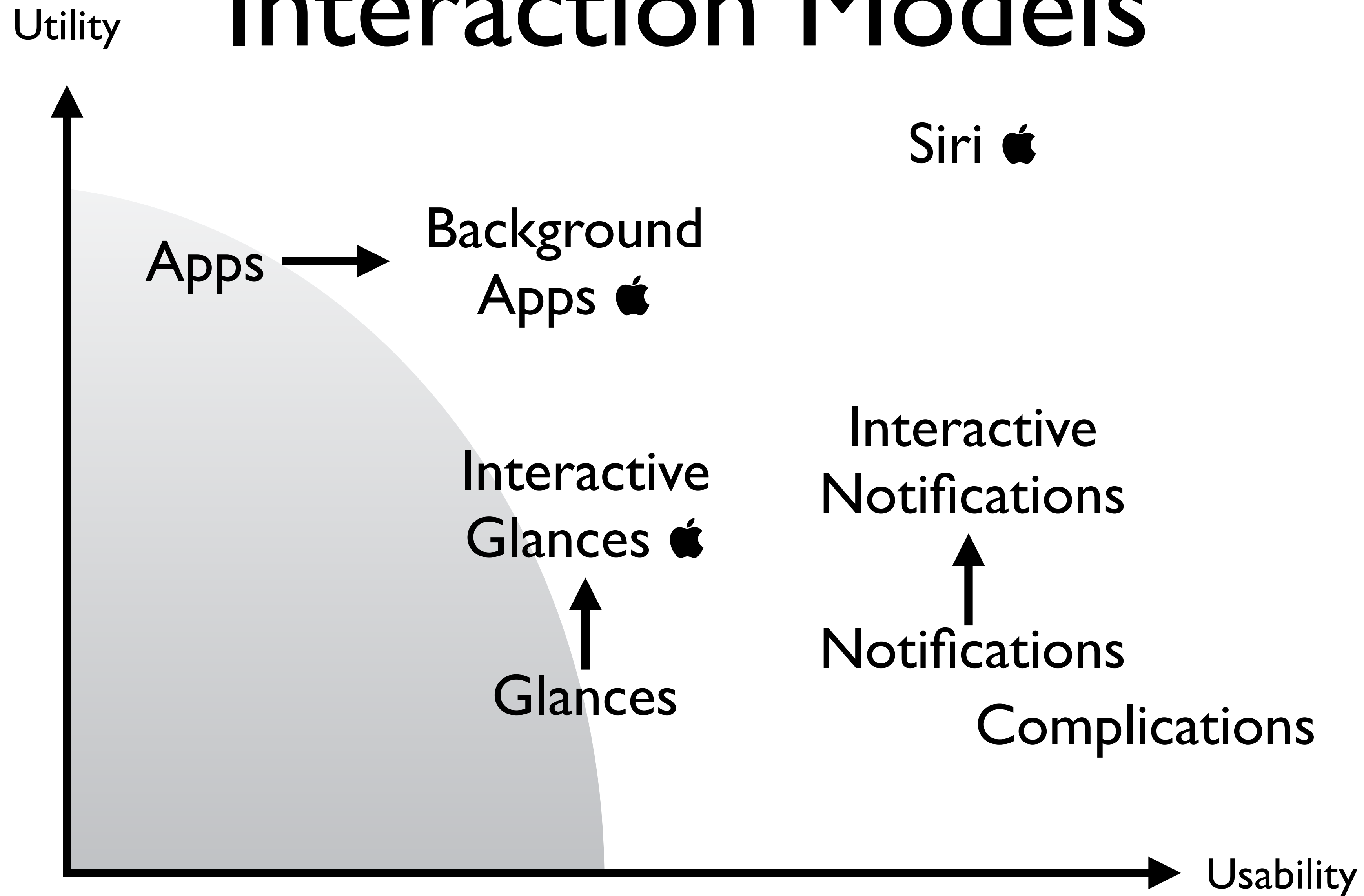
Utility/Usability Model

Utility



Usability

Interaction Models



Prioritäten

1.

Interactive
Notifications

2.

Complications

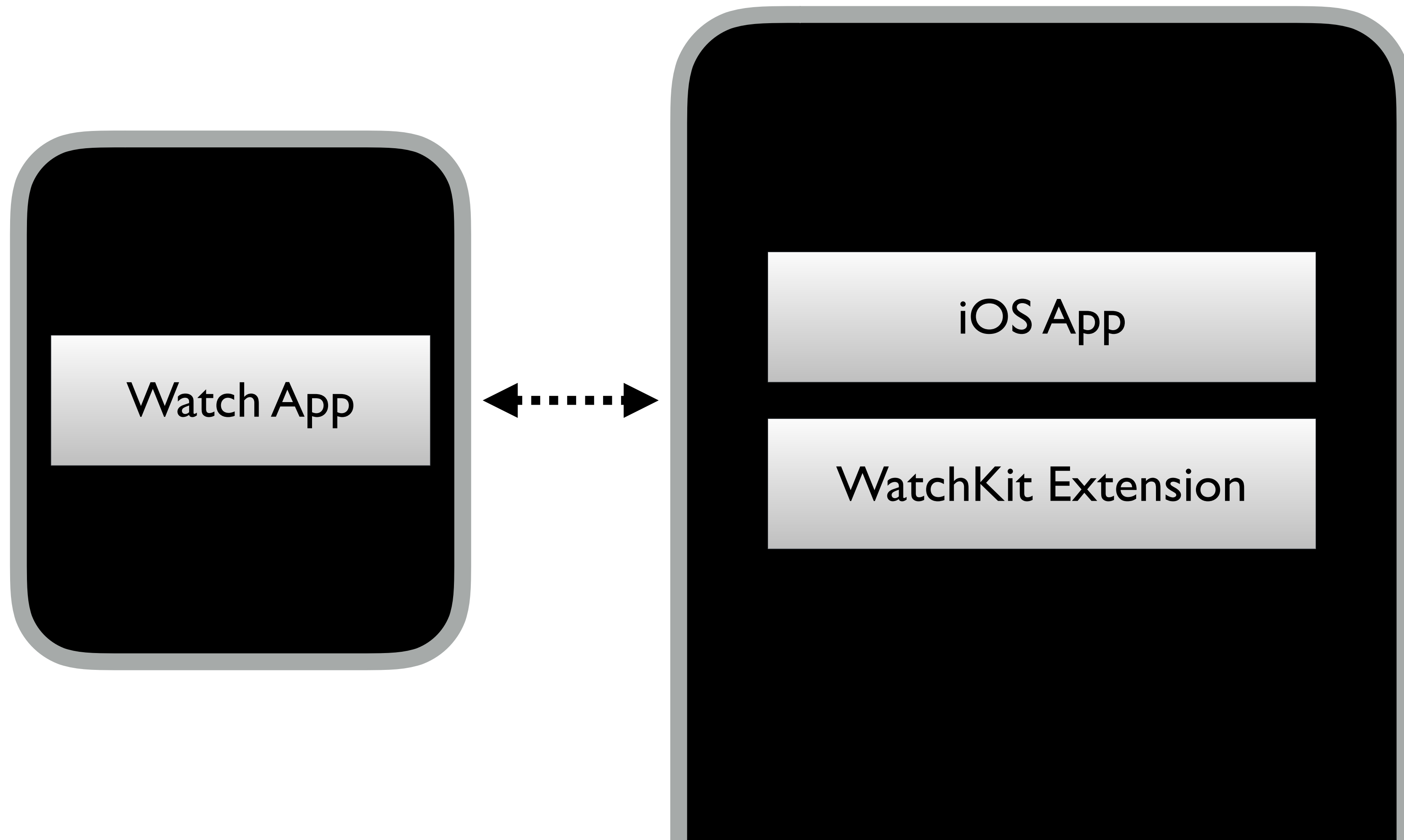
3.

