ТИНЬКОФФ

Архитектурное собеседование
По обе стороны баррикад

IT's TINKOFF

Володин Кирилл

Руководитель отдела

volodin.kirill.a@gmail.com

Telegram: @leoniknik

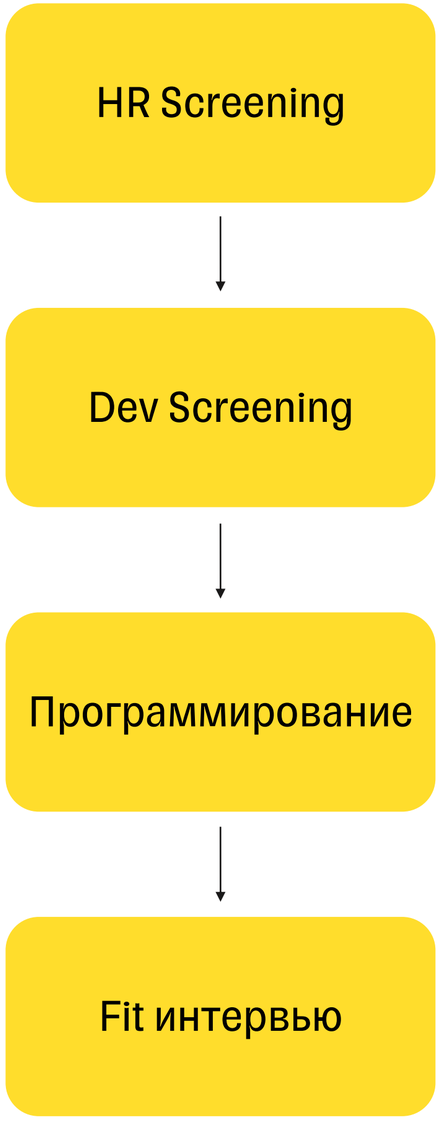
План

1. Проблемы найма
2. Классический System Design
3. Архитектурная секция
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

План

1. **Проблемы найма**
2. Классический System Design
3. Архитектурная секция
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

Старый процесс найма



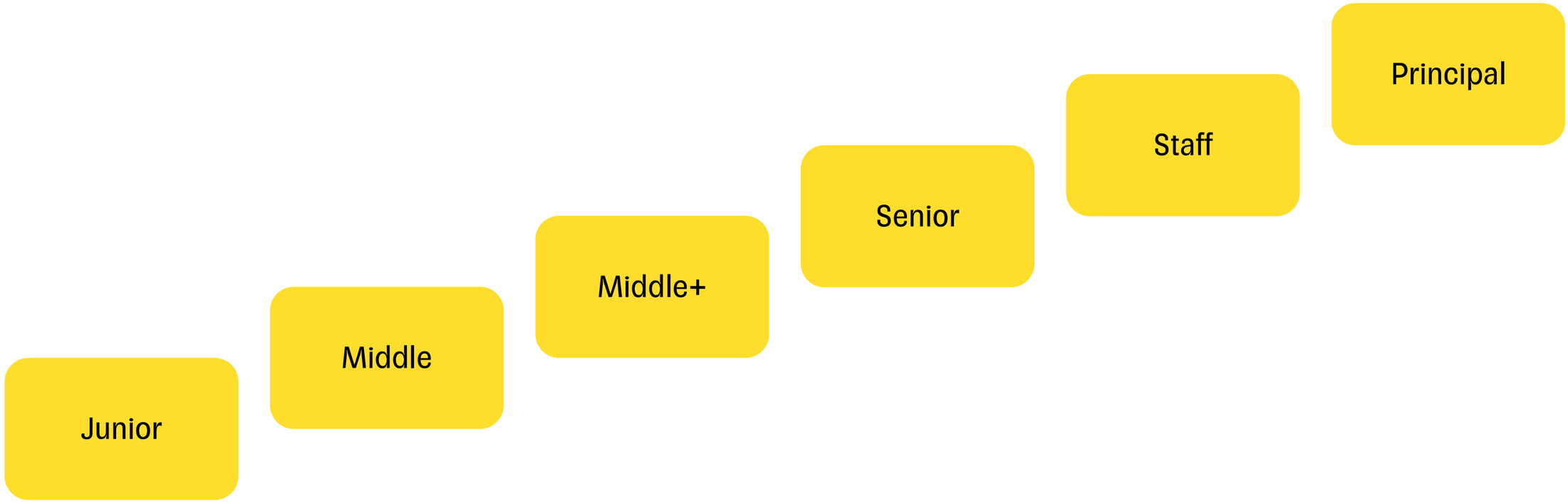
1. Адекватность кандидата
2. Подходящий стек
3. Есть позиция для кандидата

1. Теория
2. Swift
3. iOS SDK

1. Live-coding
2. Swift
3. iOS SDK

1. Насколько кандидат подходит компании (ее культуре)
2. Насколько кандидат подходит команде
3. Насколько команда подходит кандидату

