

# **Konzeption und prototypische Umsetzung einer Home Automation-Applikation für iOS-Geräte**

Eine Bachelorarbeit von  
Felizia Bernutz

Prof. Dr. Hans-Georg Hopf

Prof. Dr. Matthias Hopf

Steffen Blümm (adorsys)

# Übersicht

1. Motivation
2. Grundlagen der Home Automation
3. Produktvision
4. Klick-Dummy
5. Umsetzung
6. Ausblick

# Motivation

## Ausgangssituation

Nach Feierabend auf Sofa, aber in Küche brennt noch Licht

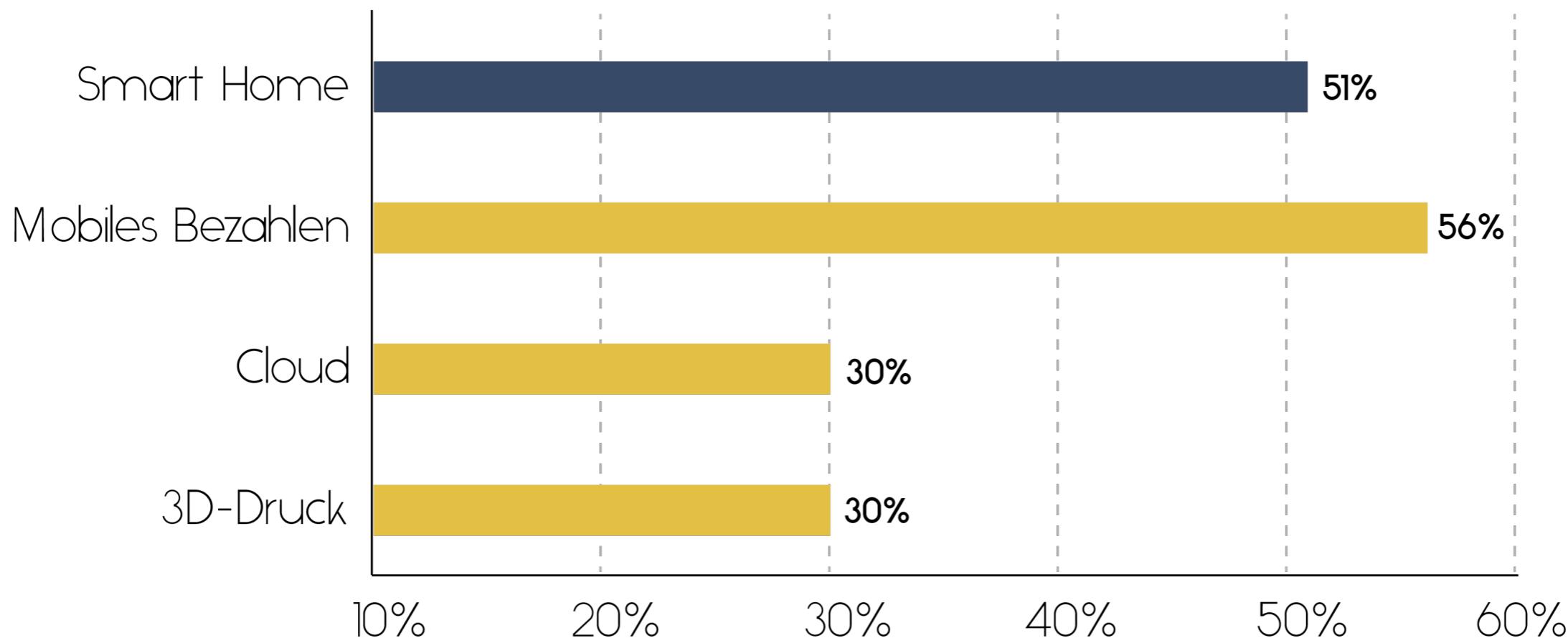
## Ausgangssituation

Nach Feierabend auf Sofa, aber in Küche brennt noch Licht

## Ziel

- Steuerung des Lichtes durch **Stimme** und durch **Smartphone**
- Unterstützung im Alltag

