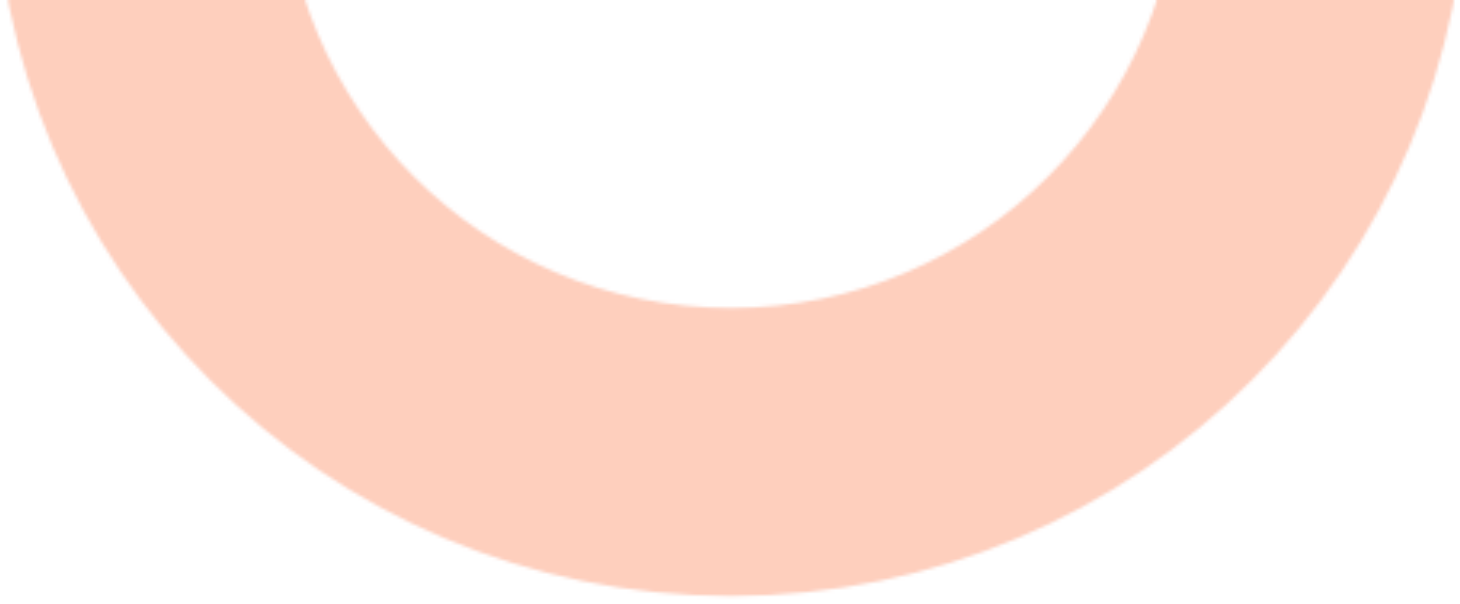
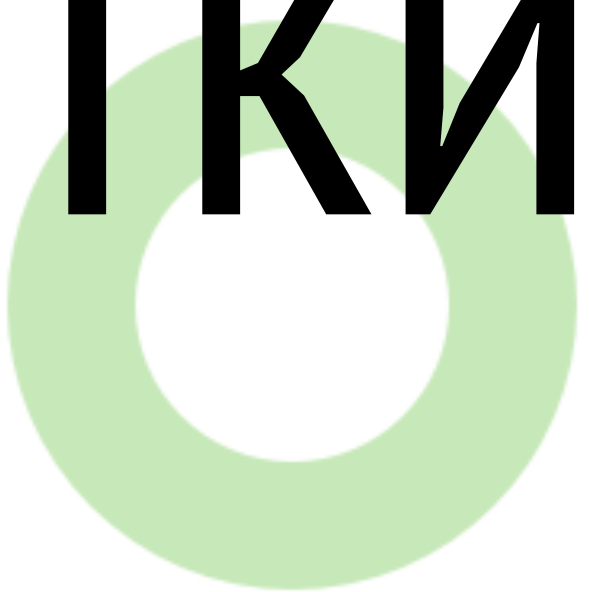
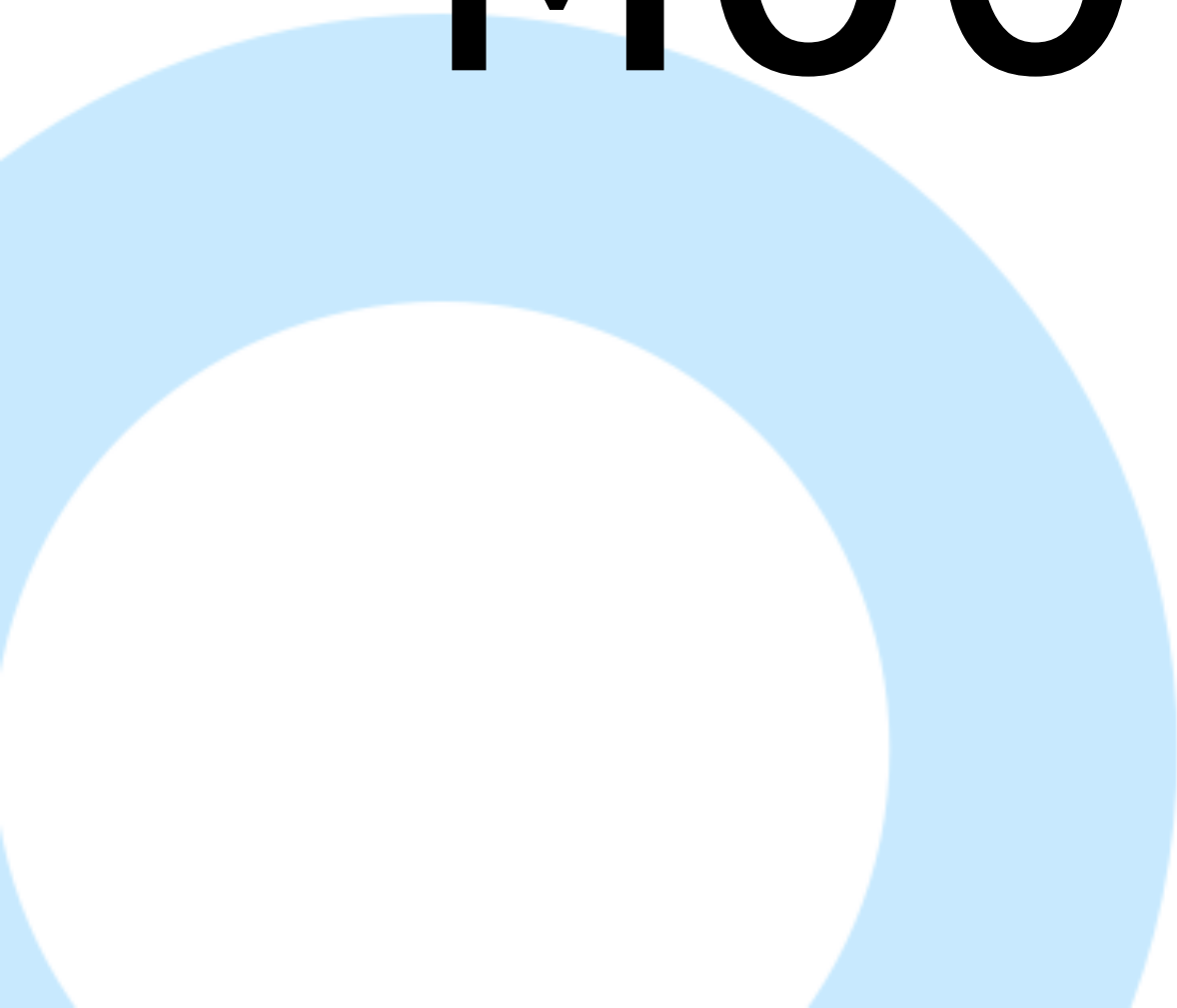