Система грейдов



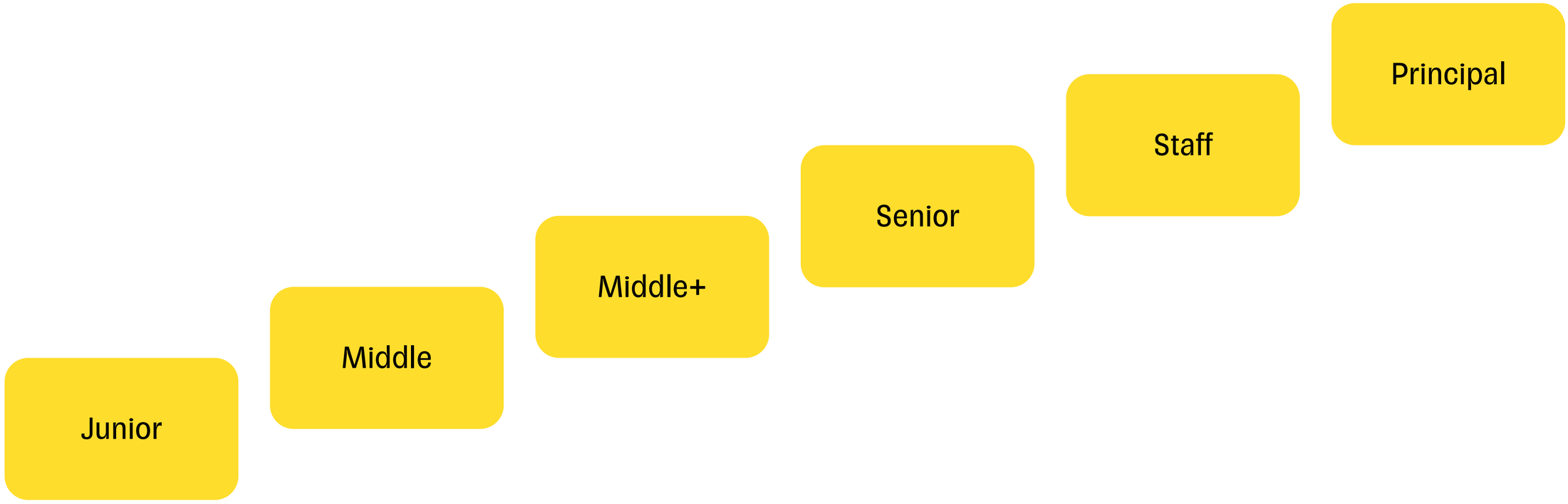
Staff инженер

Внедрить Trunk Based Development

Мигрировать проект с Cocoapods на SPM или Bazel

Развивать домен надежности, формируя практики и технические решения

Система грейдов



Все пересобрать

План

1. Проблемы найма
2. **Классический System Design**
3. Архитектурная секция
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

Классический System Design

1. Функциональные требования
2. Границы системы (что выходит за рамки задачи)
3. Нефункциональные требования

1. Концептуальная схема, модель данных
2. Потоки данных и взаимодействие систем
3. Happy path, corner cases

1. Детализация схемы
2. Конкретные технологии
3. Масштабирование

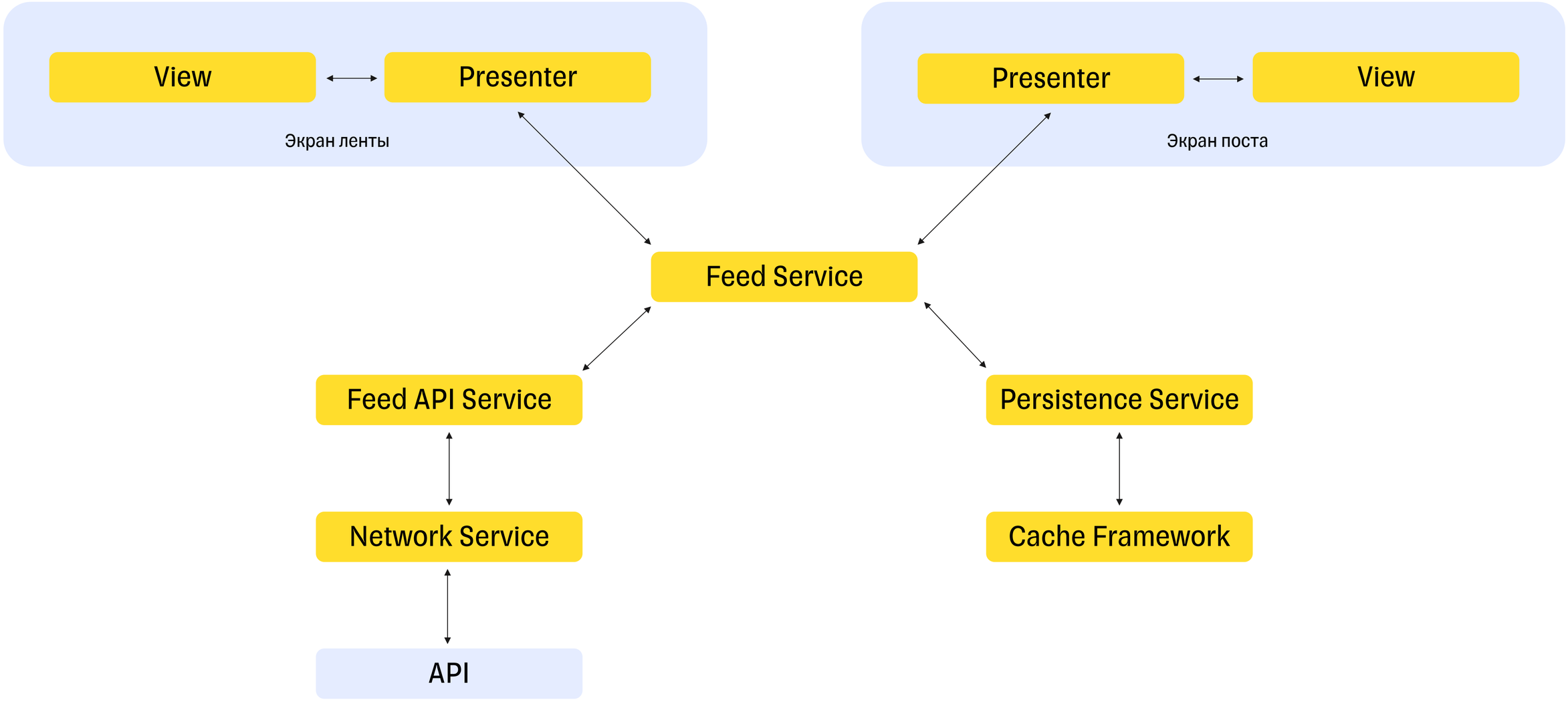
1. Расширить scope задачи
2. Усложнить нефункциональные требования
3. Обсудить альтернативные решения