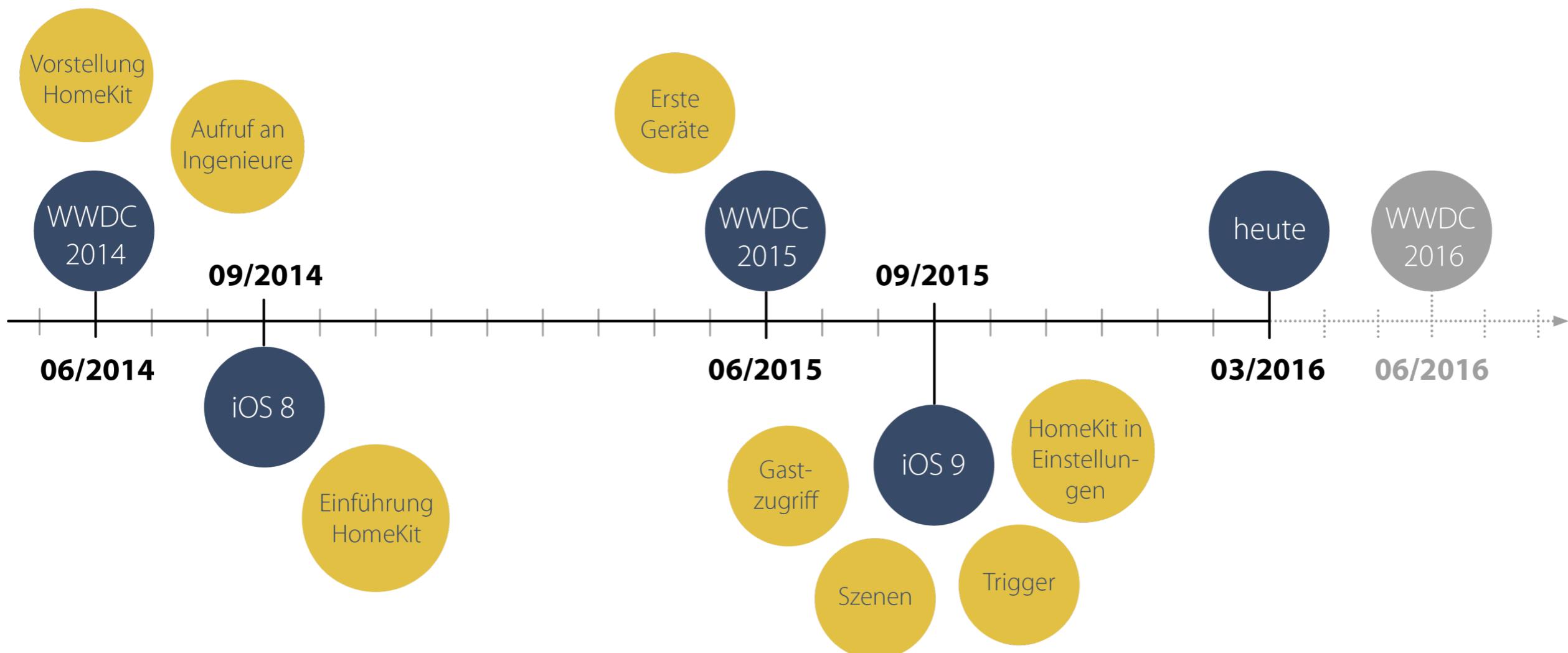
# Technologien mit großem Einfluss



Quelle: Eigendarstellung mit Daten von GfK (11/2015)

# Grundlagen der Home Automation

# HomeKit-Entstehung



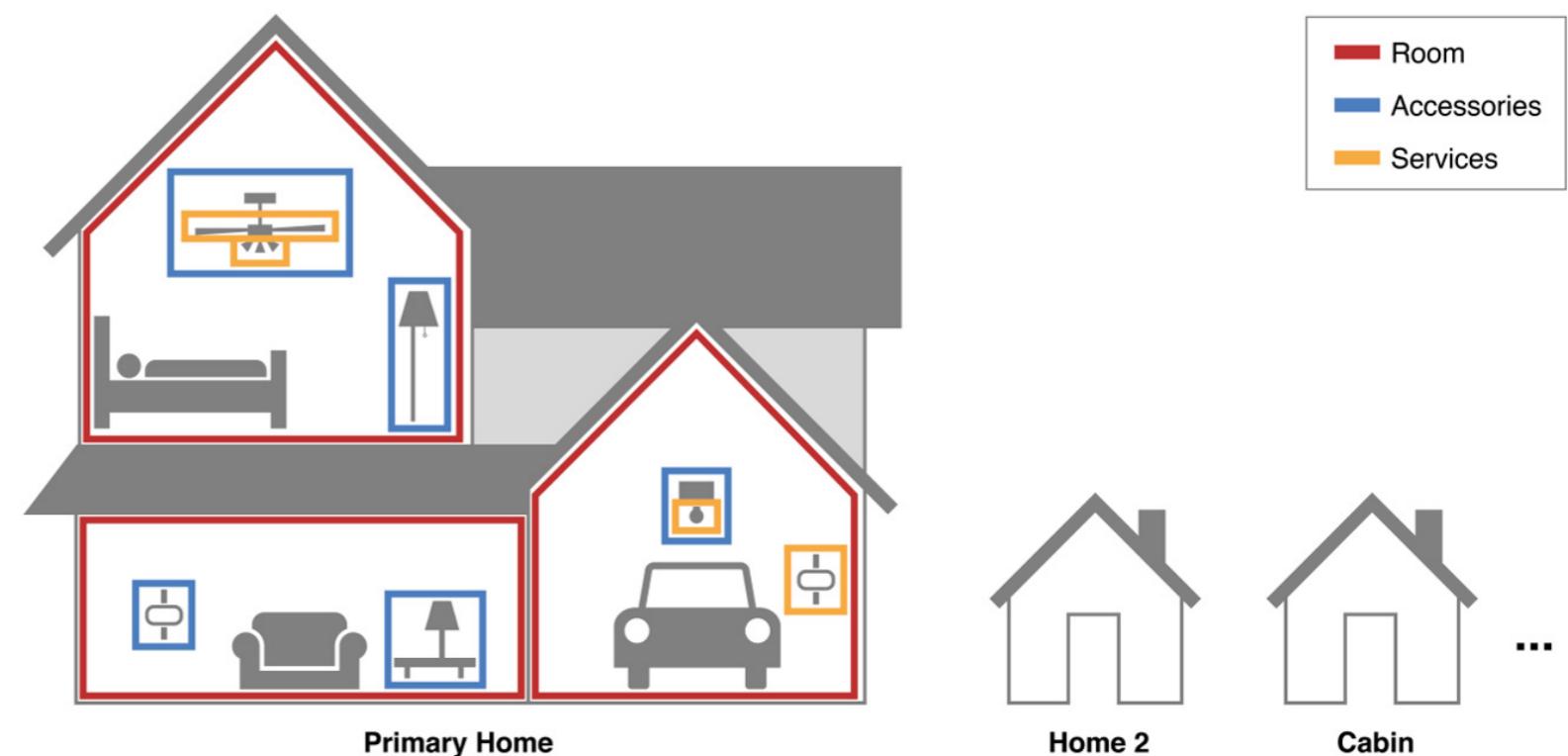
Quelle: Eigendarstellung

# Möglichkeiten

- Common Database
- Sprachsteuerung mit Siri
- Steuerung von Unterwegs
- Action Sets und Trigger