Игорь Кашкута

Пизанская башня

мобильной разработки



Кратко о себе

2



Геймдев



Reactive Objective-C++11
150+ проектов в Xcode
Offline



Swift + Objective-C
Well-Defined Online Platform

2010



2011



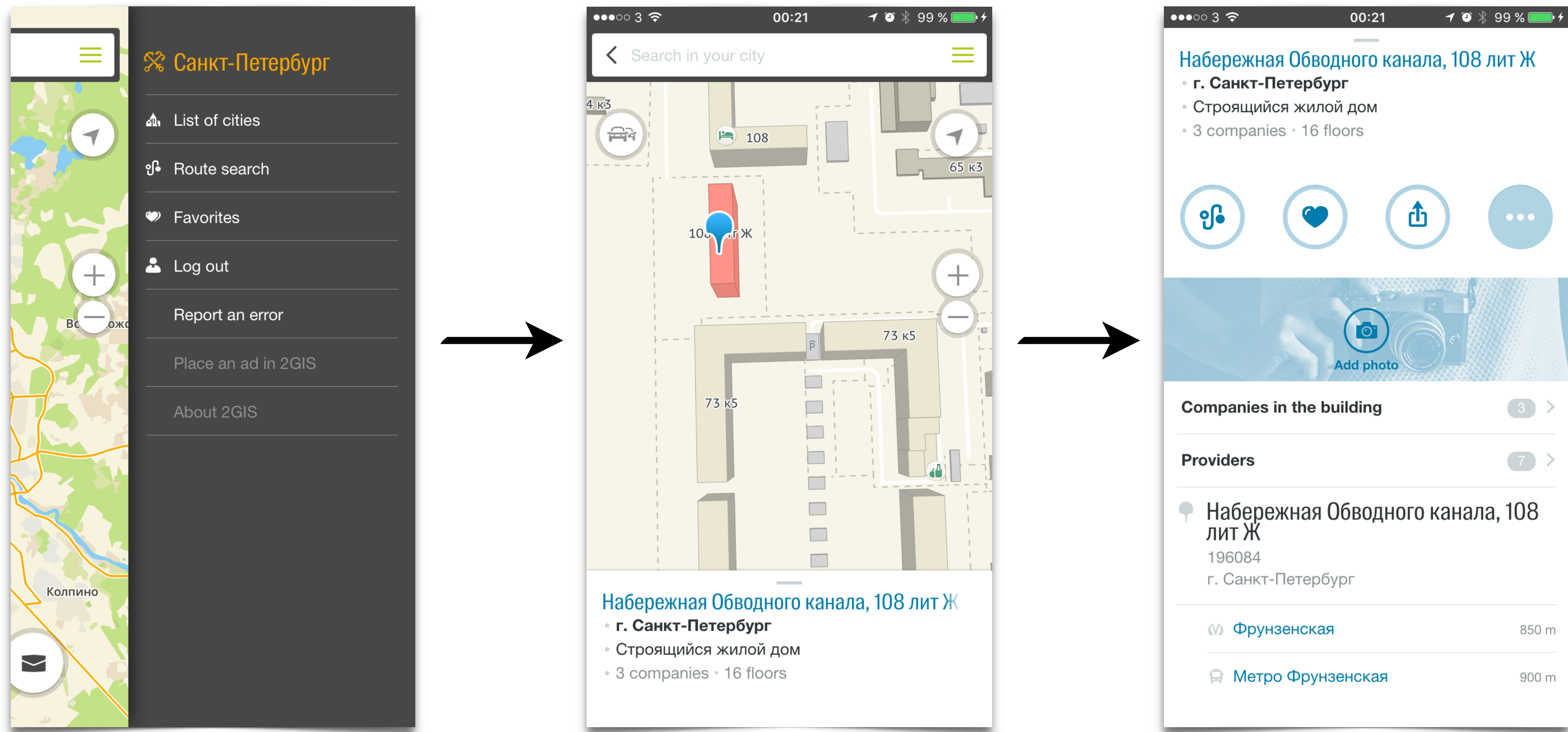
2015



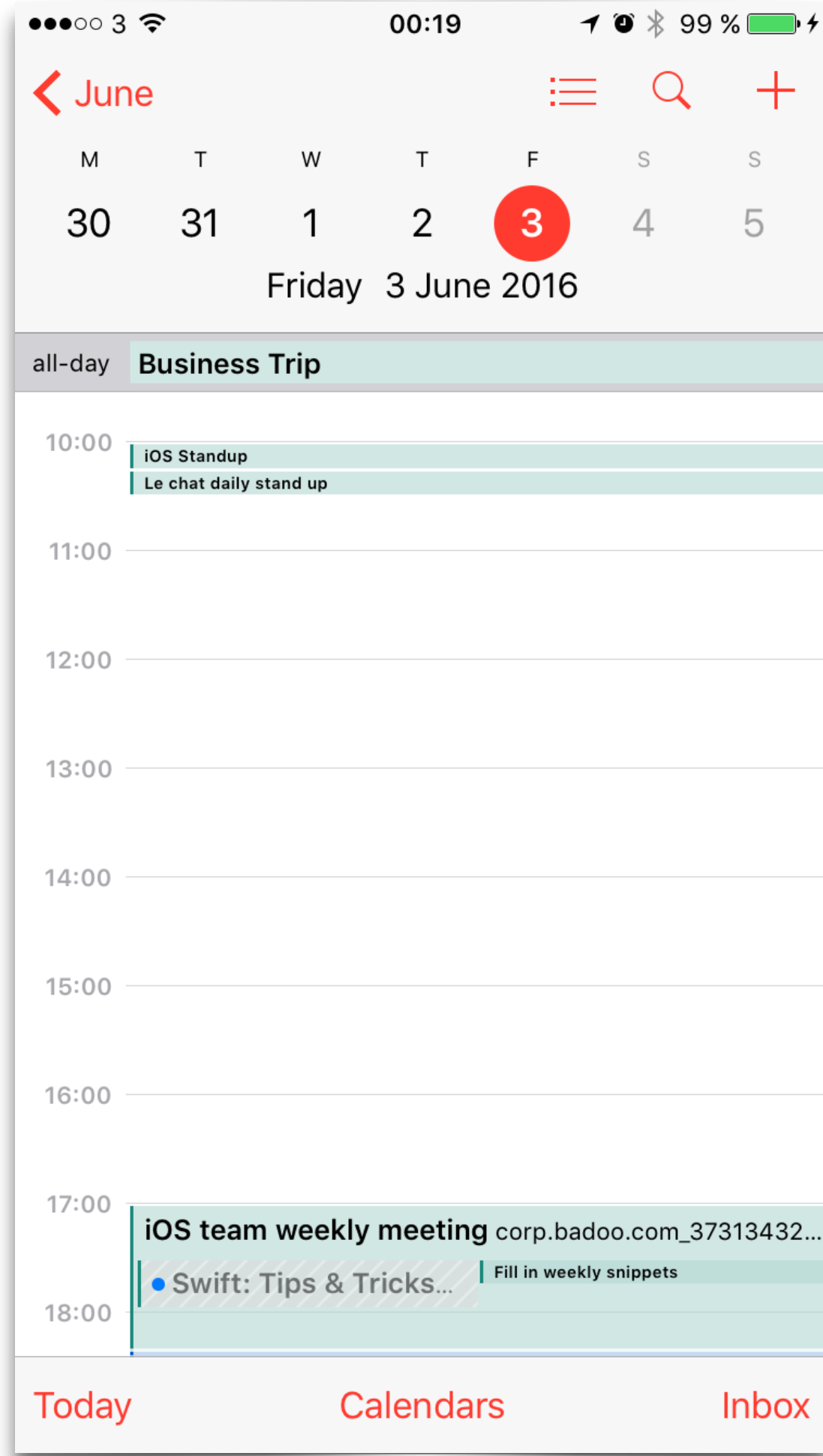
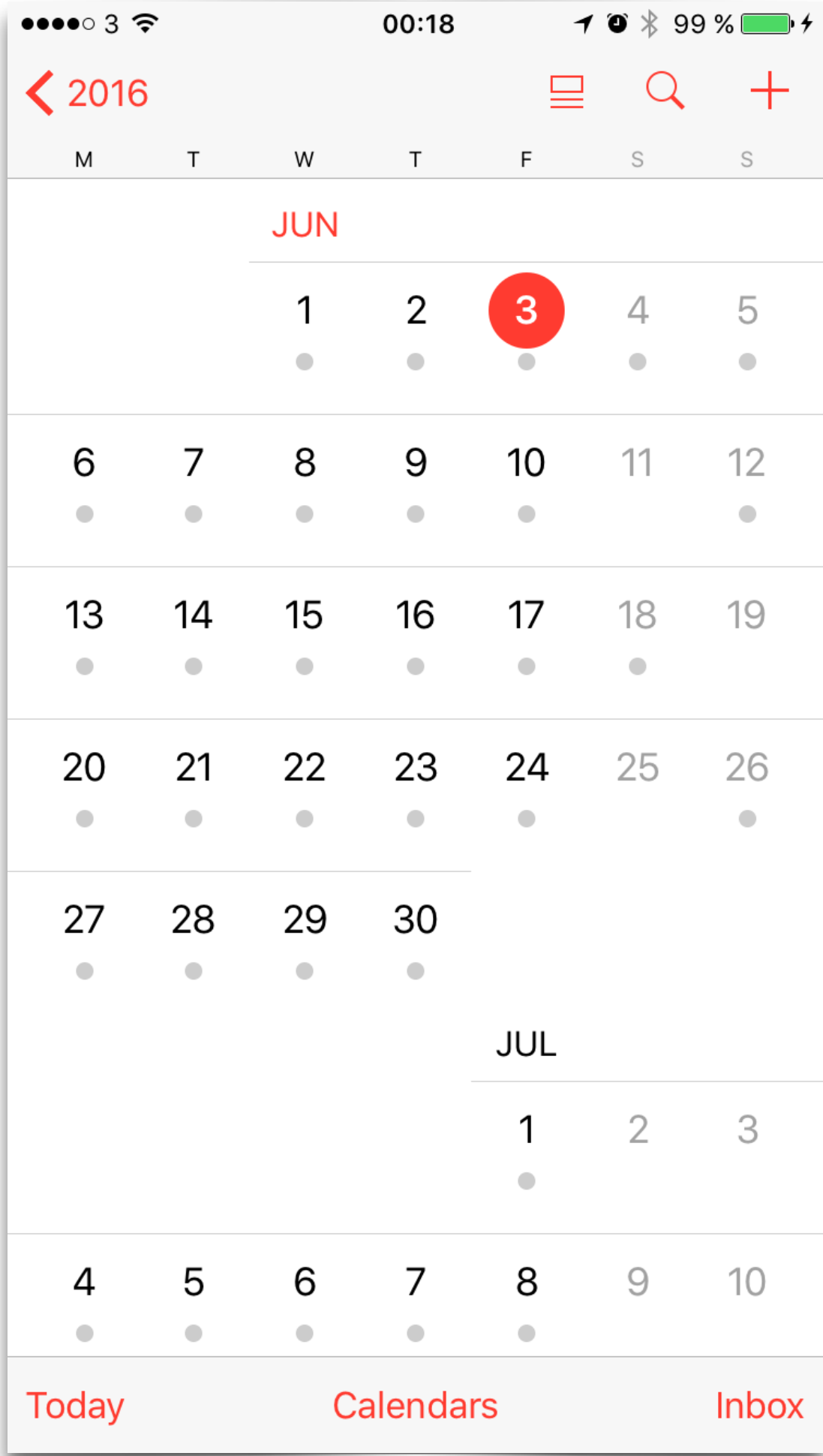
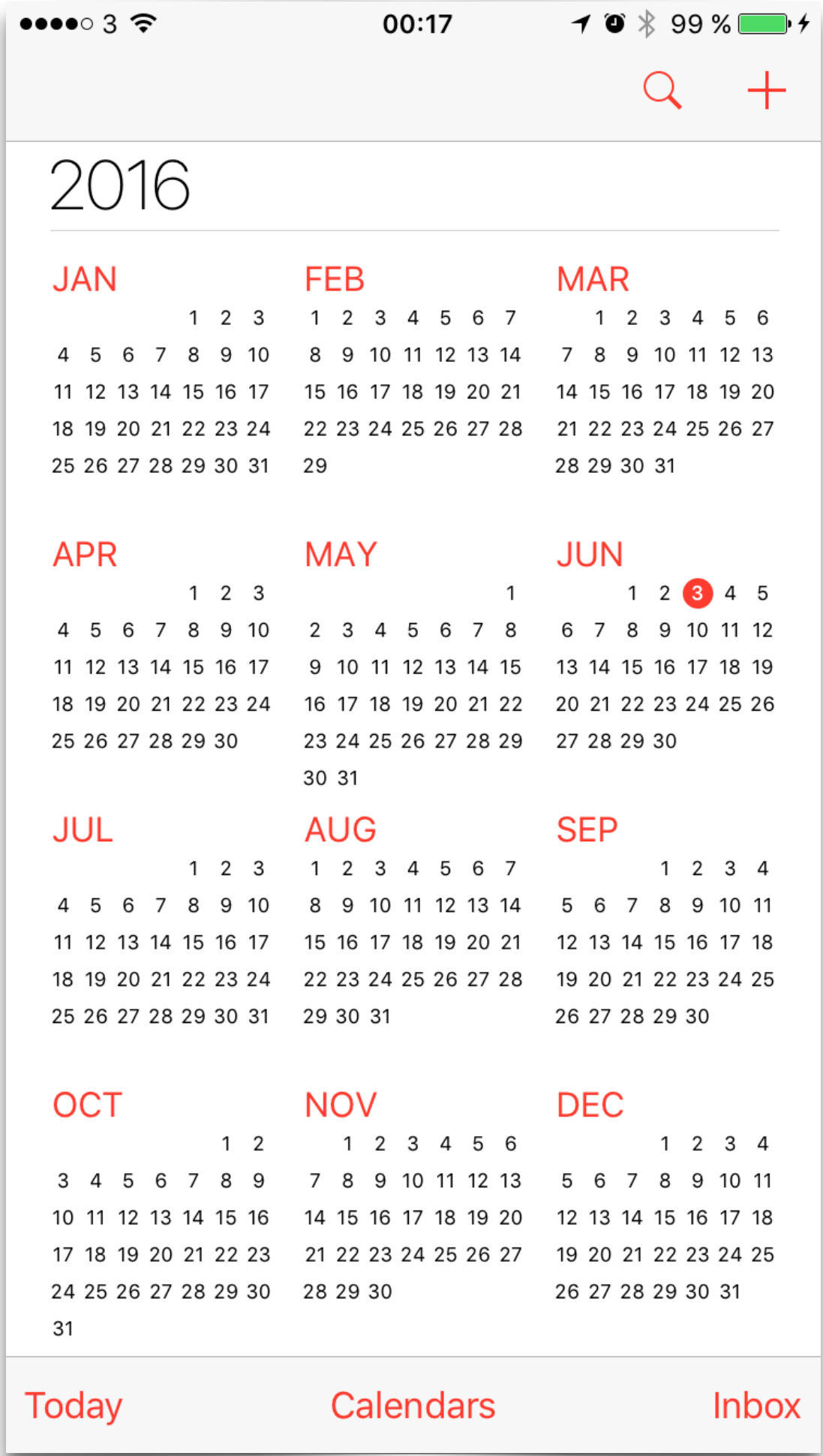
Семь бед