Glances

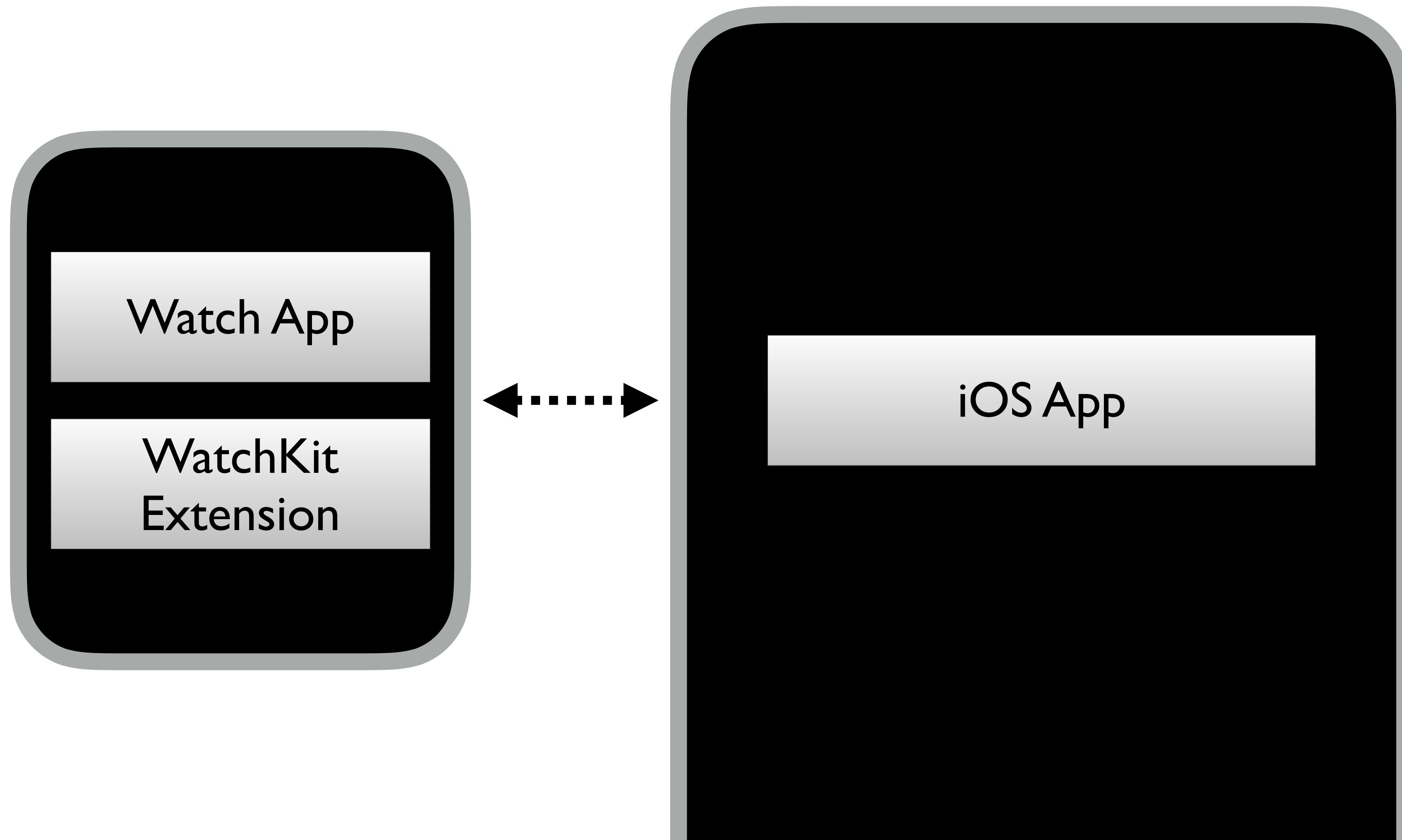
Apps sind Landingpages für Erweiterungen