План

1. Проблемы найма
2. Классический System Design
3. **Архитектурная секция**
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

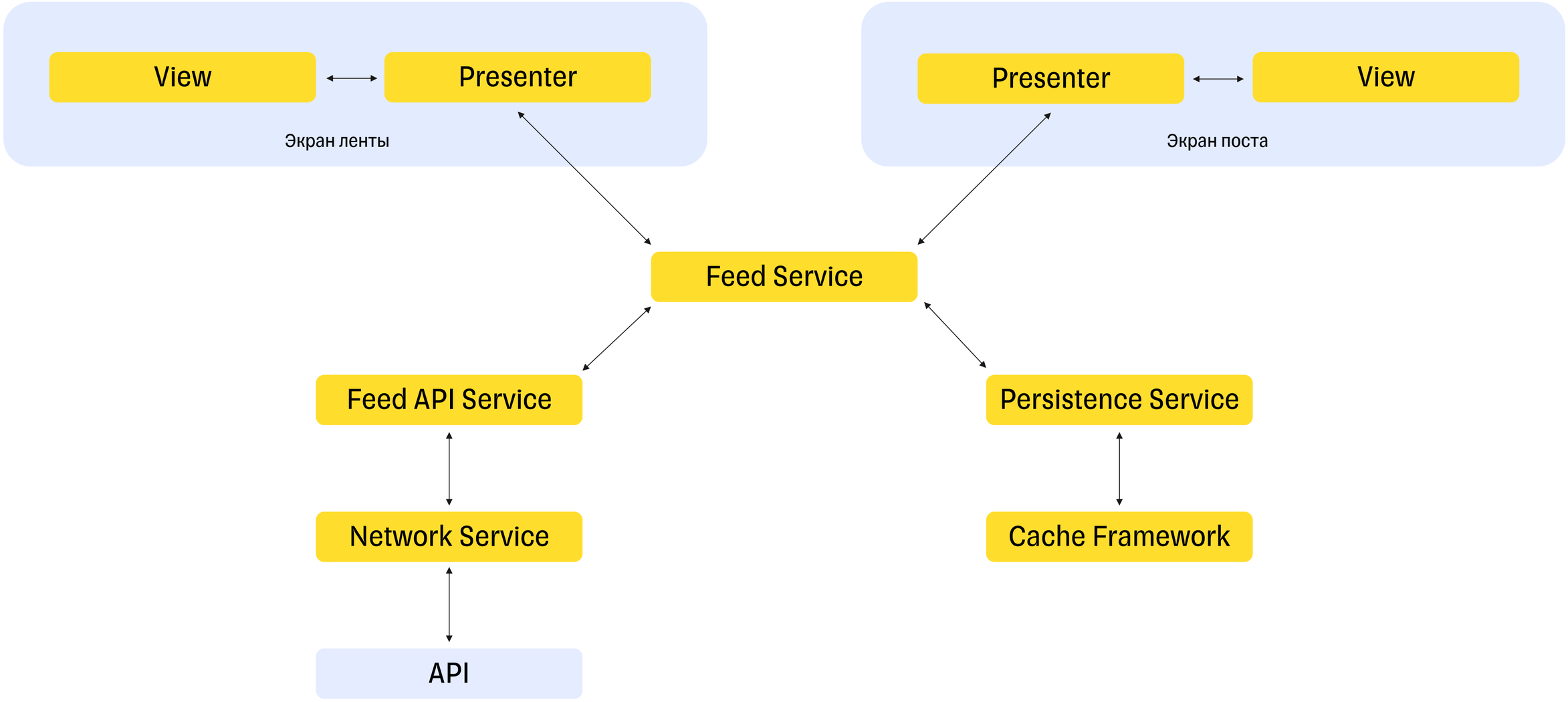
Разработать ленту новостей

Пример архитектурной секции

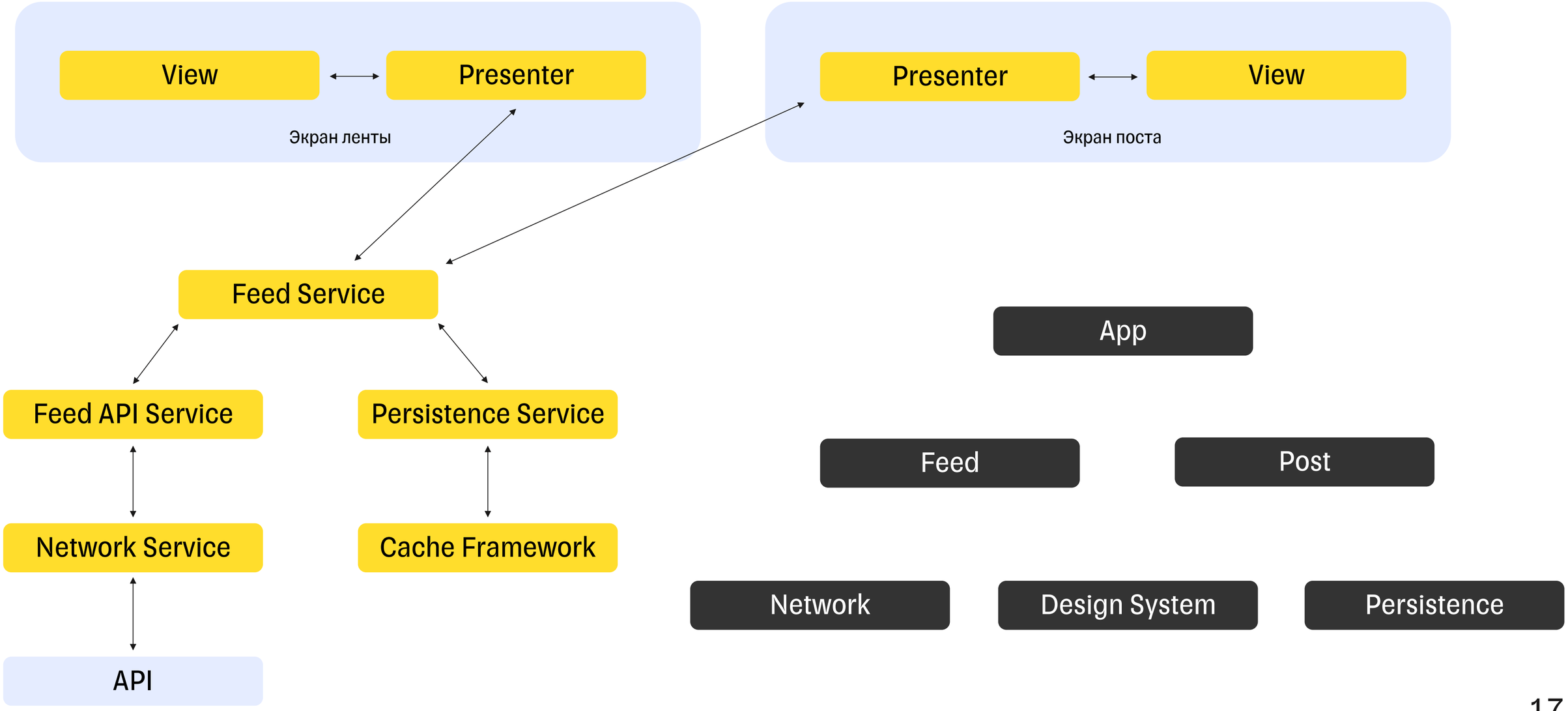


Как модуляризировать проект?