1. Custom Navigation

Custom Navigation



Custom Navigation



2. Routing

Deep Linking & Universal Links

Cloud и In-App пуш-нотификации

Нет рекомендаций по реализации

3. Asynchronous Layout

Asynchronous Layout

10

Что: лейаут UI элементов вне главного потока

Где: любые сложные списки

Зачем: отзывчивость, кэширование

Фреймворк для разработки чатов с асинхронным лейаутом текста

<https://github.com/badoo/chatto>

4. Interactive Animations

Interactive Animations

12

Что: интерактивные анимации как в UINavigationController
при возвращении на предыдущий экран

Зачем: помогают сделать работу с приложением
более “живым” и удобным

Как: с болью :—(

5. Hot Reloading

Hot Reloading

14

Что: изменение UI на лету в работающем приложении

Зачем: скорость разработки, продуктивность

Как: ?

Hot Reloading

15

“

– На какие вопросы тебе трудно ответить,
глядя в storyboard?

– Что к чему привязано,
что как называется и как что переместить,
чтобы всё в клочья не порвало.

6. Styles

Что: система стилей

Зачем: ночная тема навигатора или читалки, скорость

Как: сложно

Подробнее про Ångström Style System:

<https://habrahabr.ru/post/278787/>

7. State Control

State Control

19

Что: сохранение/загрузка состояния приложения

Зачем: восстановление контекста,
синхронизация между устройствами,
интроспекция

Как: дисциплинированно

Хочу системный подход

Системный подход

21

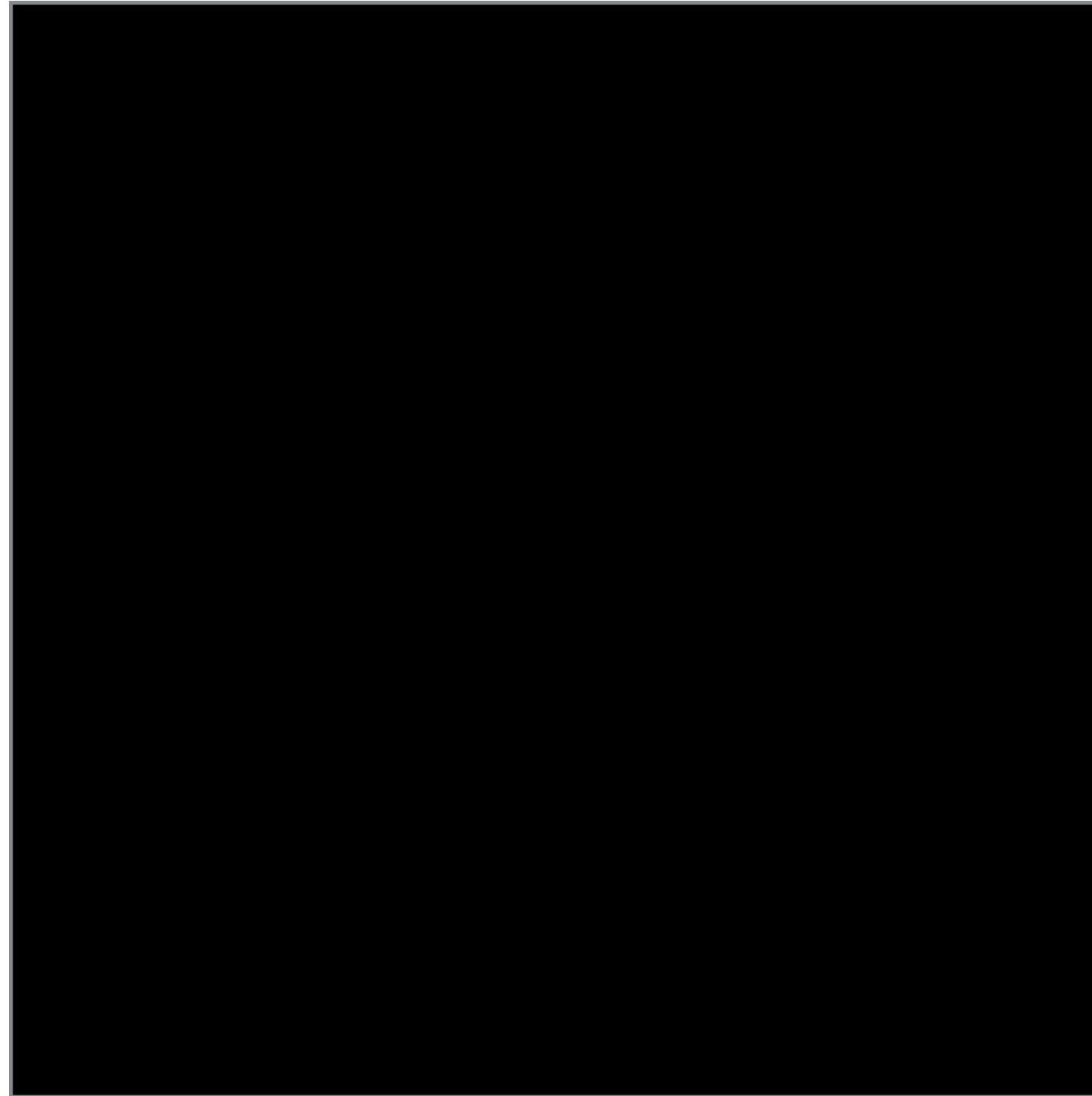
Сейчас — пластилиновая фигурка из нескольких слоёв