WatchKit Extension



WatchKit Extension

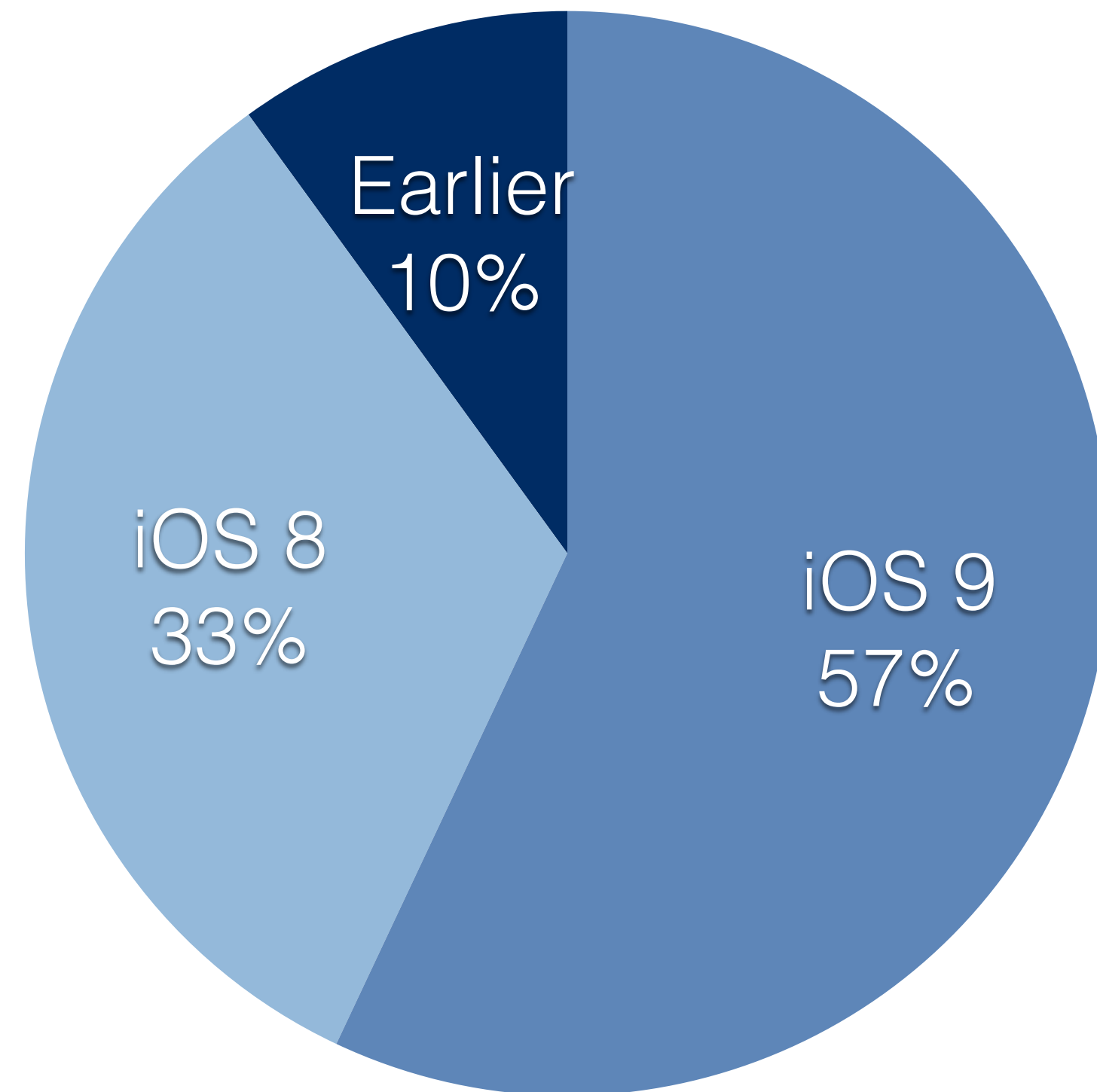


Demo

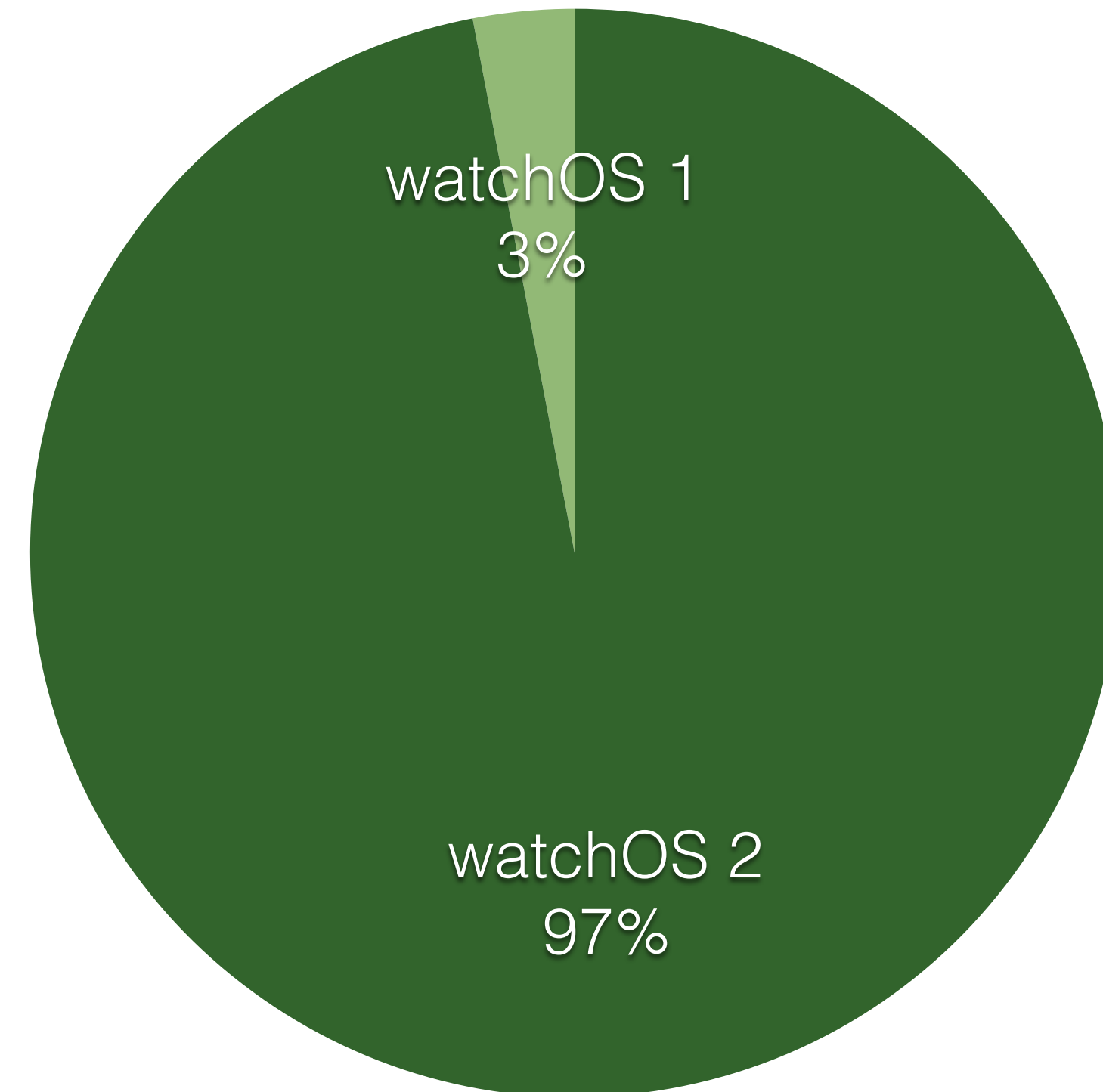
watchOS 2

- Extension wird auf der Uhr ausgeführt
- Animationen
- Erweiterter Zugriff auf Hardware
- Complications (ClockKit)
- Uhr kann direkt auf Wi-Fi zugreifen
- Watch Connectivity Framework