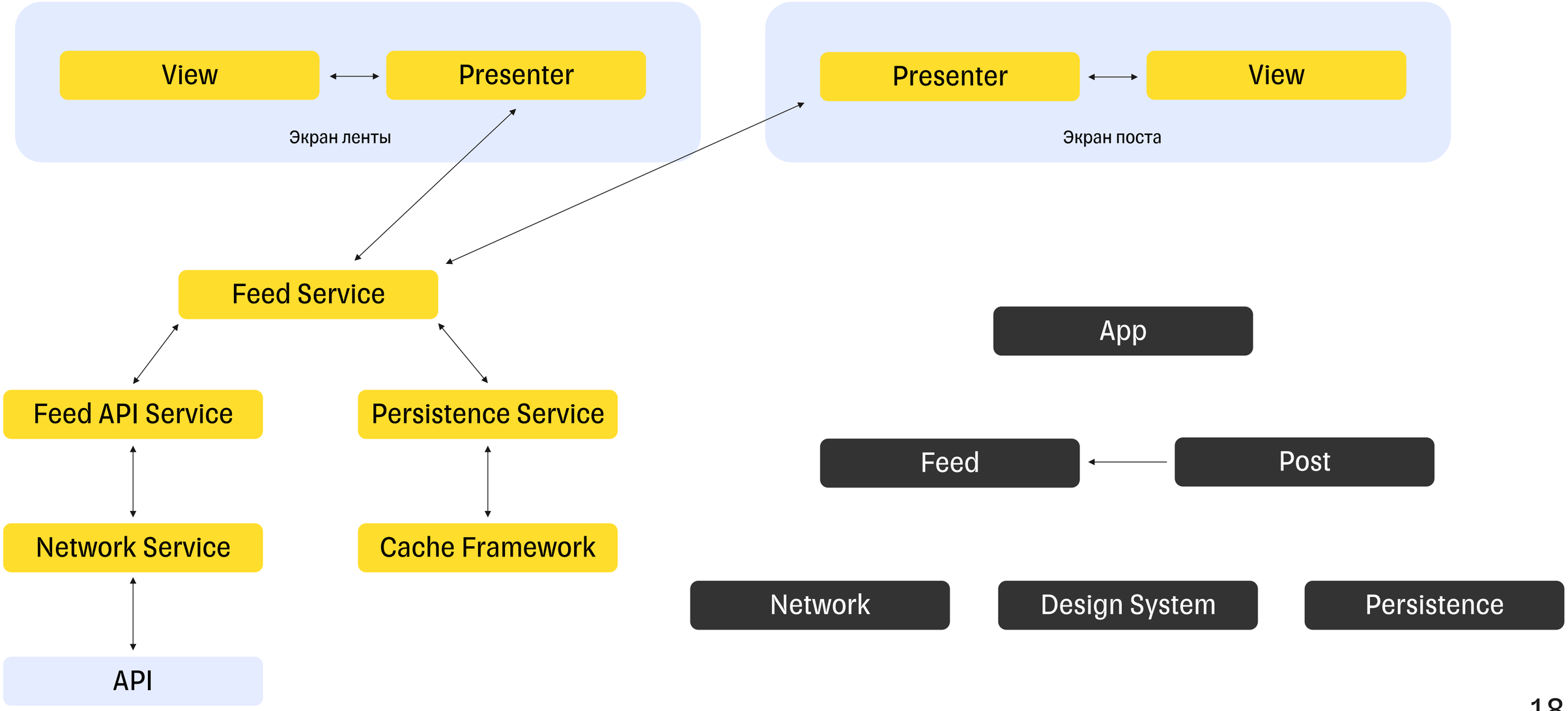
Как модуляризировать проект



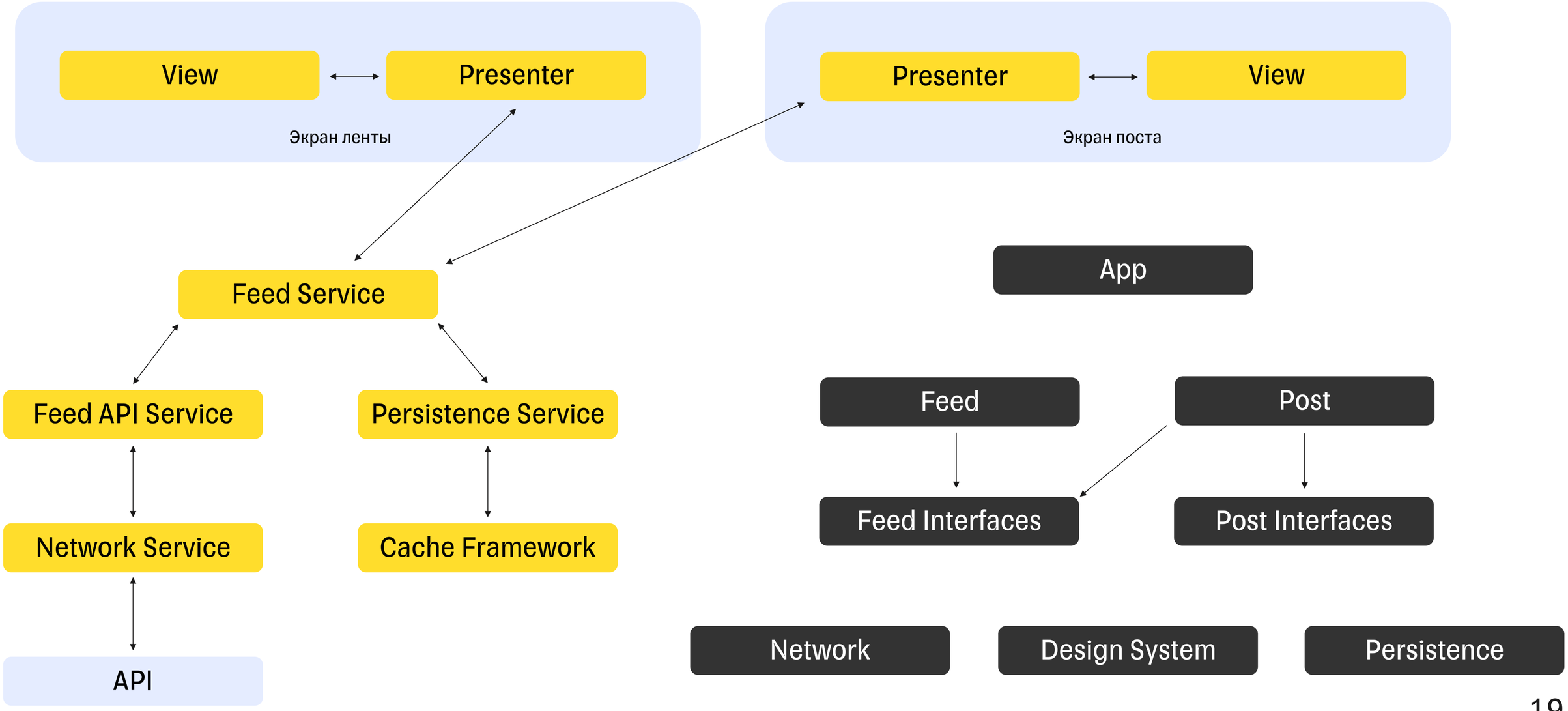