А я хочу библиотеку, где всему есть своё место
и понятно, куда класть новое

Наблюдения

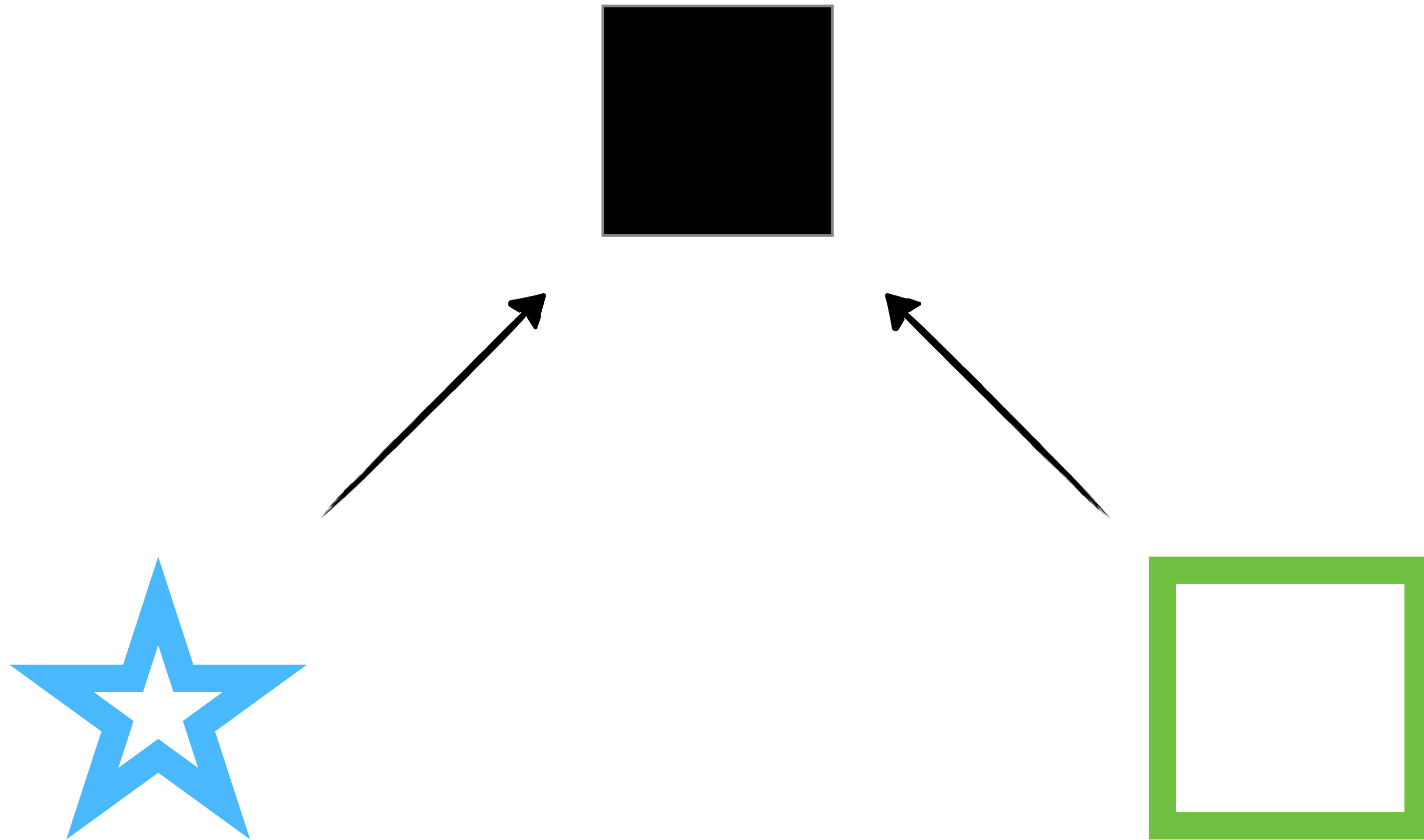
UIView & UIViewController

23



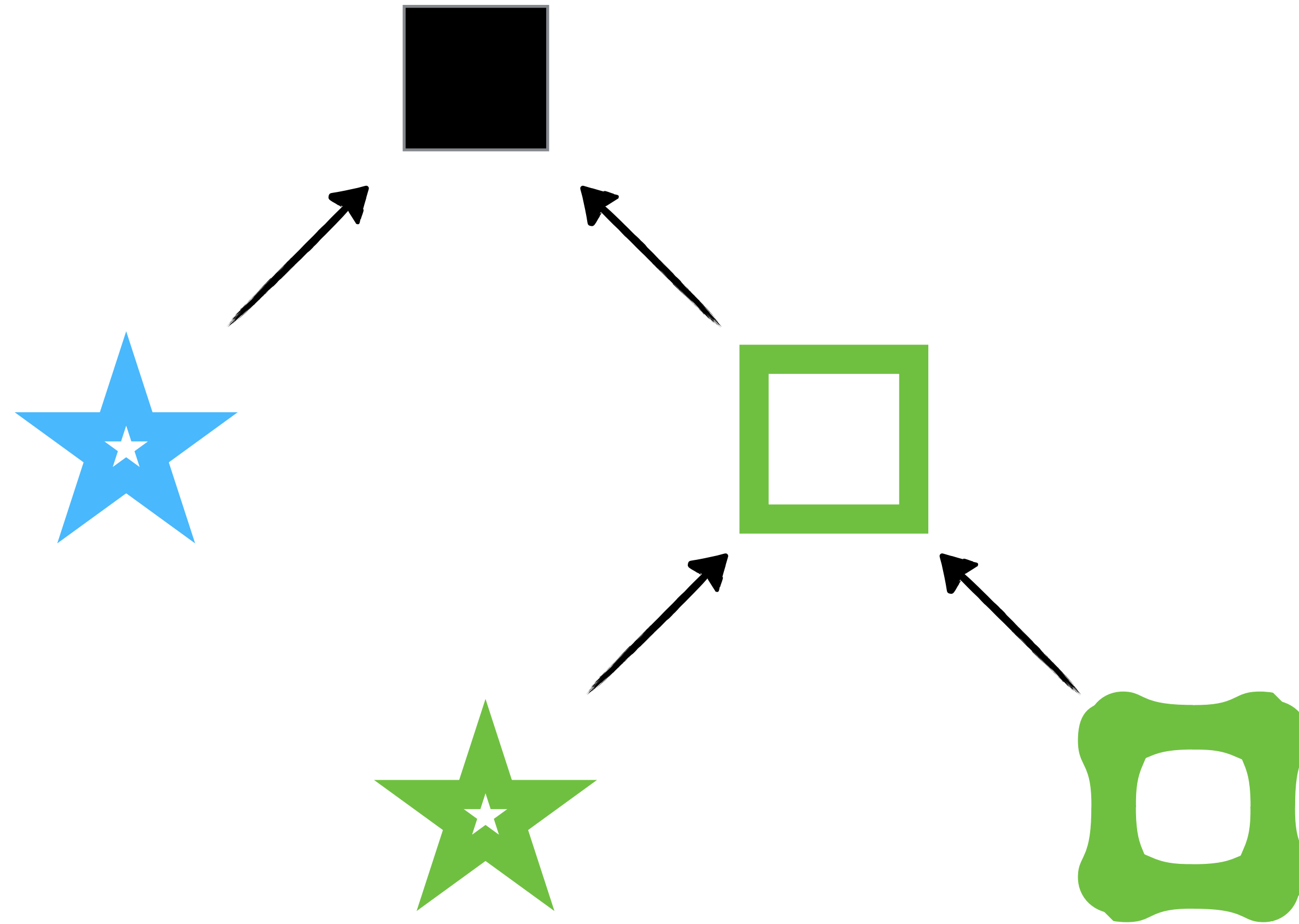