Verbreitung



<https://developer.apple.com/support/app-store/>



<http://www.wristly.co>

Complications



Complication Families

Modular Small



Modular Large

Circular Small

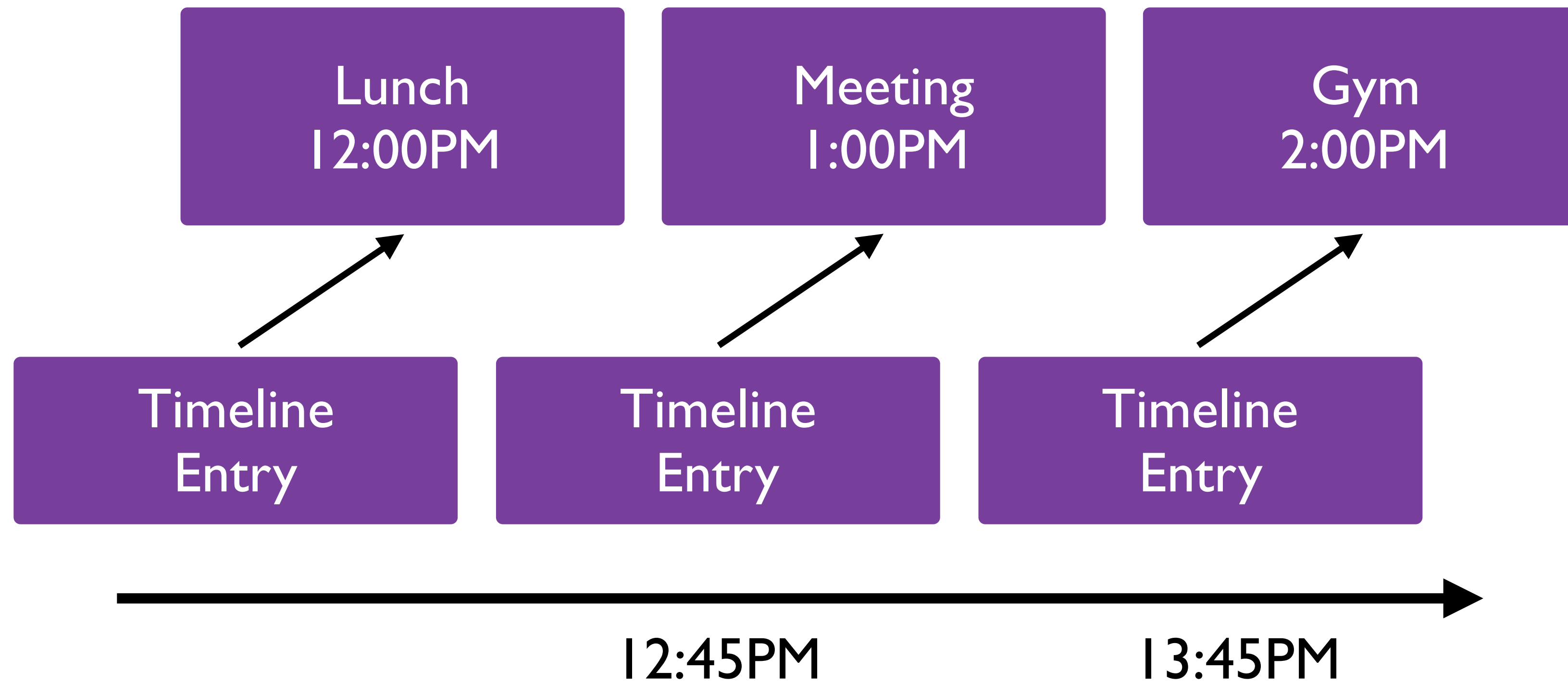


Utilitarian Small



Utilitarian Large

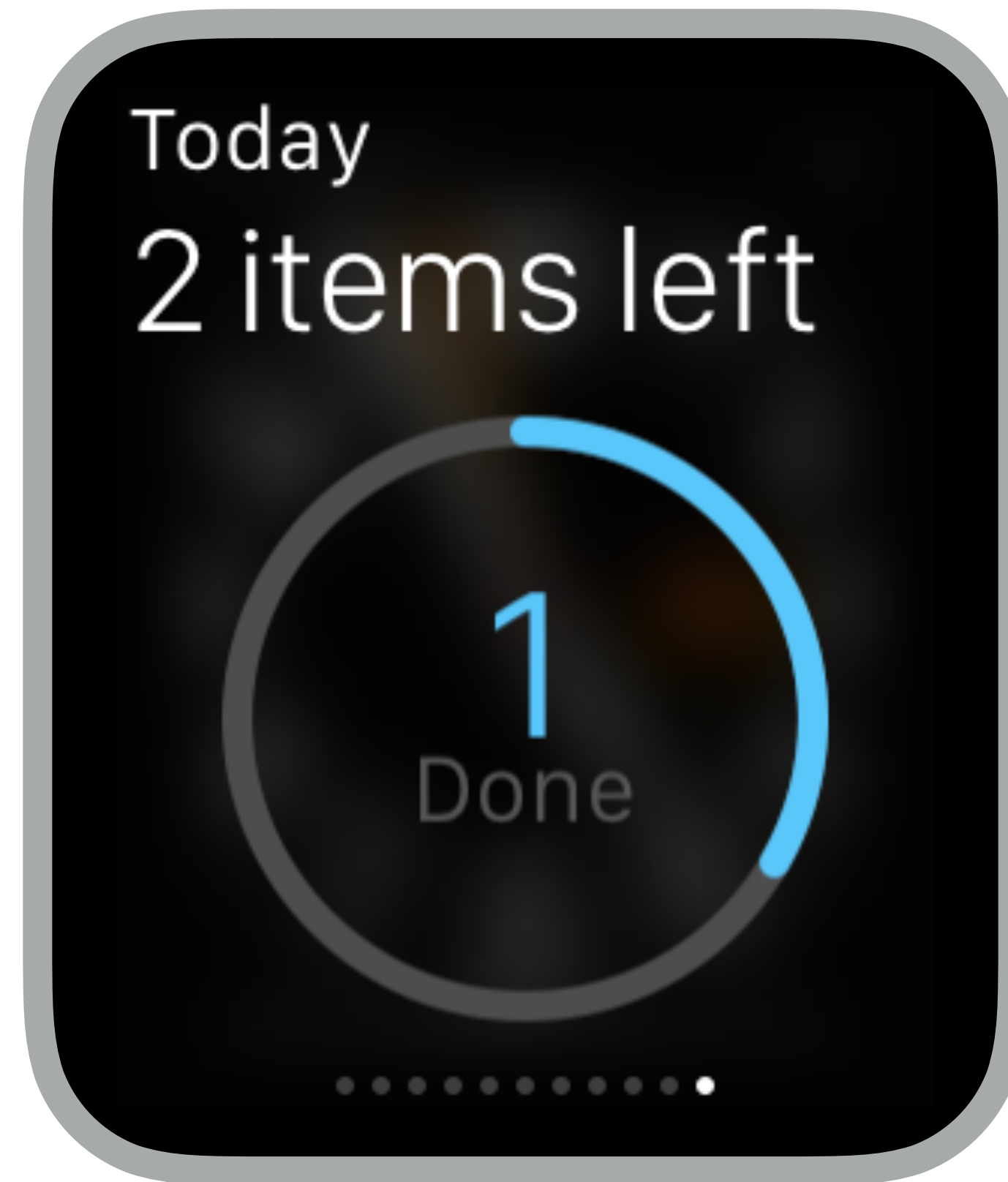
Time Travel



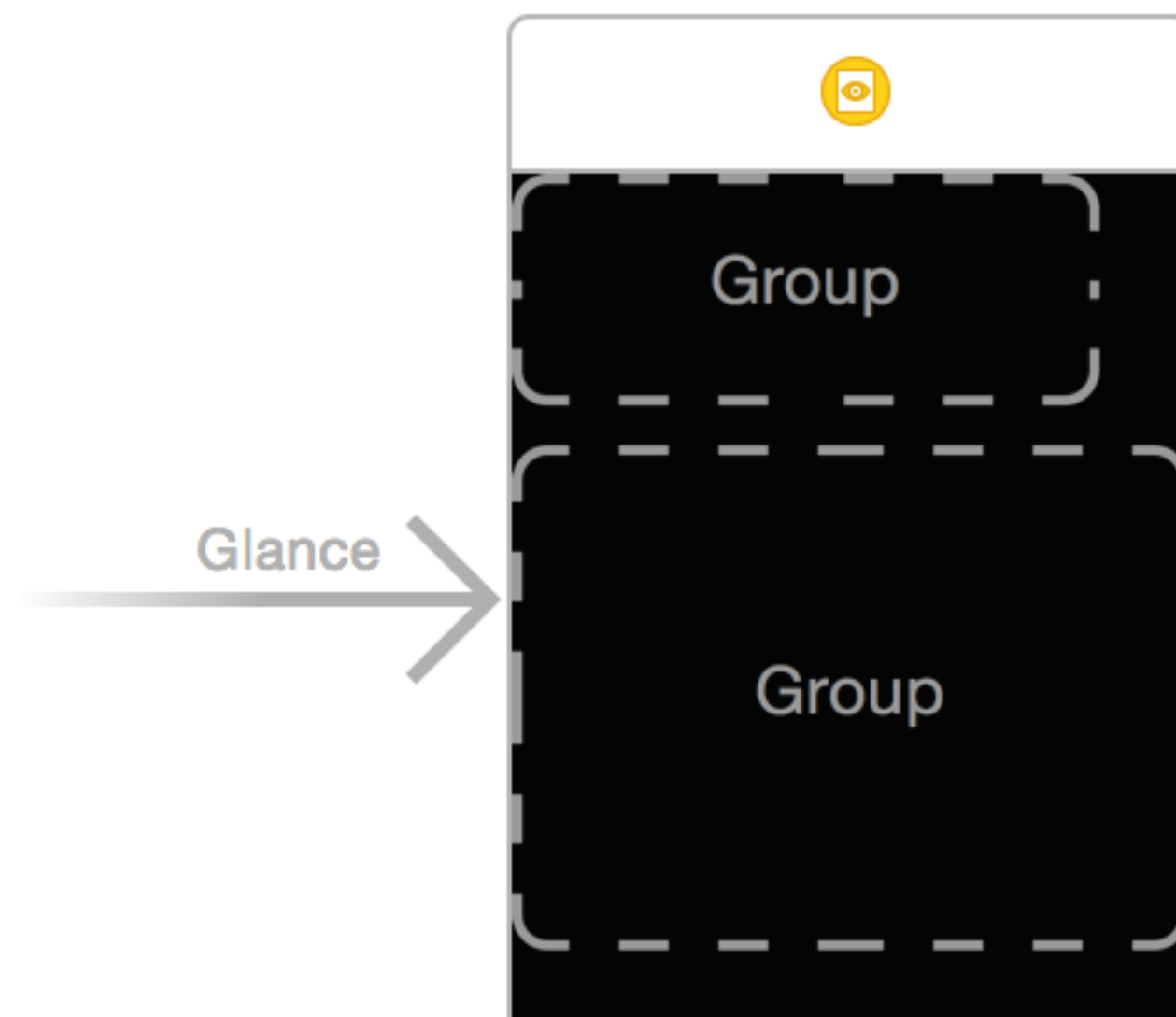
Implementierung Complications

- Wenig Platz
- Data Update
- PushKit
- Timed update
- Budgets für Ausführung

Glances (Checks)



Glance Templates



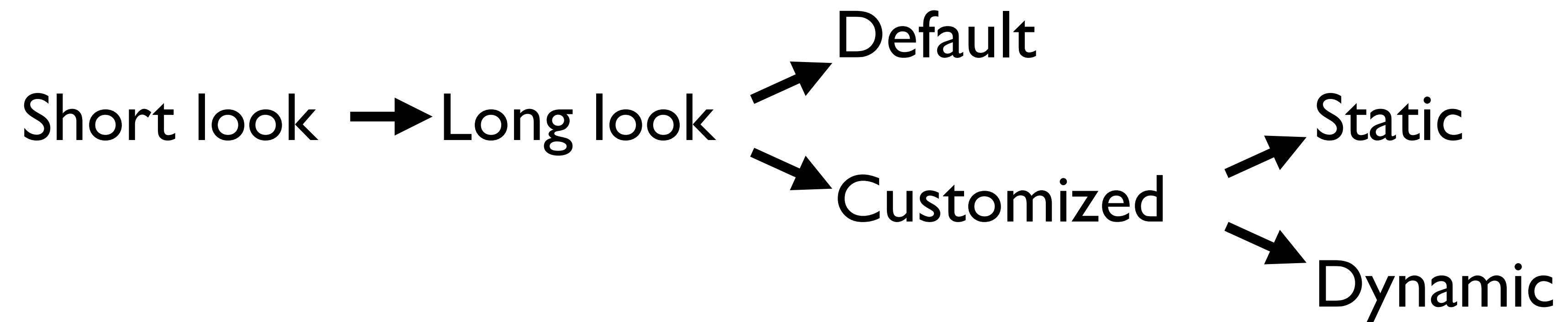
Implementierung Glances

- Nicht interaktiv
- UI basiert auf Templates
- Handoff um Daten an WatchKit App zu übergeben
- Netzwerkzugriffe profitieren von `performExpiringActivityWithReason:usingBlock:`