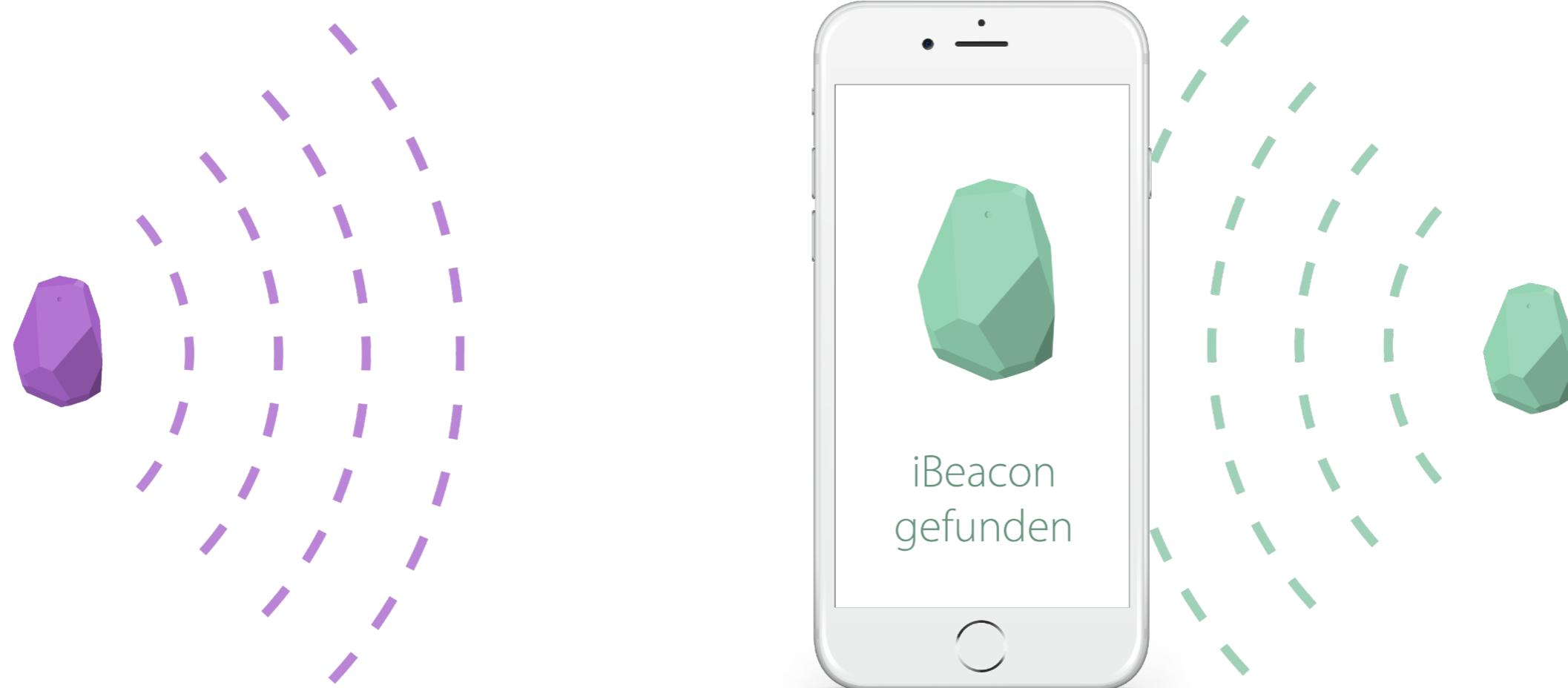
# Struktur

- Homes
- Rooms
- Accessories
- Services
- Characteristics
- Zones



Quelle: Apple Developer

# iBeacons



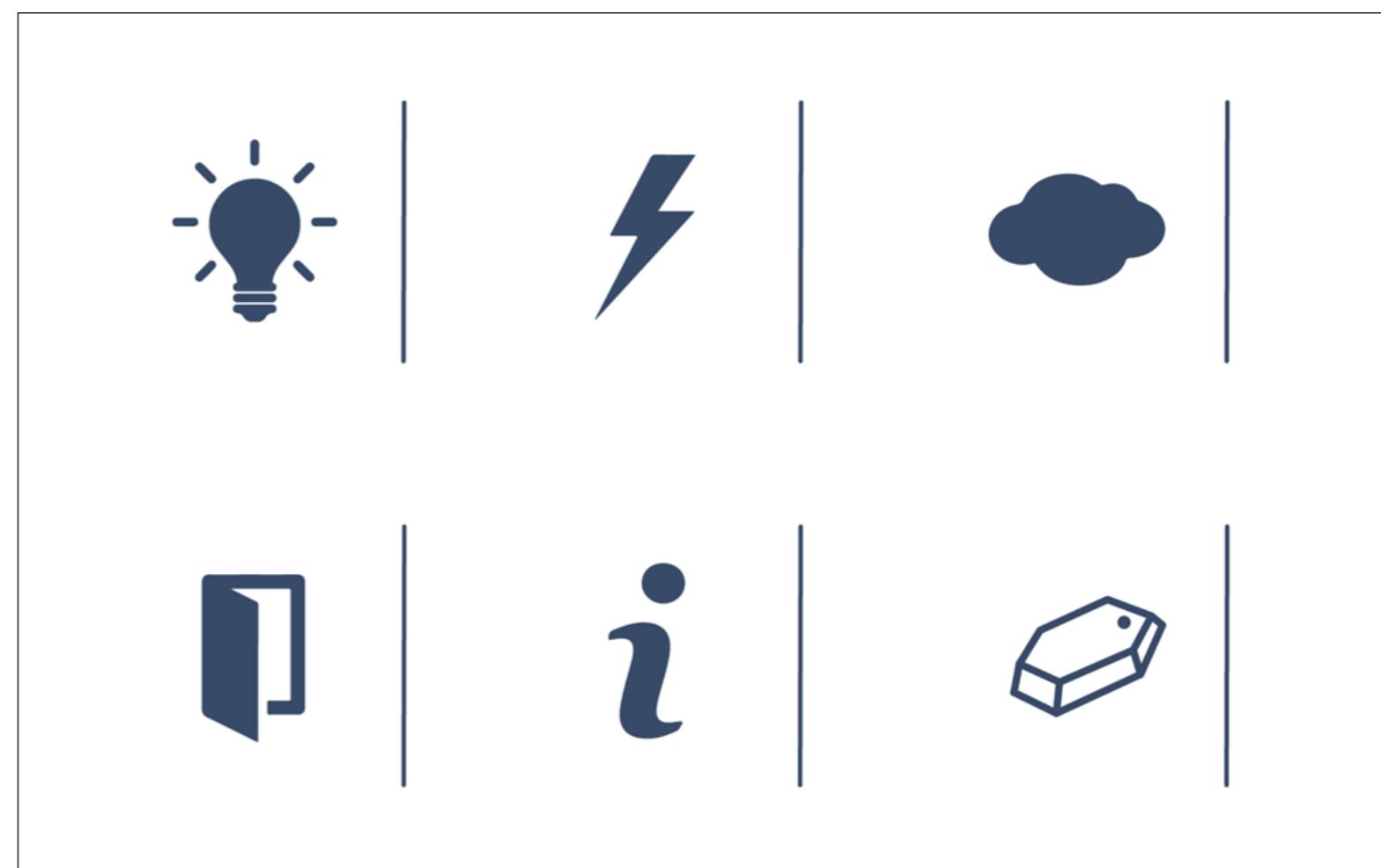
Quelle: Eigendarstellung mit Grafik von [Sketch App Sources](#)

# Produktvision

# Zielgruppe

- Besitzer eines iPhones mit iOS 9.0 oder neuer
- Zwei Gruppen:
  - technikbegeistert, offen, Vertrauen in neue Technologien
  - hilfsbedürftige Menschen
- vermögend