Как модуляризировать проект



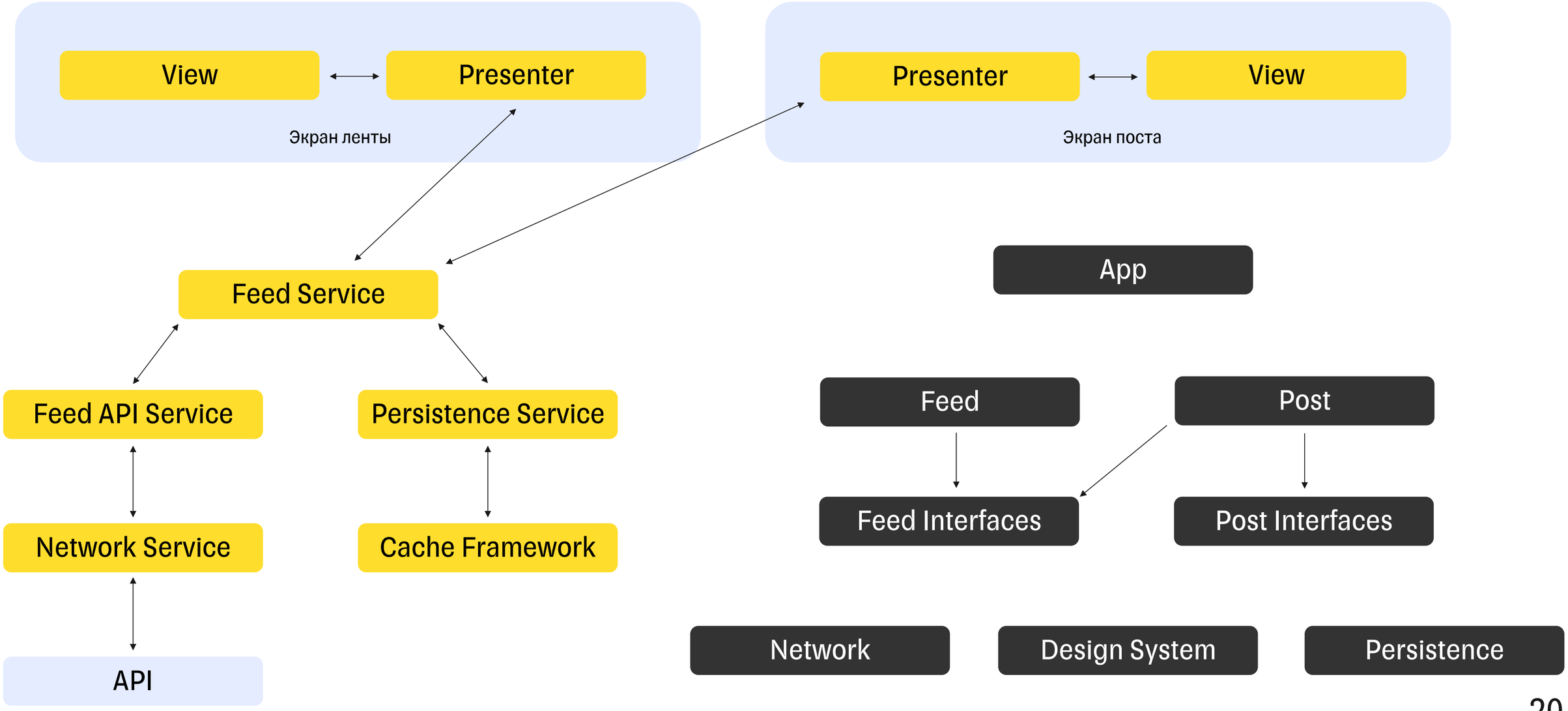
Как модуляризировать проект



