

Muster im User Interface Reengineering in ein deklaratives User Interface Framework

Antrittsvortrag BA

Adrian Meister, 07.07.2022

HINWEIS

Persönliche Ziele:

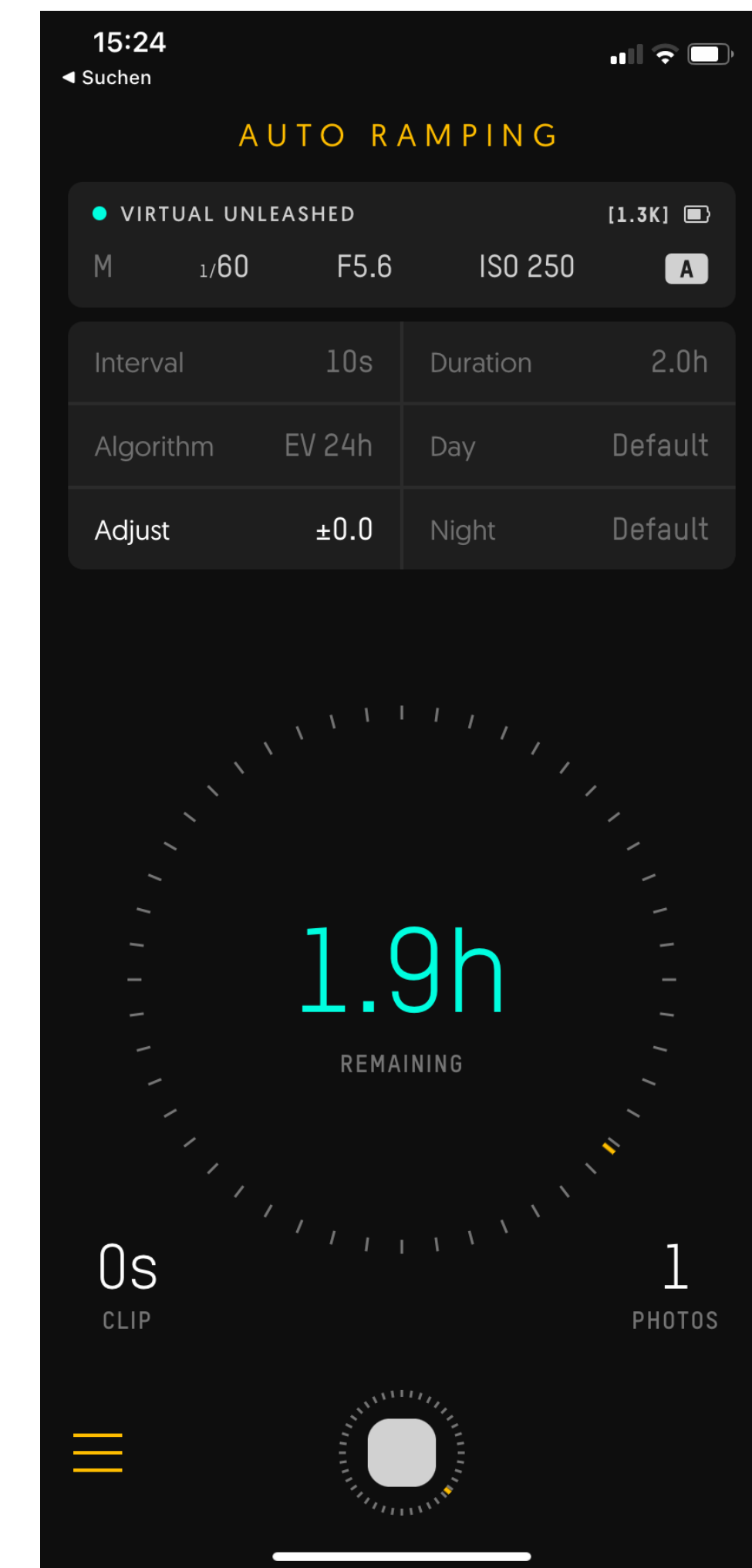
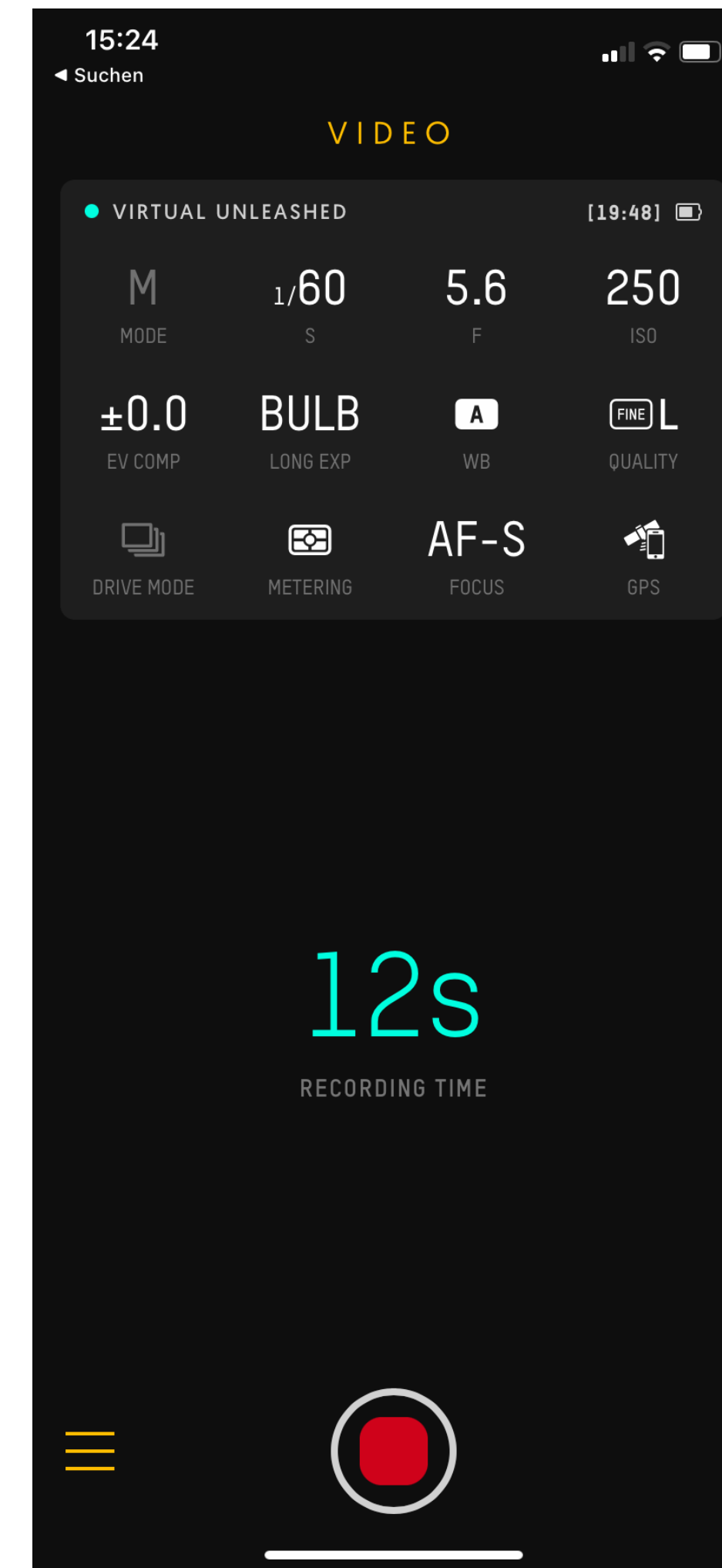