UIView & UIViewController

24



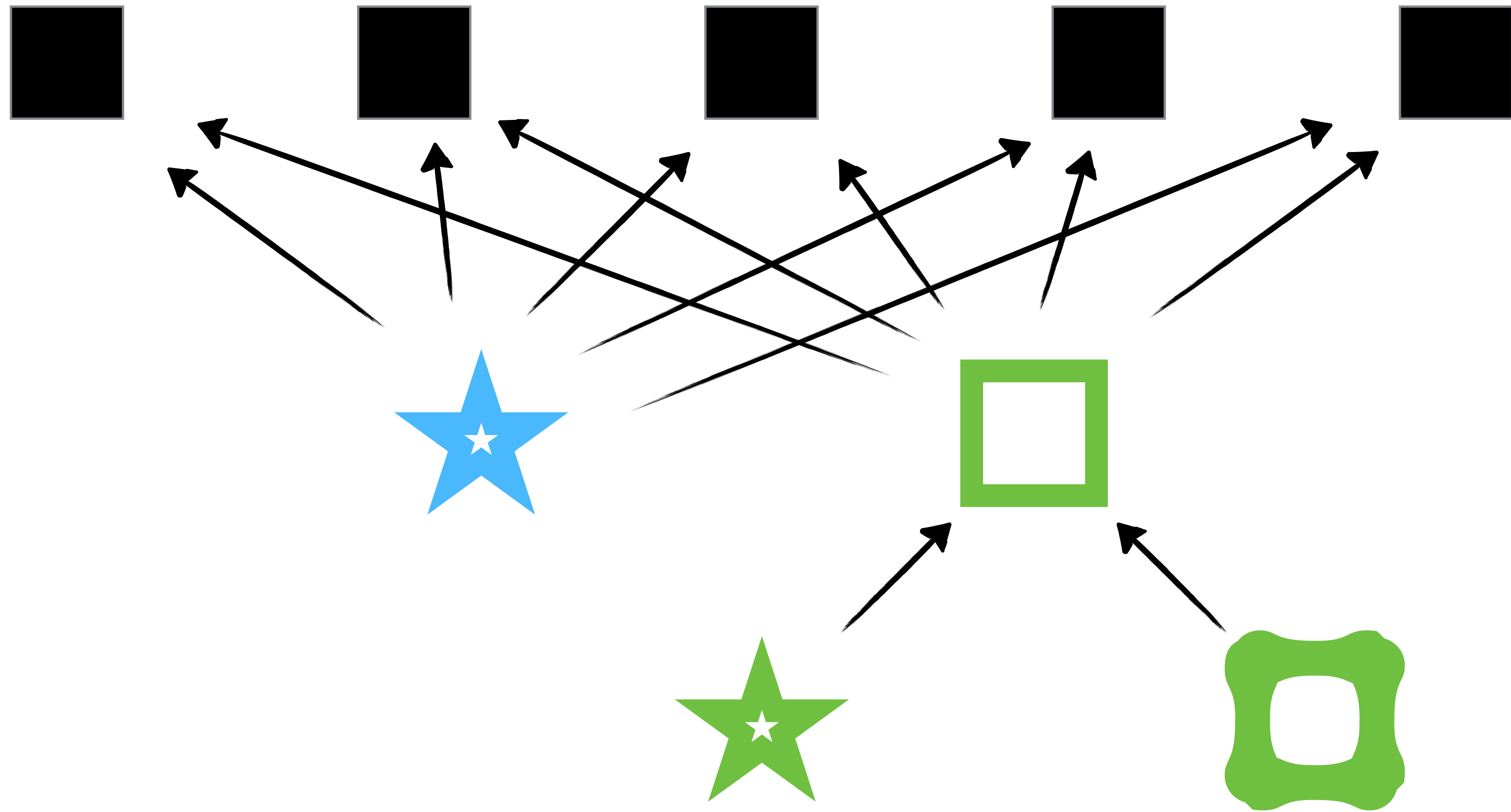
UIView & UIViewController

25



UIView & UIViewController

26



Паттерны

27

MVC

MVP

MVVM + FRP

VIPER

Подробнее про всех:

<https://habrahabr.ru/company/badoo/blog/281162/>

Чего не хватает

28

Полного контроля за происходящим

Прозрачных взаимосвязей

React Native

29

- ✓ Asynchronous Layout
- ✓ Hot Reloading
- ✓ State Control
- Interactive Animations
- JavaScript
- Layout Customisation

Ещё один ответ

Геймдев — коллеги по цеху

31

Полностью кастомные, как геймдизайнеру угодно

Ресурсов гораздо больше, чем в приложениях

CES — базовый паттерн движка Unity

System

32

Render

Physics

Audio

Scripts

Component

33

Render

1

3D-модель
человека

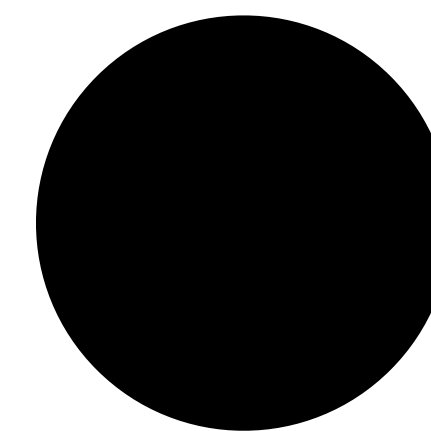
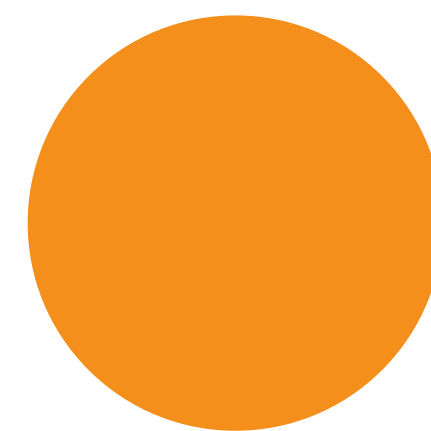
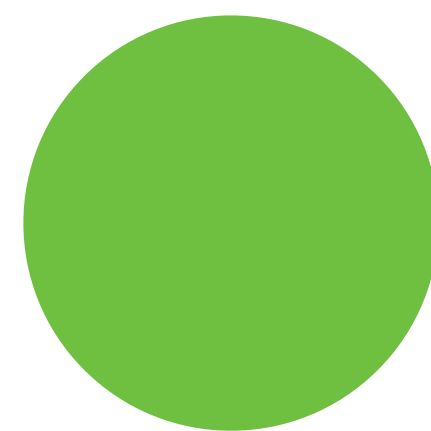
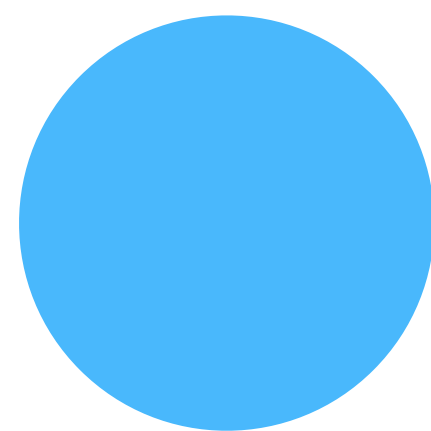
2

3D-модель
пришельца

3

3D-модель
дерева

Солдат



Component-Entity-System

35

	Render	Physics	Audio	Scripts
Солдат	1	1	1	1
Дом	2	2		
Мина		3	2	2

Component-Entity-System

36

	Render	Physics	Audio	Scripts
Солдат	1	1	1	1
Инопланетный солдат	2	1	2	1
Мёртвый солдат	1	1		

CES in Apps

A blue rounded square box containing the text 'UIKit' in white.

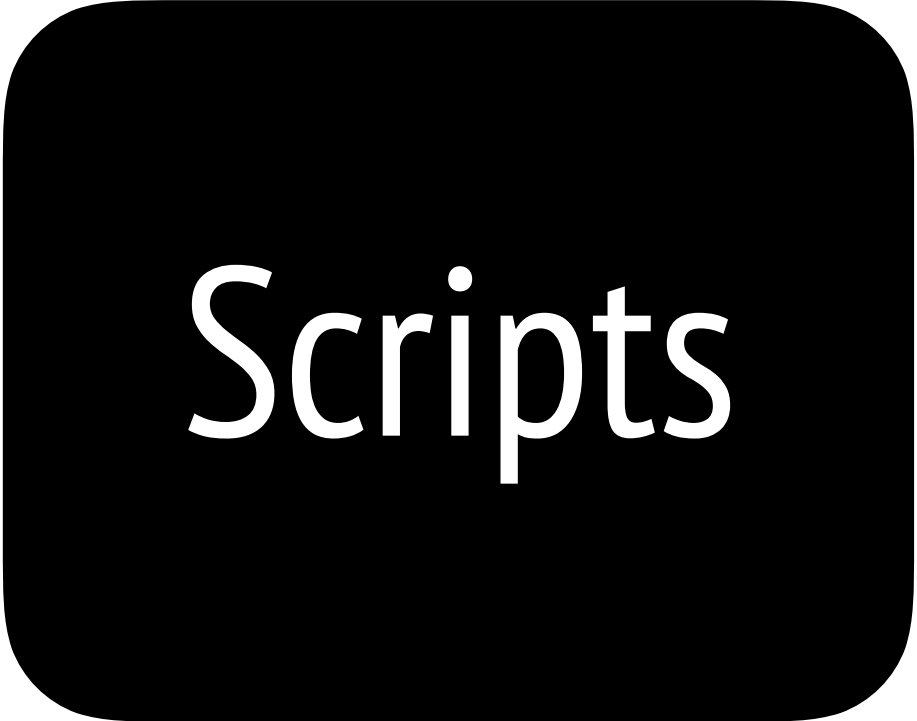
UIKit

A green rounded square box containing the text 'Layout' in white.

Layout

An orange rounded square box containing the text 'Style' in white.

Style

A black rounded square box containing the text 'Scripts' in white.