Как модуляризировать проект

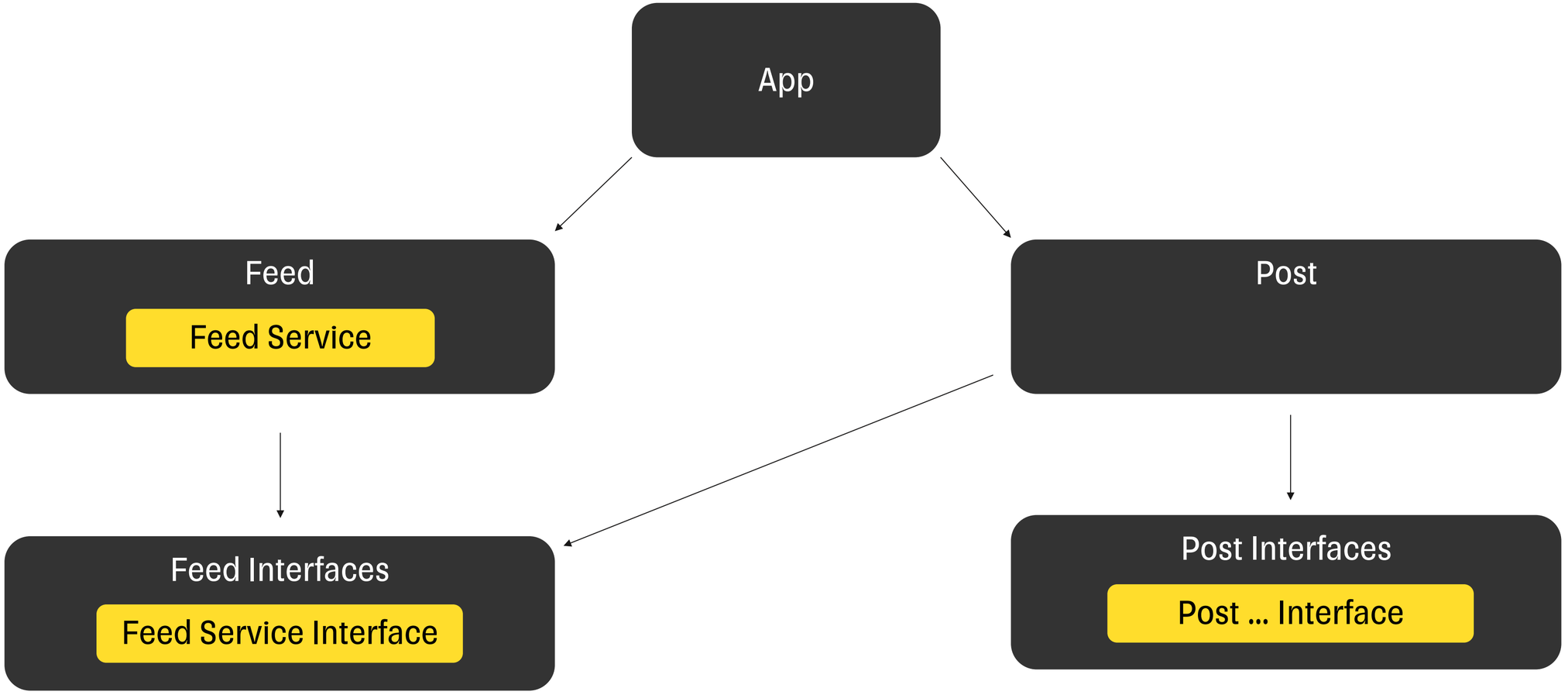


Как модуляризировать проект



Как организовать DI?