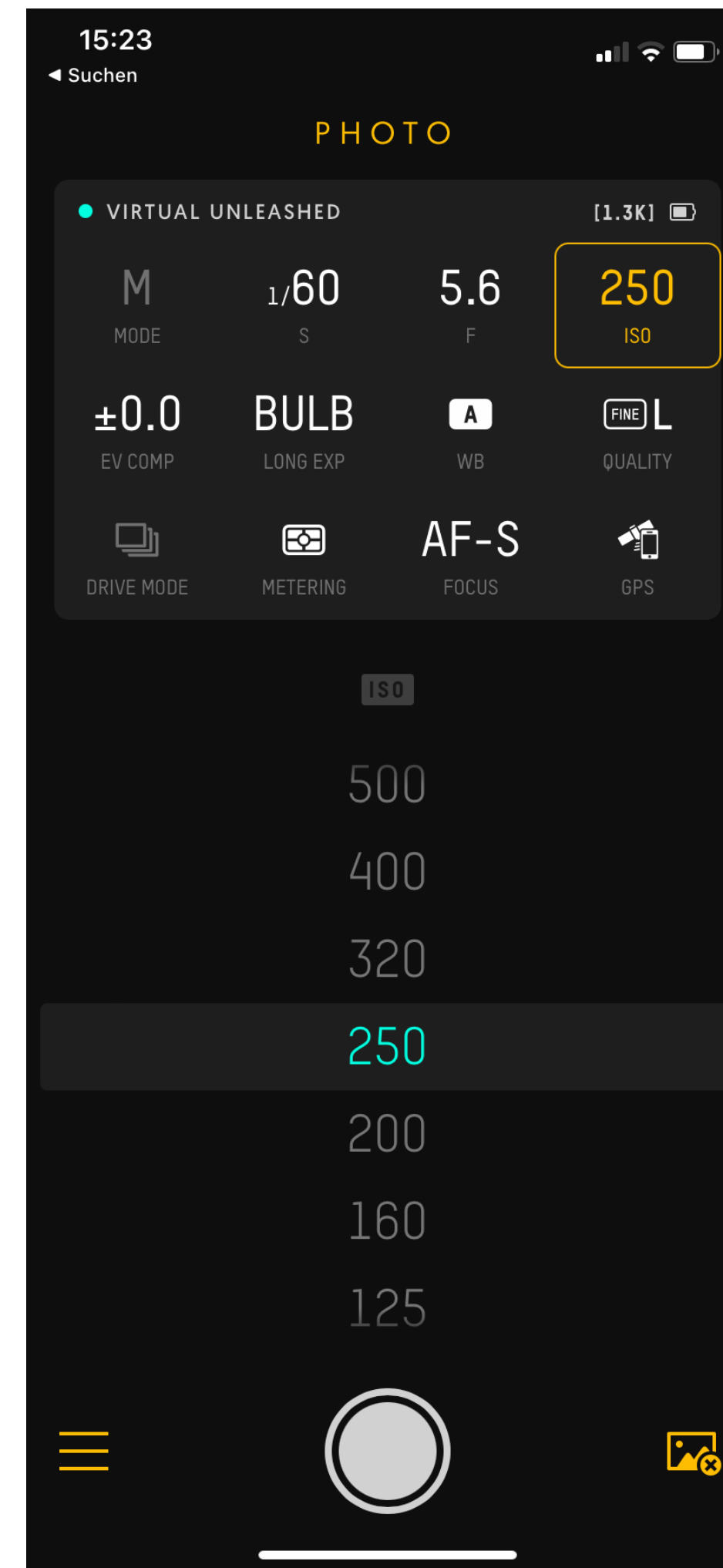
- **Feedback sammeln**
- **Fehler beim Abschlussvortrag vermeiden**