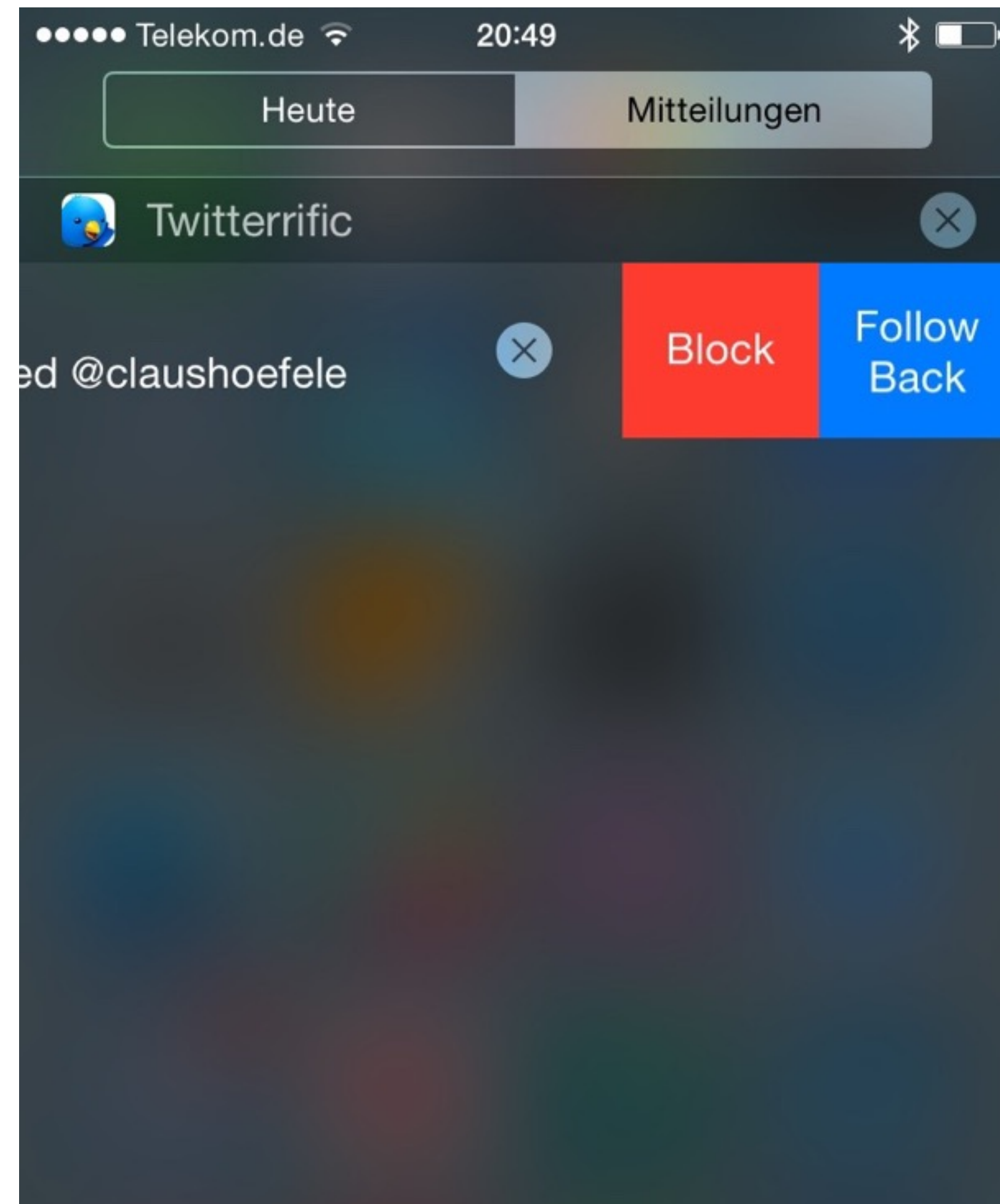
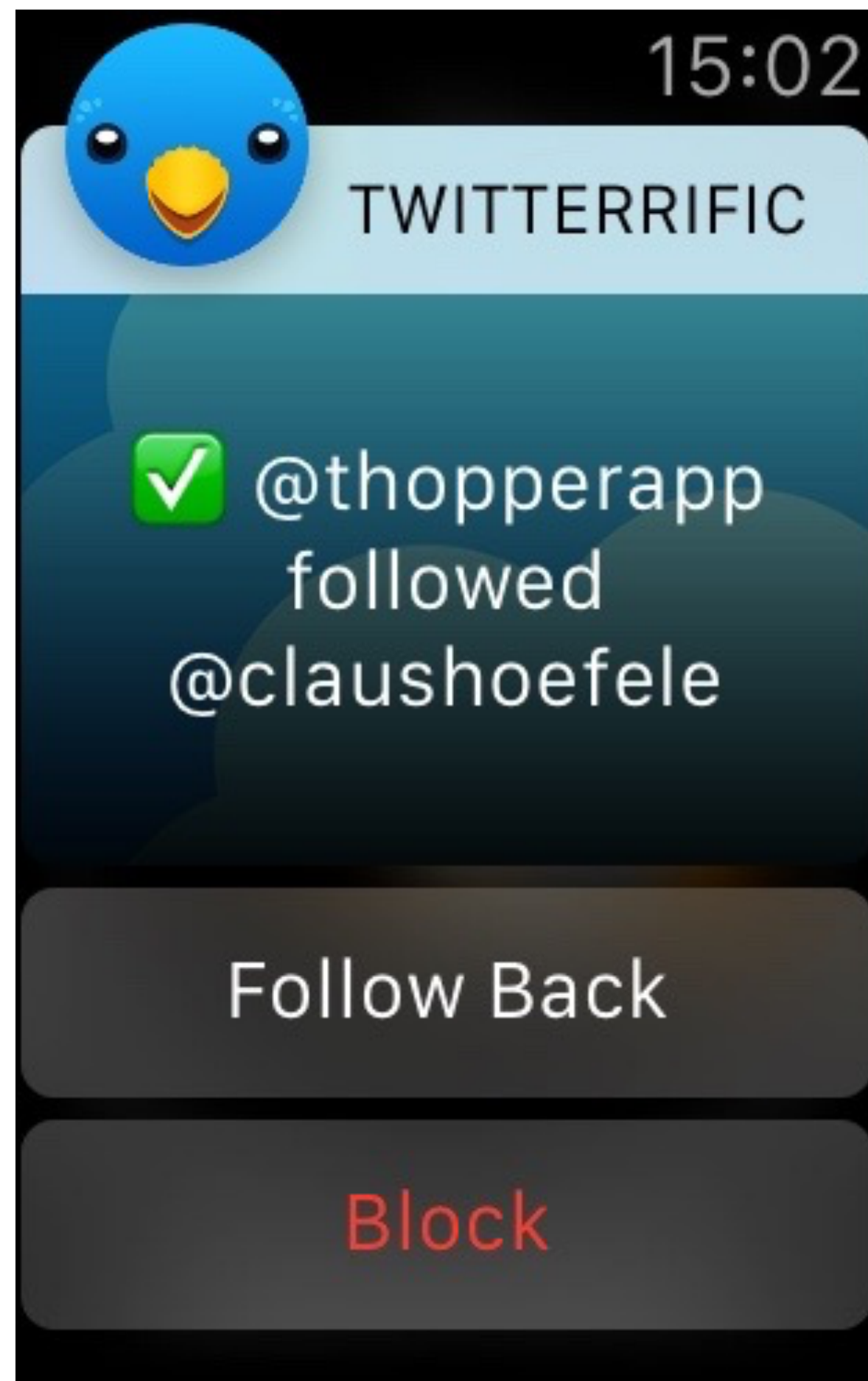
Notifications