# Icons



Quelle: Eigendarstellungen

# Usability

- einfach und selbsterklärend
- Individualität
- Schnellzugriffe



Quelle: Eigendarstellung

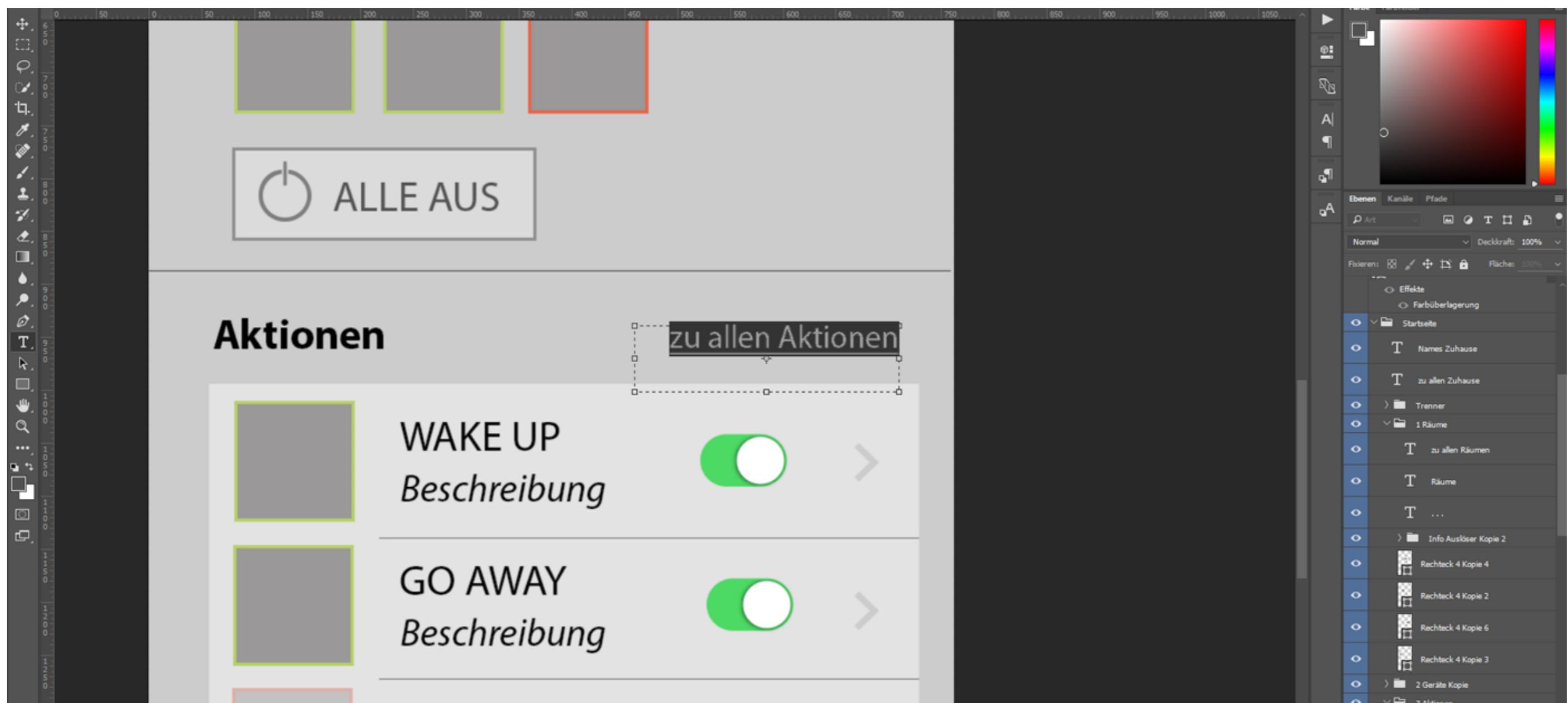
# **Klick-Dummy**

# Skizzen



Quelle: Eigendarstellung

# Photoshop



Quelle: Screenshot