Как организовать DI



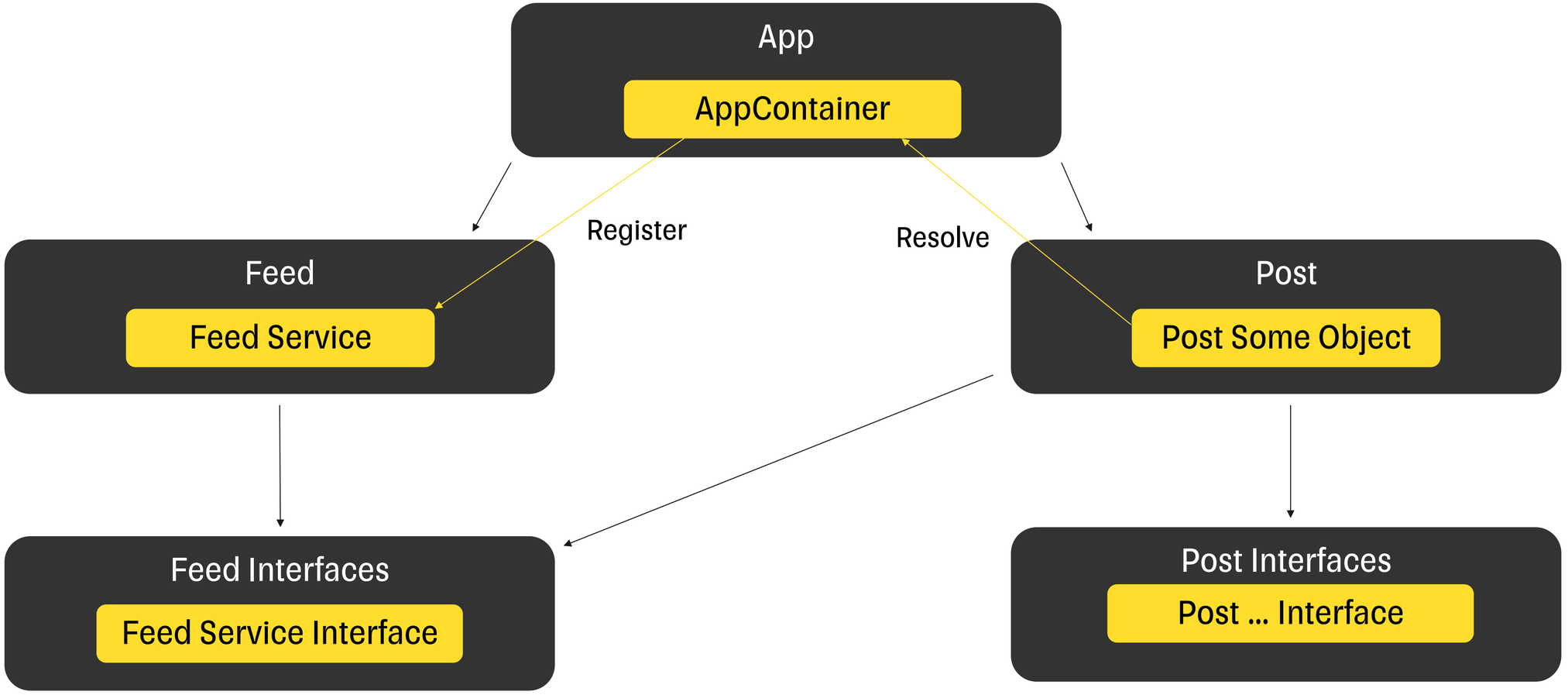
Swinject

Swinject is a lightweight dependency injection framework for Swift.

Dependency injection (DI) is a software design pattern that implements Inversion of Control (IoC) for resolving dependencies. In the pattern, Swinject helps your app split into loosely-coupled