Überblick

- **Applikation und Domäne**
- **Probleme**
- **Motivationen**
- **Implementierung und Volumenarbeit**
- **Ziele**
- **Ausblick**
- **Zusammenfassung**

Applikation und Domäne

Unleashed App (Foolography GmbH)



Muster im User Interface Reengineering in ein deklaratives User Interface Framework.

07.07.2022

Applikation und Domäne

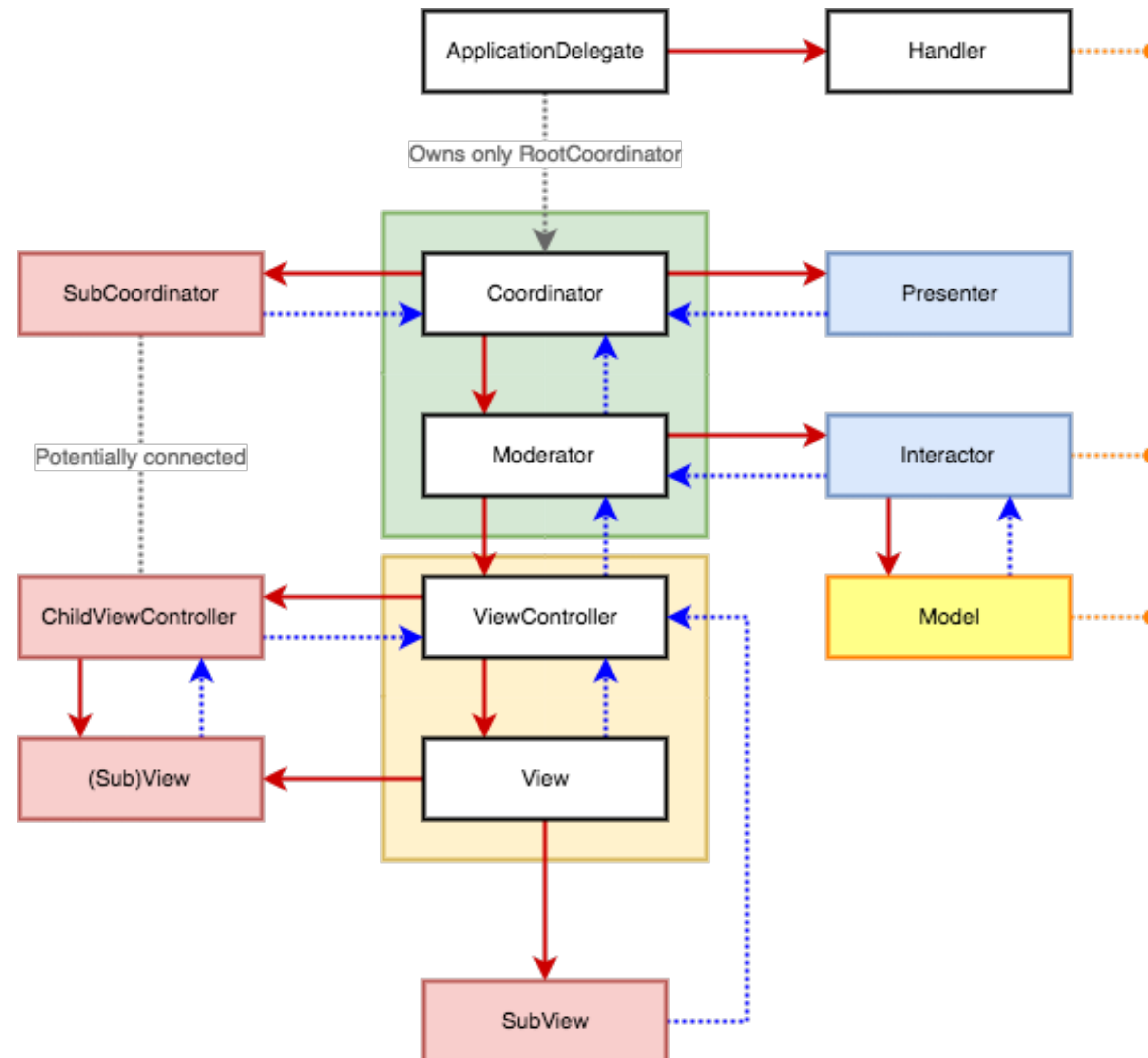
DEMO

Applikation und Domäne

- App Betriebssystem: iOS
- Programmiersprache: Swift
- Entwicklung mit Xcode (IDE) auf macOS