Scripts

Component

39

UIKit

1

UILabel

2

UIImageView

3

MKMapView

Component

40

Layout

1

Fixed

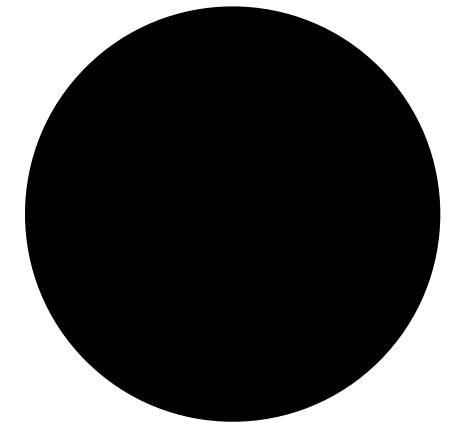
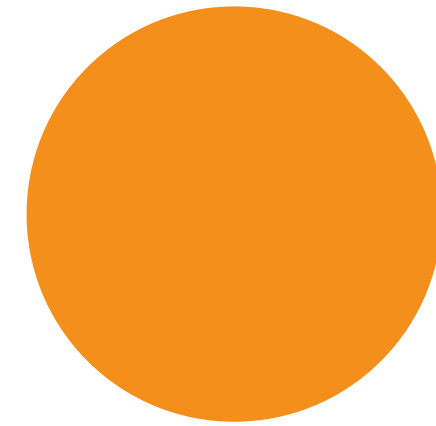
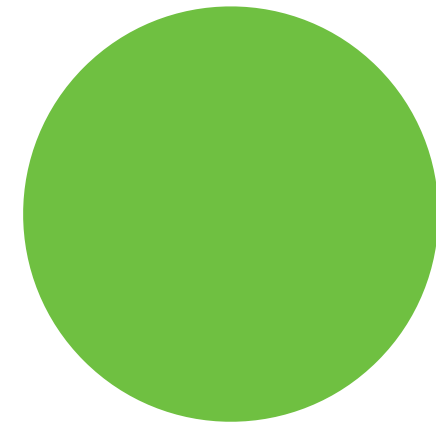
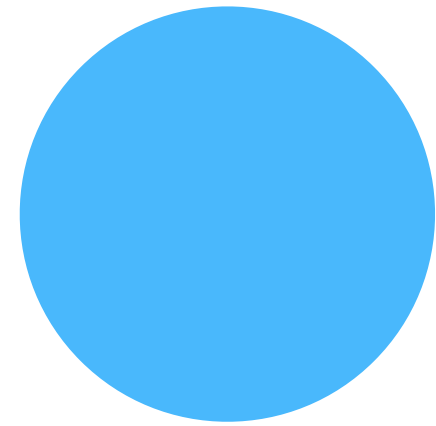
2

Stack

3

Flexbox

Chat Bubble



Component-Entity-System

	UIKit	Layout	Style	Scripts
Chat Text Bubble	1	1	1	1
Chat Image Bubble	2	2	2	
Chat Table		3		2

Следствия

Следствие первое

44

Компоненты можно переиспользовать,
собирать из них другие entity

Следствие второе

45

Состояние можно вынести
в “компоненты состояния”, что позволит
контролировать всё приложение системно

Следствие третье

46

Entity могут образовывать иерархию,
не привязанную к UIKit – проще тестировать.
Можно делать интеграционные тесты на лейаут!

Следствие четвертое

47

Системы консолидируют общую логику,
которую часто непонятно куда класть

Следствие пятое

48

Проблема владения и доступа решилась:

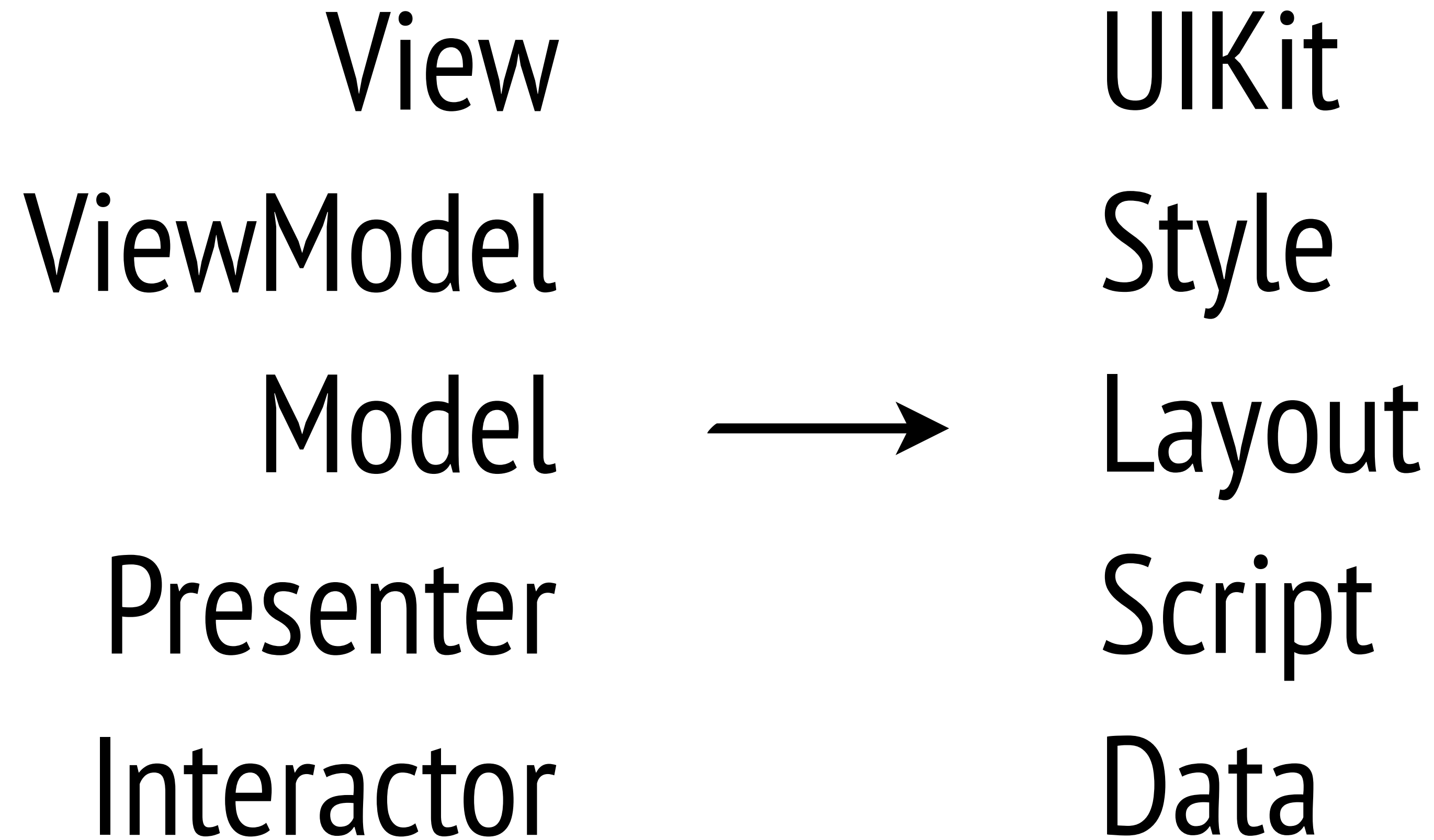
Entity – владелец компонент, поэтому они могут
ссылаться друг на друга как угодно.

Их время жизни совпадает

Замечания

Однозначные “полочки”

50



Вопросов ещё больше, чем ответов

Это лишь гипотеза, надо проверять

Я потихоньку начал, следите за новостями ;—)



Спасибо!

Игорь Кашкута

@ikashkuta

gmail, twitter, telegram, etc.



<https://twitter.com/BadooTech>