# Marvel

Hotspot Destination workflow\_37.png

Hotspot Destination	Screen Transition	Action
workflow_37.png	None	Click or tap

Link to last visited

Link to URL or embed

Scroll to anchor

Keep scroll position

Quelle: Screenshot

# Präsentation – Klick-Dummy

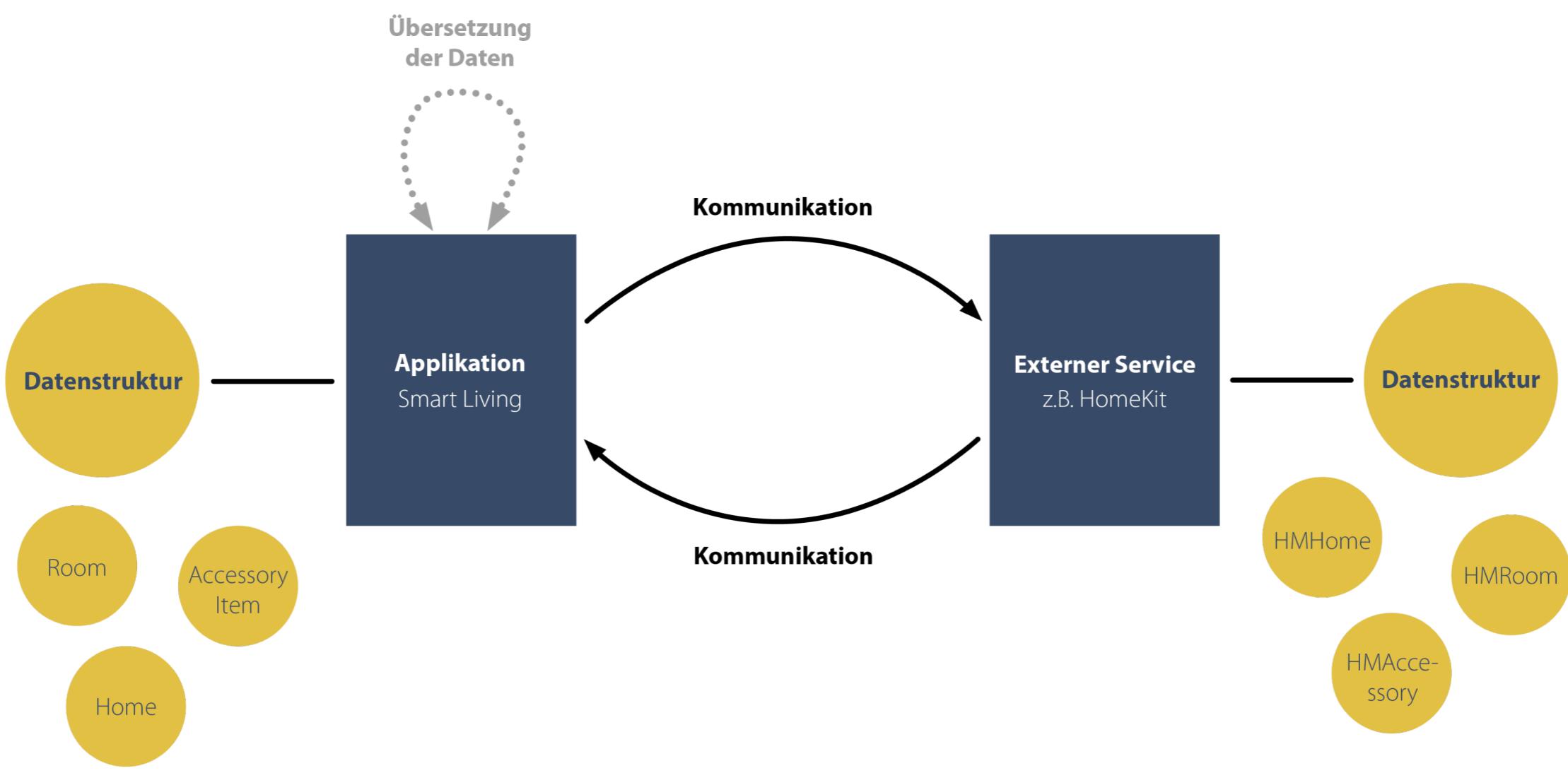
...