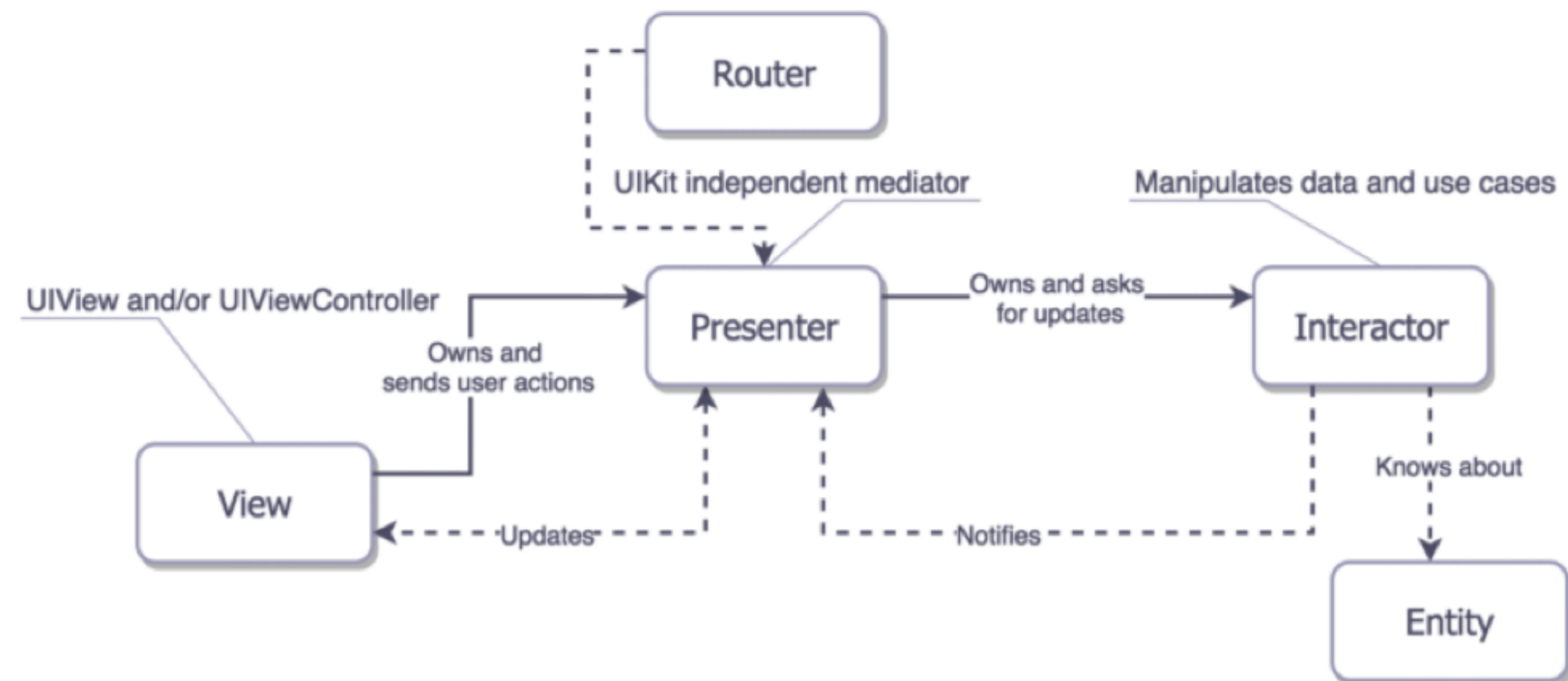
Probleme

Aktuelle Architektur:

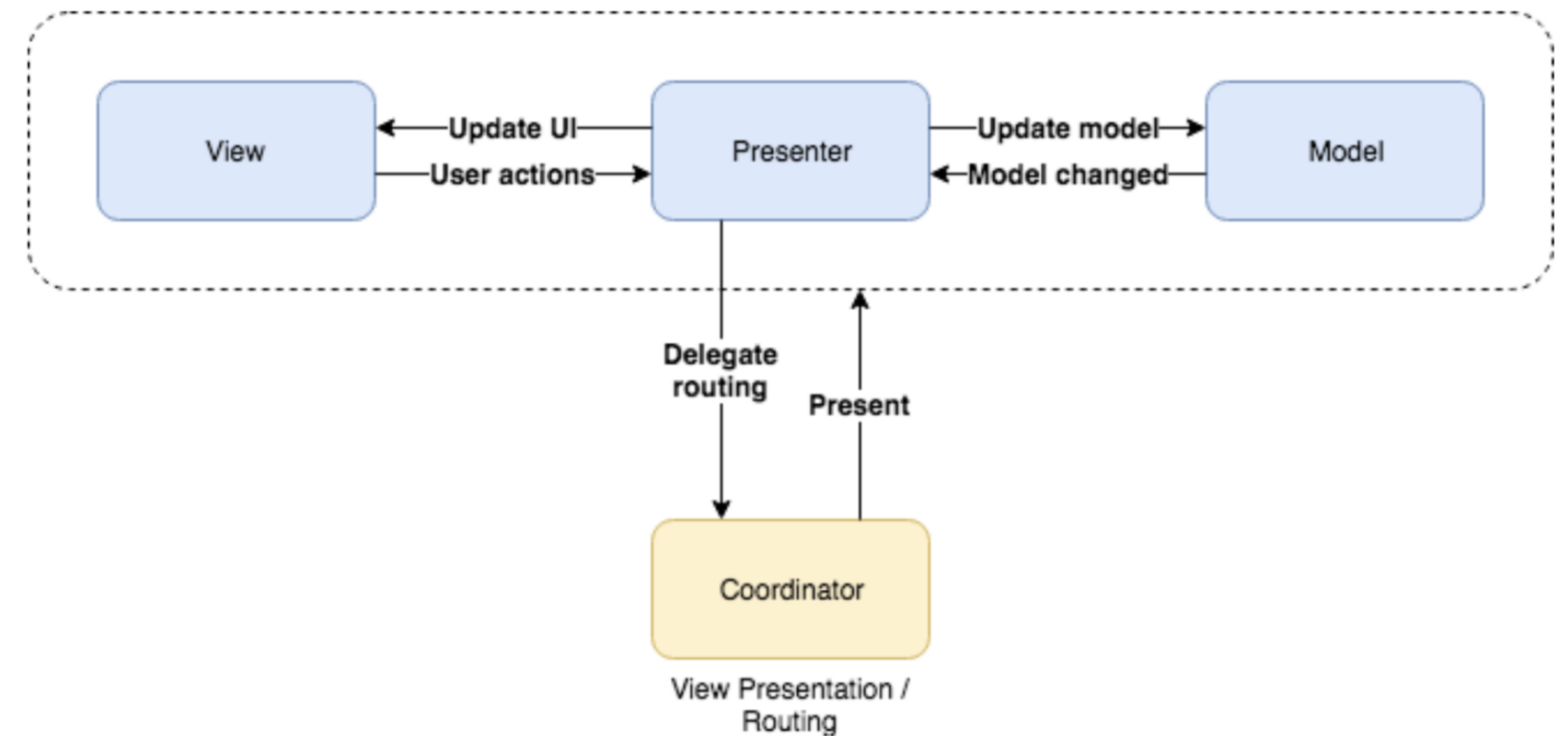


Probleme

VIPER



Coordinator- & Presenter-Pattern



Probleme

- **weitere Symptome durch technische Schulden**
- **viel Copy & Paste statt Komposition**
- **Caching Probleme**
- **zu viele verschiedene UI-Lösungsansätze**
 - **UI in Code**
 - **Storyboards und XIBs**
 - **UIKit Wrapper Frameworks**
 - **eigene SwiftUI Portierung in UIKit**

Motivation

- **Wiederherstellung der Wartbarkeit**
- **freie Boni des deklarativen Frameworks**
- **komprimierte Umgebung -> weniger Komplexität**
- **“Single source of truth”**

Implementierung und Volumenarbeit

- **User Interface aus zwei Gruppen:**
 - **Navigation**
 - **UI-Elemente durch Komposition**