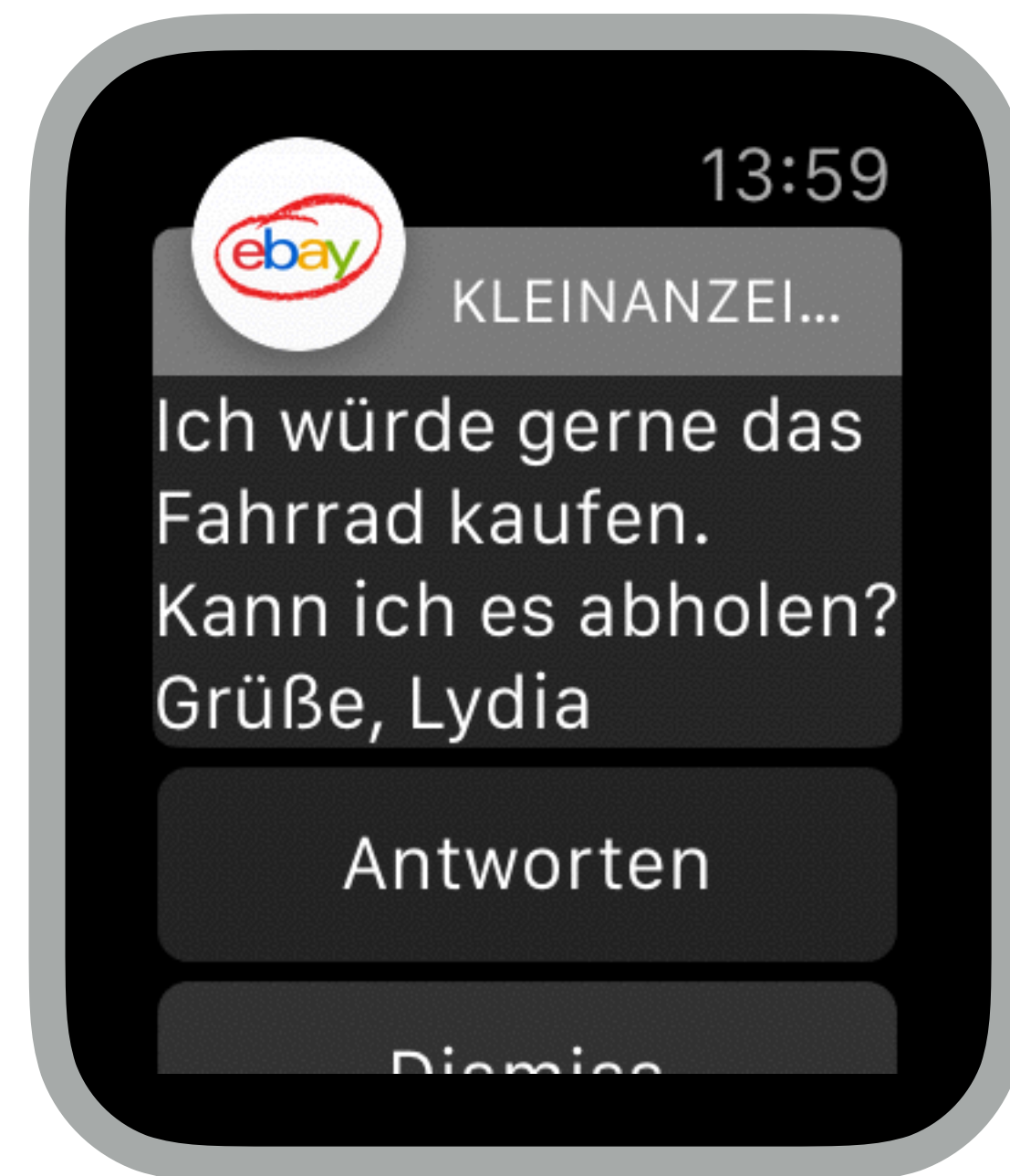
Notification Appearances



Interactive Notifications

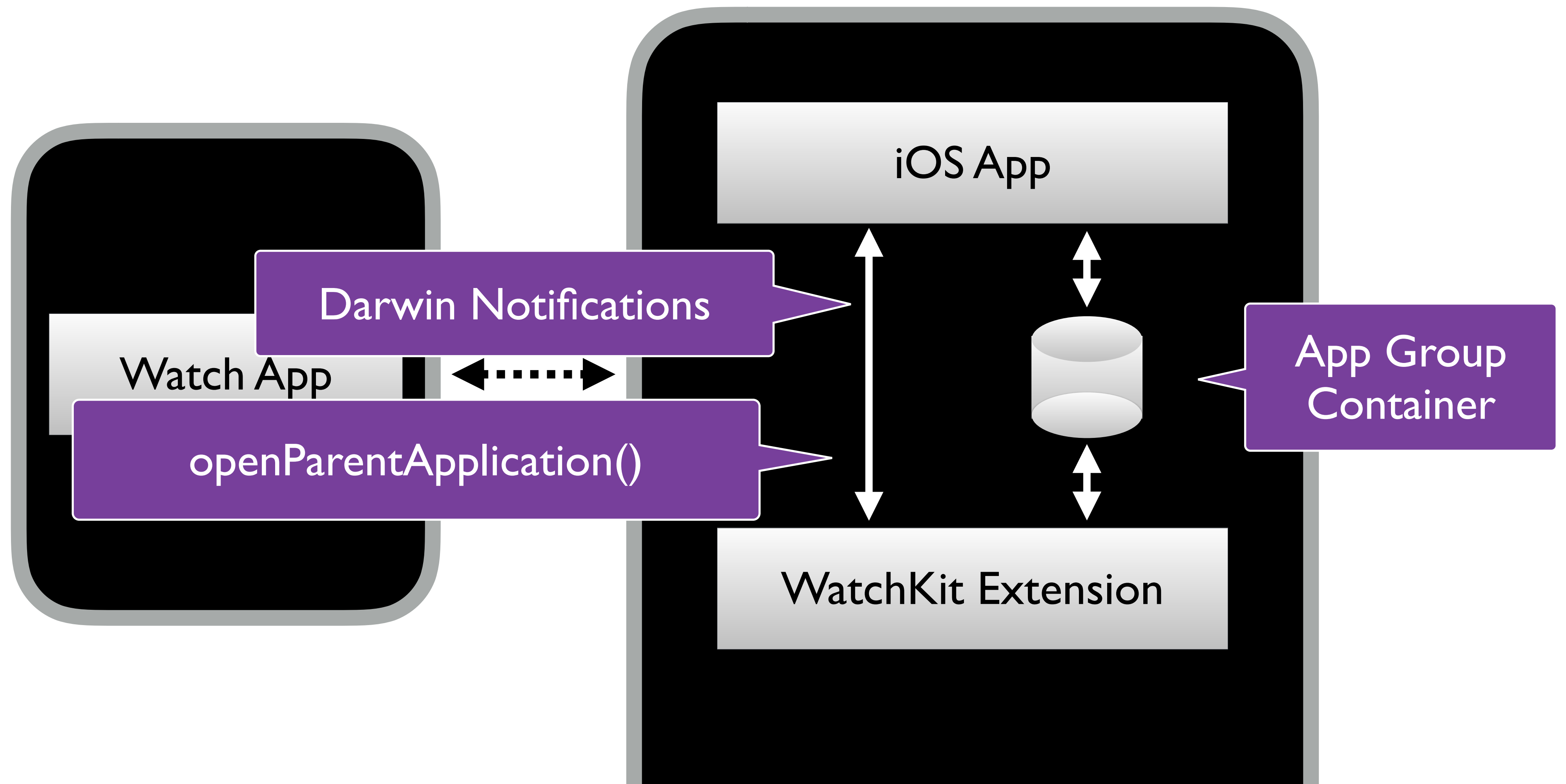


Foreground Actions

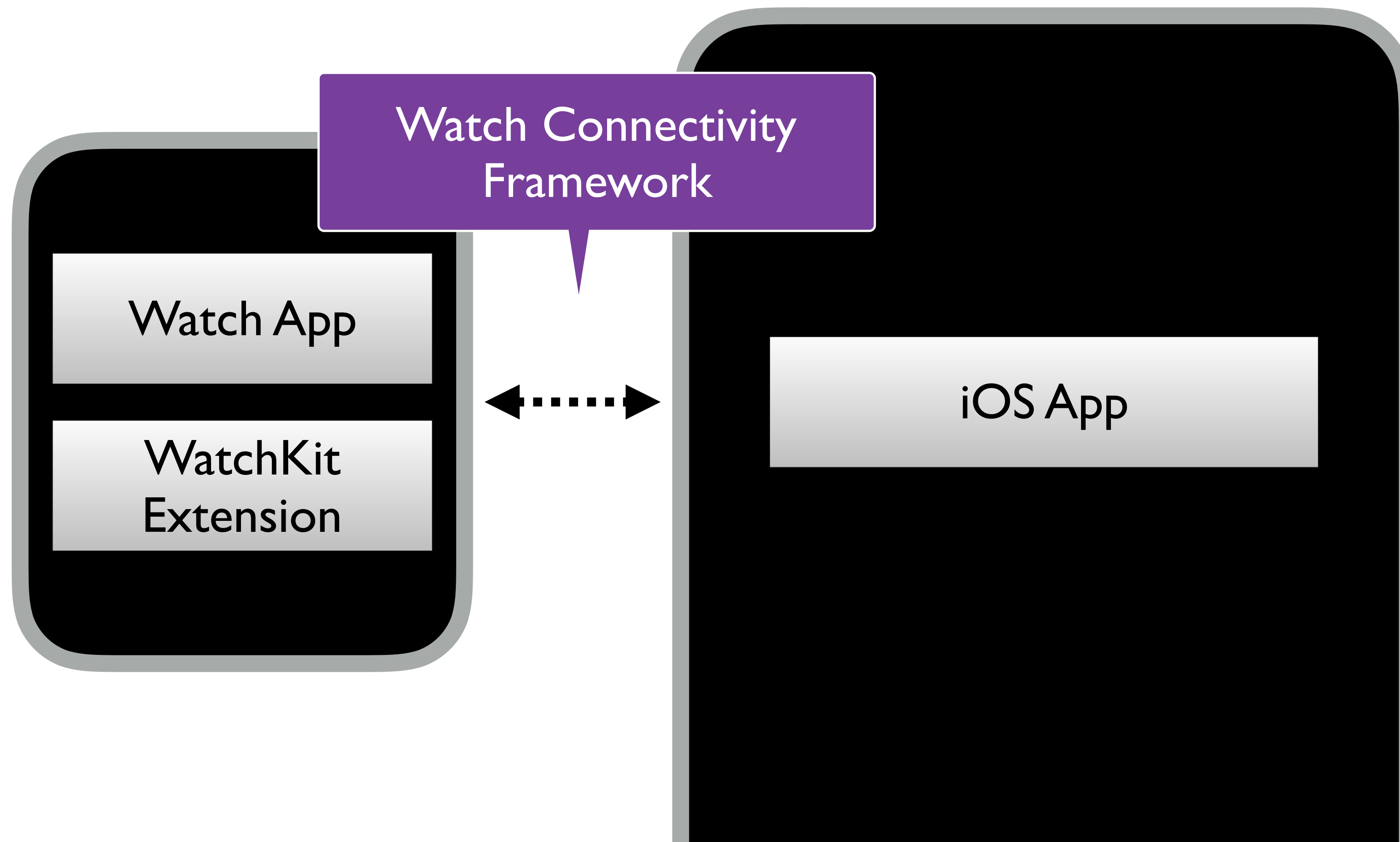


Demo

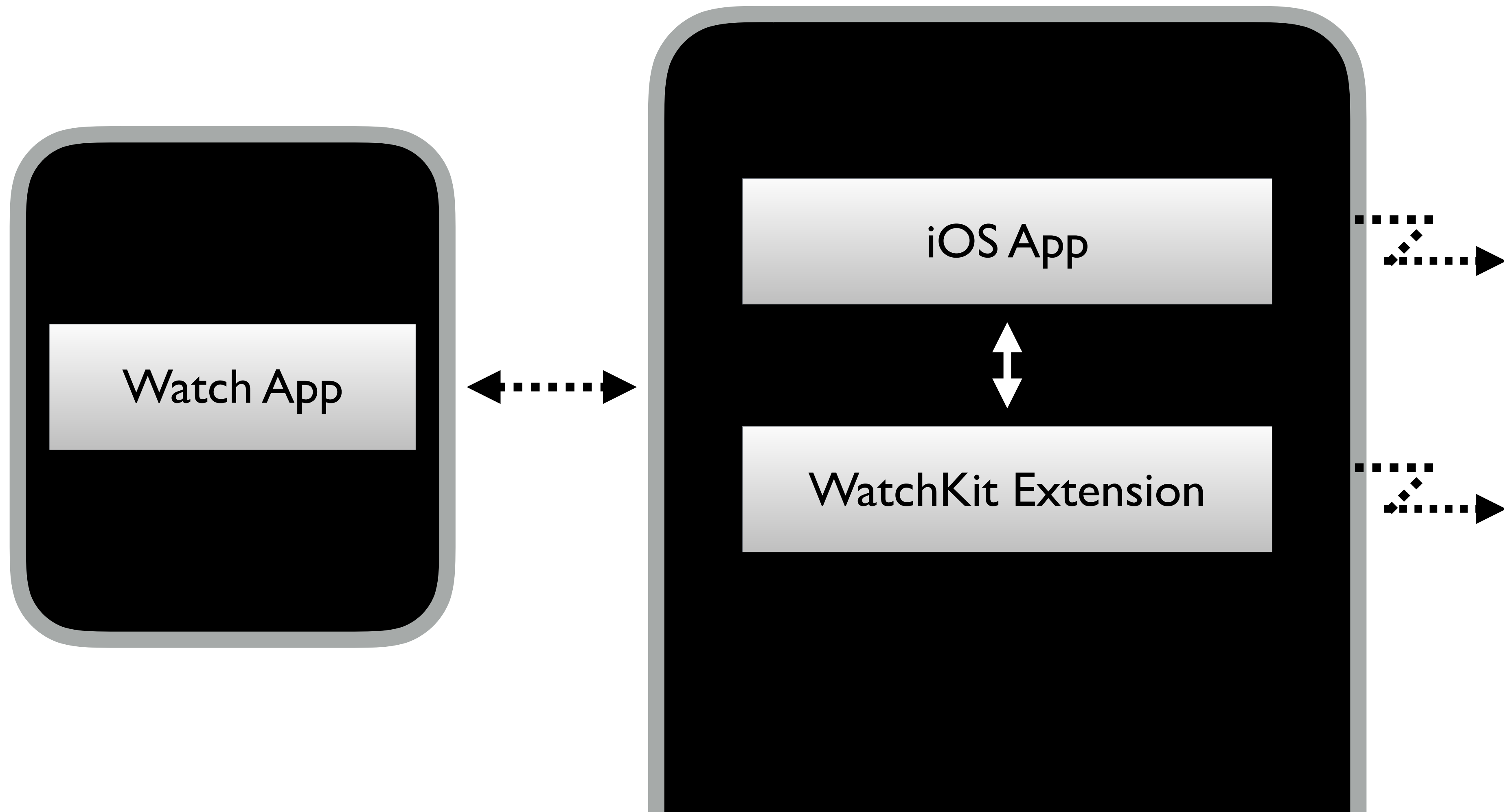
Parent App



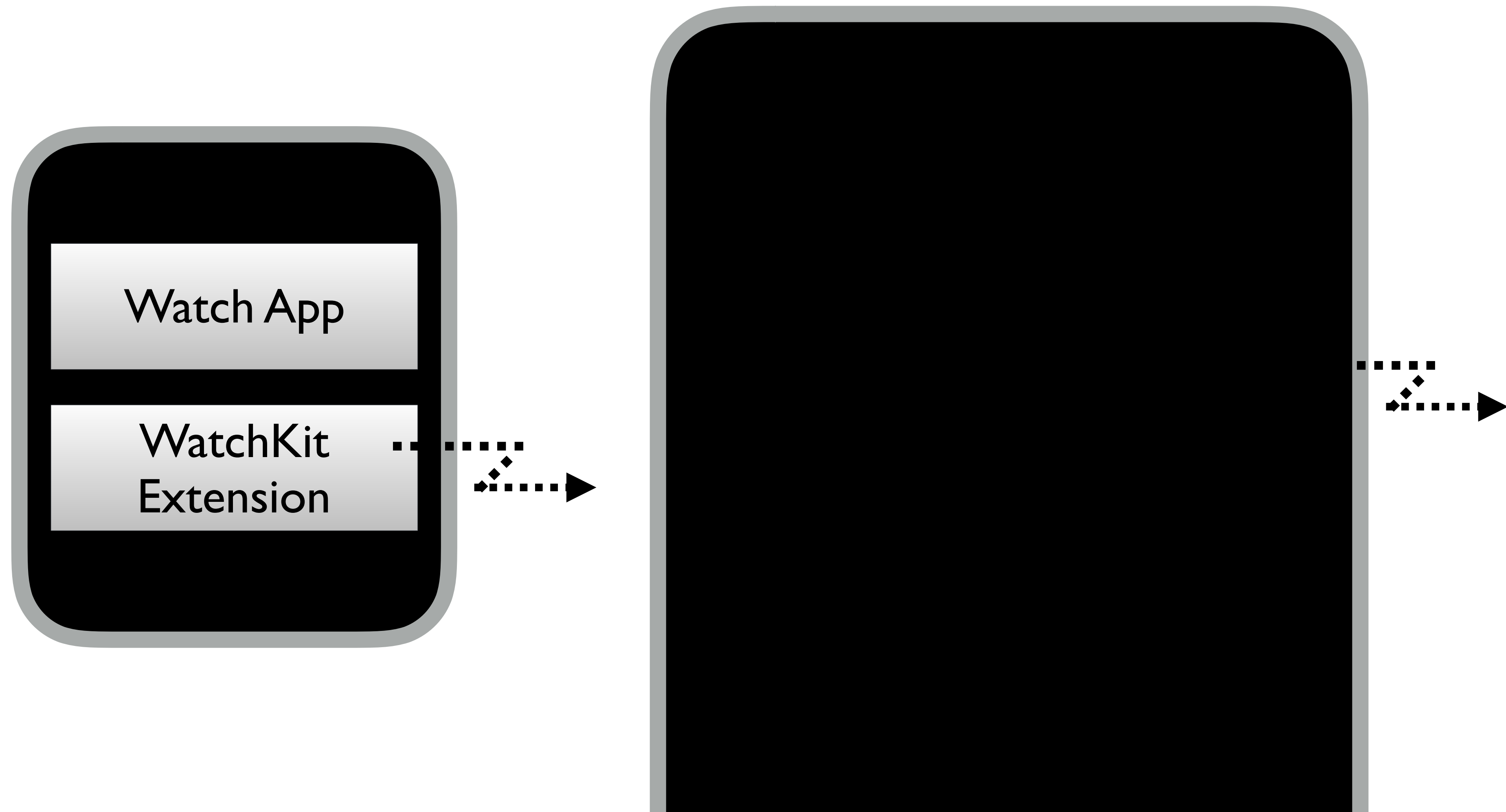
Parent App