# Vorschläge

- Feedback von mehreren Personen
  - Reduzieren der Seiten
  - Zeitangabe für Bearbeitungsdauer
  - Fokus auf Geräte und Räume legen

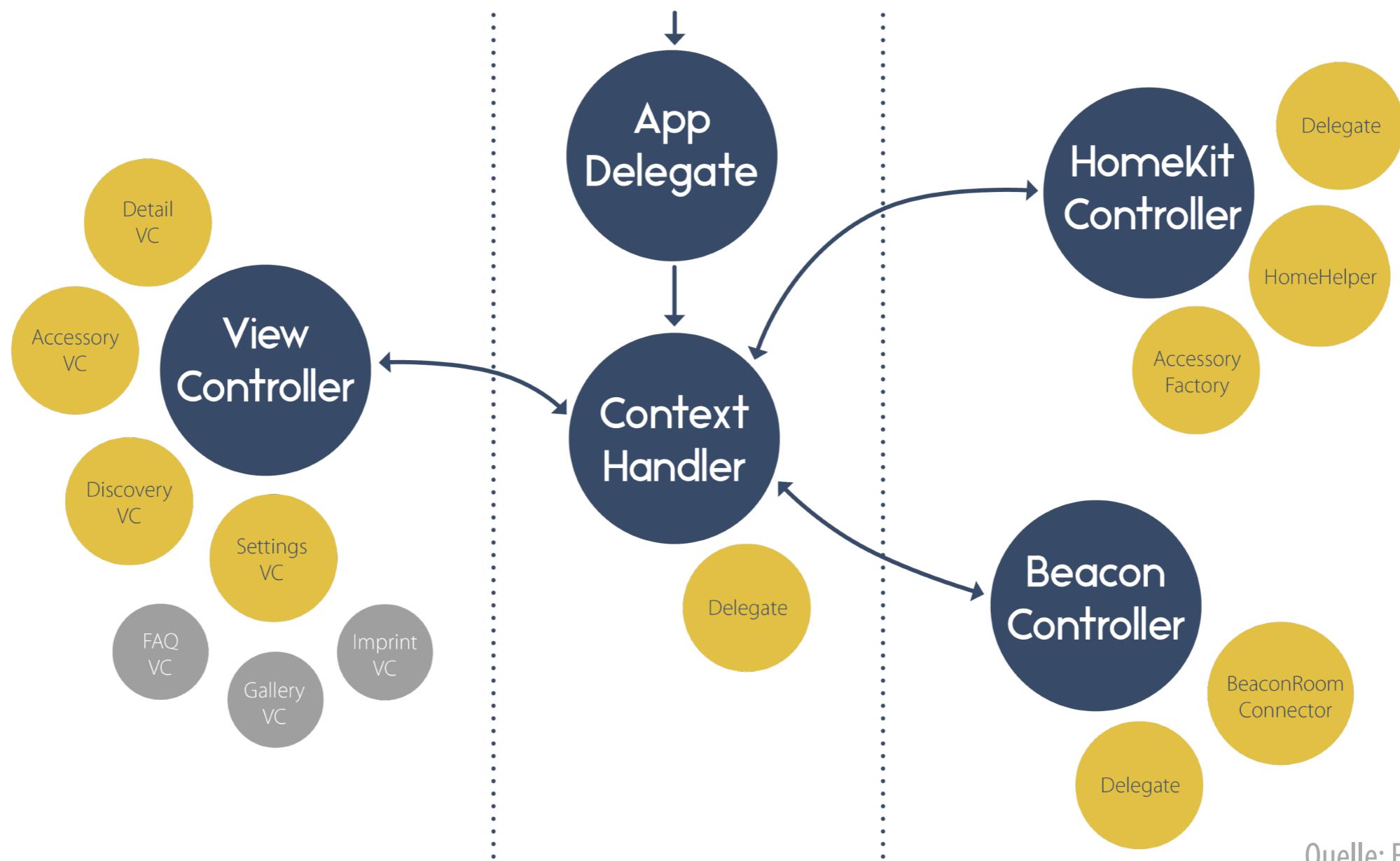
# Umsetzung

# Trennung der Datenstrukturen



Quelle: Eigendarstellung

# Aufbau der Klassen-Struktur

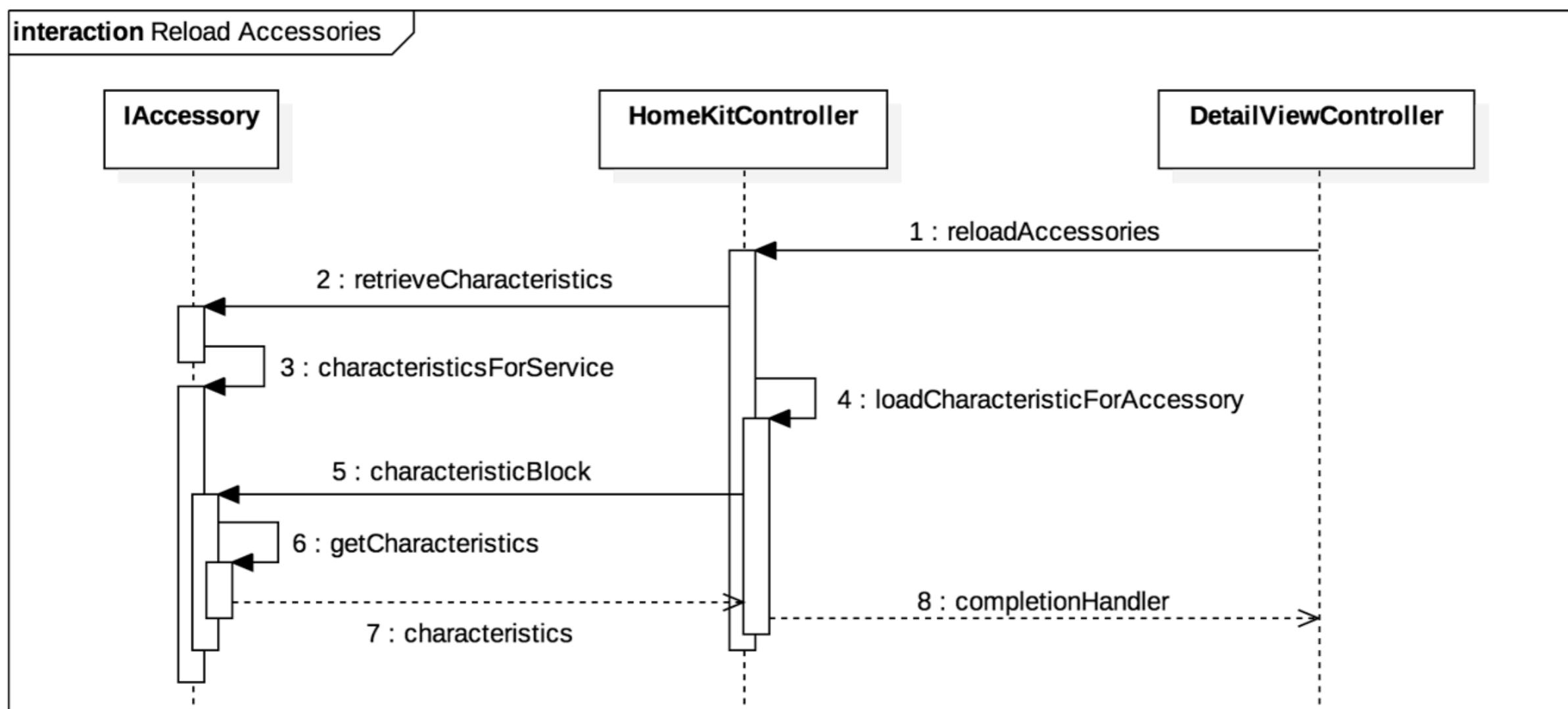


Quelle: Eigendarstellung

# HomeKit Service

1. Laden der HomeKit Daten
2. Umwandeln in interne Datenstruktur
- 3. Laden der Accessories**
4. Laden der Characteristics
5. Steuern der Accessories