Implementierung und Volumenarbeit

UI-Elemente durch Komposition

- **drei Schwierigkeitsgrade:**
 - **einfach (viele) - native SwiftUI Lösung vorhanden**
 - **mittel schwer (nicht viele) - erfordert etwas Kreativität**
 - **schwer (wenige) - benötigt wahrscheinlich UIKit**
- **MVVM Pattern wo es Sinnvoll ist**

Implementierung und Volumenarbeit

Chaos Basis



MVVM UIKit Pseudocode



Deklarative SwiftUI Repräsentation

Implementierung und Volumenarbeit

Nicht Teil der Aufgabe:

- App Einbindung
- Brücken für den Datenfluss

Alternative:

- Dummy Daten
- Dummy Logik

Implementierung und Volumenarbeit

Zusätzliche Datensätze:

- Vorgehensweise
- Gedankengänge
- Fragestellungen
- Einschätzung der Zeit
- Entscheidungen

Ziele

Es gibt zwei Hauptziele:

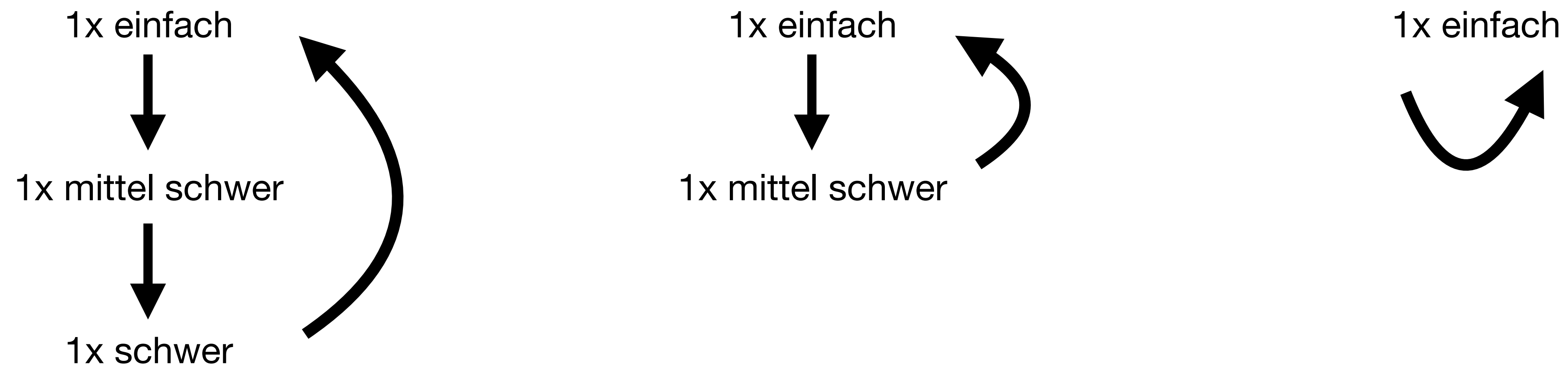
- die Muster welche im Prozess sich zu erkennen geben
- UI-Reengineering der App

Wöchentliche Zyklen:

- Implementierung
- Sammlung von Datensätzen
- Auswertung / Analyse

Ziele

Geplante Bearbeitung der UI-Elementen nach deren Grad:



Ziele

Geplante Schreibphase:

- kann man noch schlecht einschätzen**
- wöchentliche Zyklen + schriftliche Auswertung der Datensätze**

Ausblick

**Welche Entwurfsmuster wurden verwendet?
Welche Prozessmuster sind erkennbar?
Gibt es neue Muster, oder nur Abwandlungen?**

Finales Resultat noch unklar.

Zusammenfassung

- (1) Reengineering eines UI-Teils der Unleashed App mit einem deklarativen Framework.**
- (2) Untersuchung auf verwendete Entwurfs- und vorkommende Prozessmuster.**

Quellen

- <https://gcherubini.medium.com/mvp-with-coordinators-for-ios-applications-7d668b8a0b0e>
- <https://medium.com/yay-its-erica/intro-to-the-viper-design-pattern-swift-3-32e3574dee02>
- <https://developer.apple.com/documentation/swiftui/state-and-data-flow>
- <https://www.foolography.com/products/unleashed/>

DANKE