Networking



Networking



Demo

Watch Connectivity

- Background mode
 - Application context
 - User Info Transfer
 - File Transfer
- Interactive messaging
 - Dictionary
 - Data

Networking

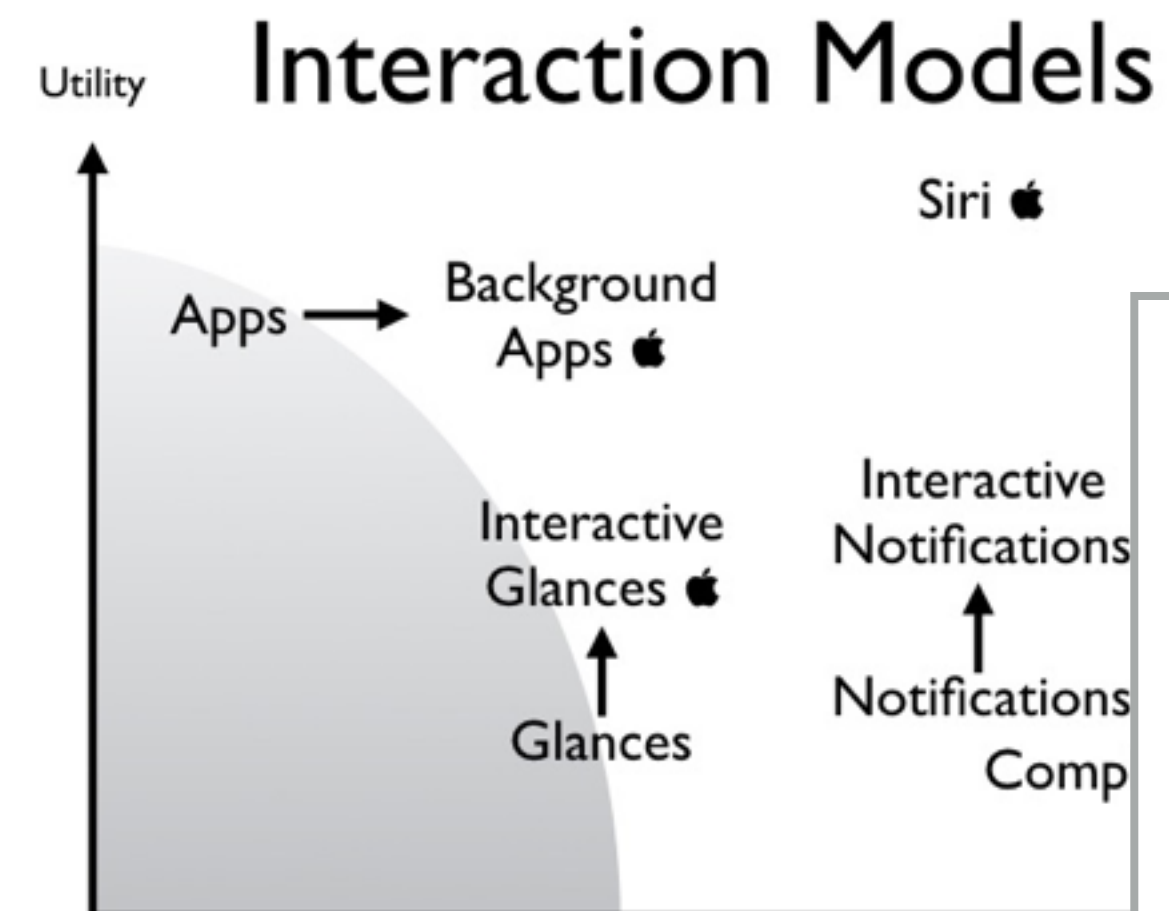
- Unterstützung für HTTP/S (2) & ATS
- App-Laufzeit sehr kurz
- Request schnell halten
- Background Uploads/Downloads
- Background Task Assertion
- Keine NSURLConnection

Komplexität



@claushoefe

<https://github.com/choefe/macoun-2015>



Prioritäten

1. Interactive Notifications
2. Complications
3. Glances

Apps sind Landingpages für

Komplexität



CC BY 2.0 Natalie Lucier <https://flic.kr/p/6weBij>