# Reload Accessories



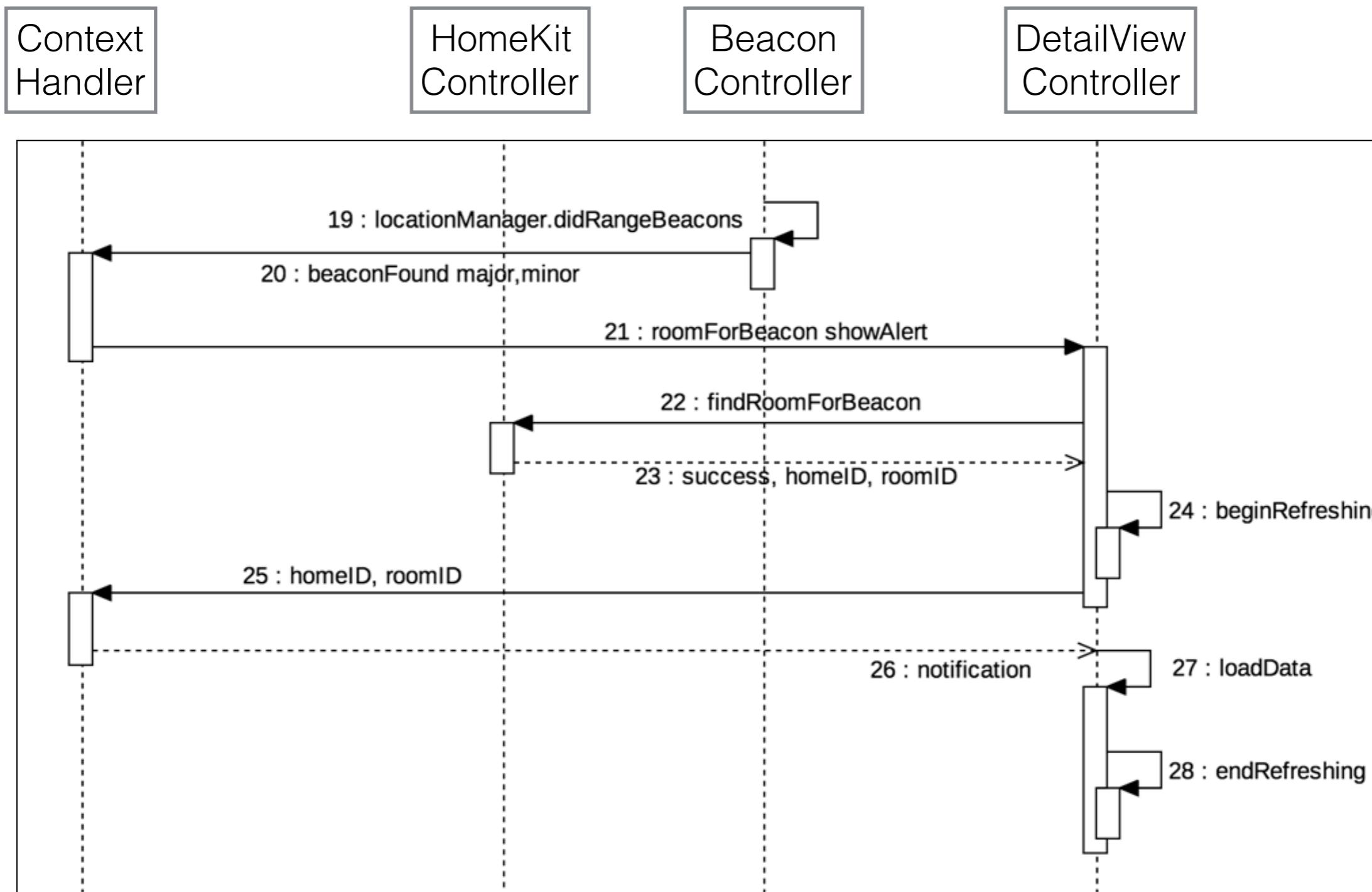
Quelle: Eigendarstellung

# iBeacon Service

1. Autorisierung
2. Monitoring starten
- 3. Finden eines iBeacons**
4. Verbindung HomeKit und iBeacon

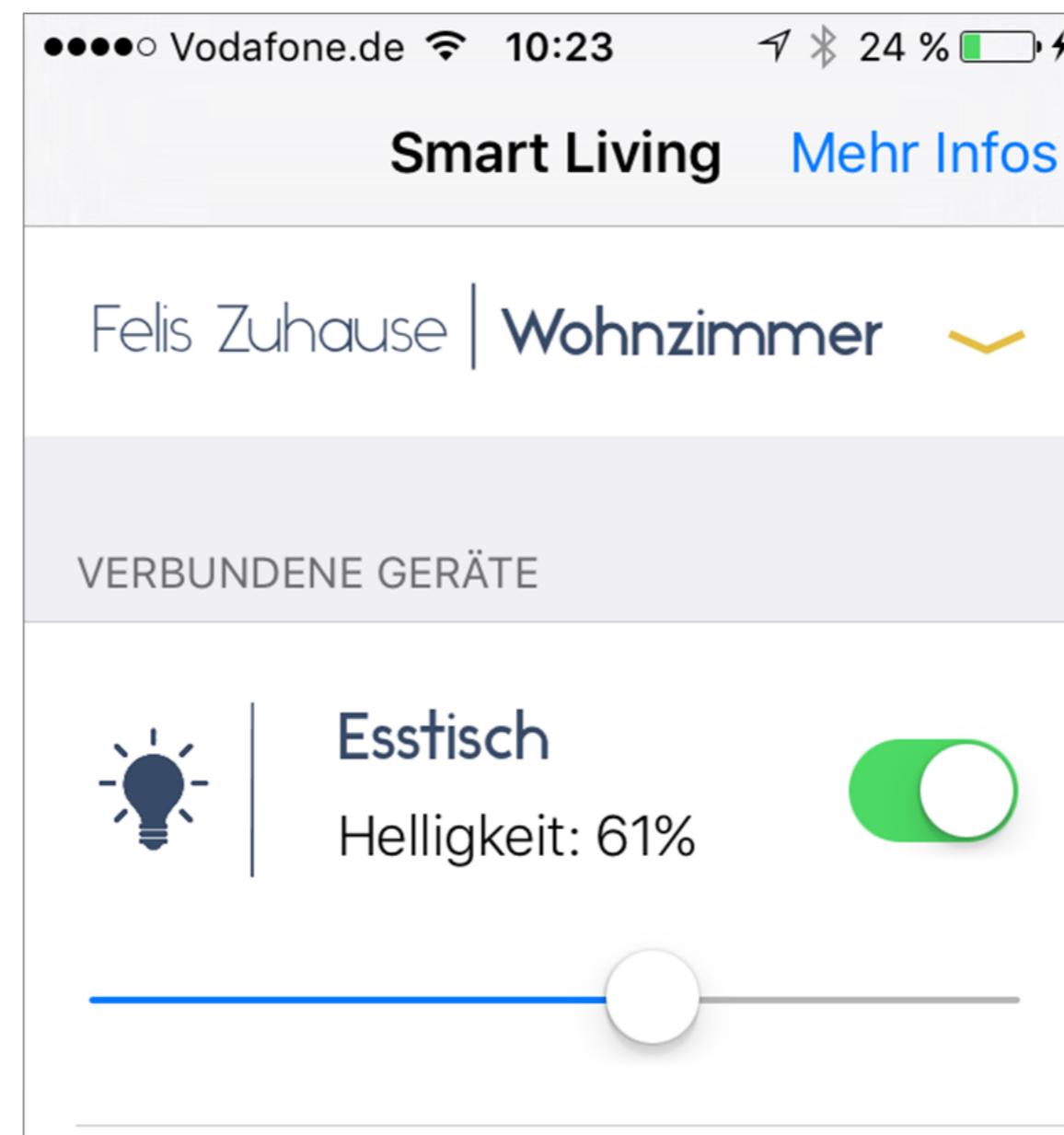
# Finden eines Beacons

```
(1) func locationManager(manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon],  
inRegion region: CLBeaconRegion) {  
(2)  
(3)    //1 filter ranged beacons  
(4)    let knownBeacons = beacons.filter{ $0.proximity != CLProximity.Unknown }  
(5)  
(6)    if !knownBeacons.isEmpty {  
(7)        //2 set closest beacon  
(8)        let closestBeacon = knownBeacons[0]  
(9)  
(10)       //3 check if last beacon is closest beacon  
(11)       if closestBeacon.major != lastBeacon?.major &&  
closestBeacon.minor != lastBeacon?.minor {  
(12)  
(13)           //4 delegate with information which beacon was found  
(14)           delegate?.beaconFound(self, major: closestBeacon.major.integerValue,  
minor: closestBeacon.minor.integerValue)  
(15)  
(16)           //5 set closestBeacon as last found beacon  
(17)           lastBeacon = closestBeacon  
(18)       }  
(19)   }  
(20) }
```



Quelle: Eigendarstellung

# Präsentation – App



# Funktionsumfang

- Auslesen der HomeKit Daten
- Hinzufügen, Auslesen und Steuern von Accessories
- iBeacon Erkennung und Reaktion
- Umsetzung des Designs

# Ausblick

- Action Sets
- Core Data: Speichern von Nutzereinstellungen
- Schnellzugriff: Mitteilungszentrale

**Vielen Dank für die  
Aufmerksamkeit**