components, which can be developed, tested and maintained more easily. Swinject is powered by the Swift generic type system and first class functions to define dependencies of your app simply and fluently.

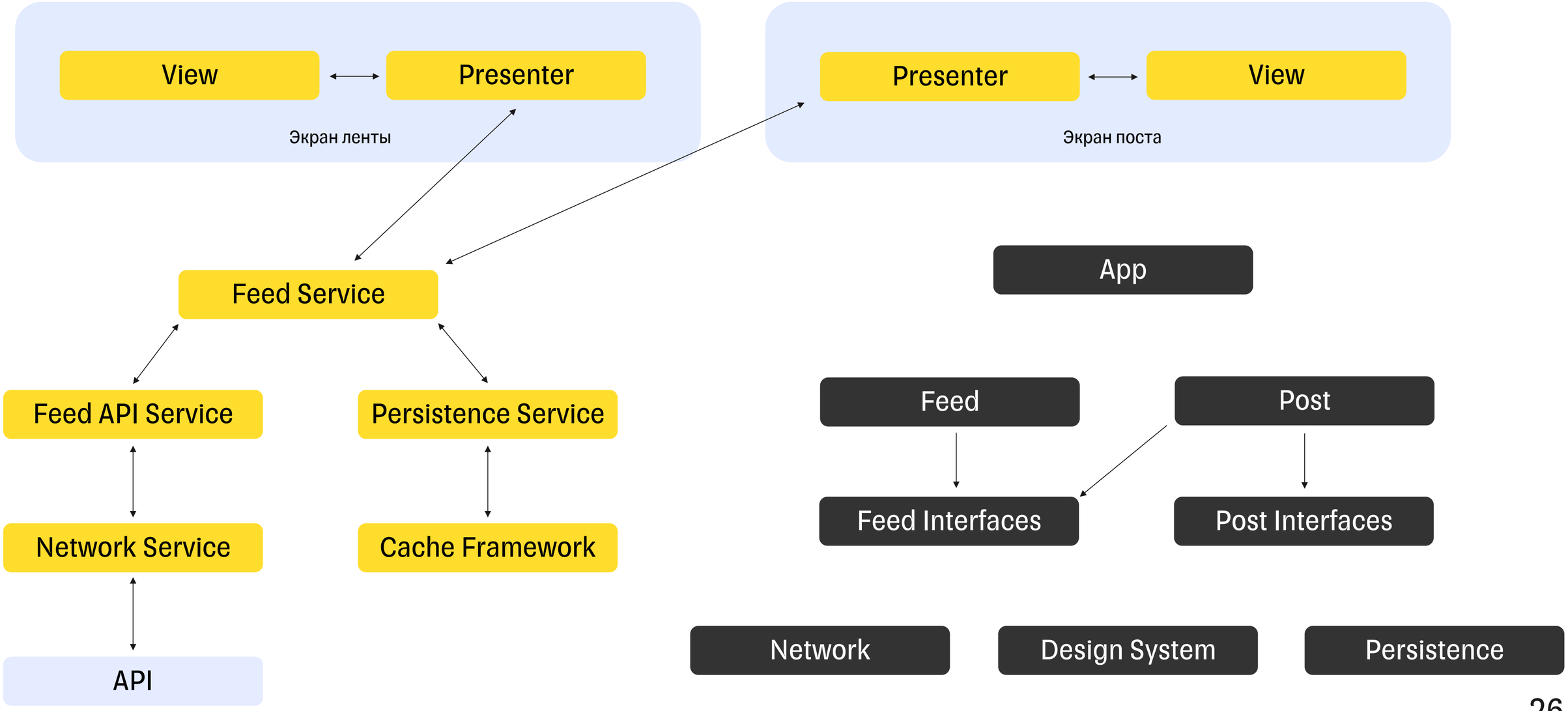
Swinject is maintained by the Faire Wholesale Inc. mobile platform team.

Как организовать DI

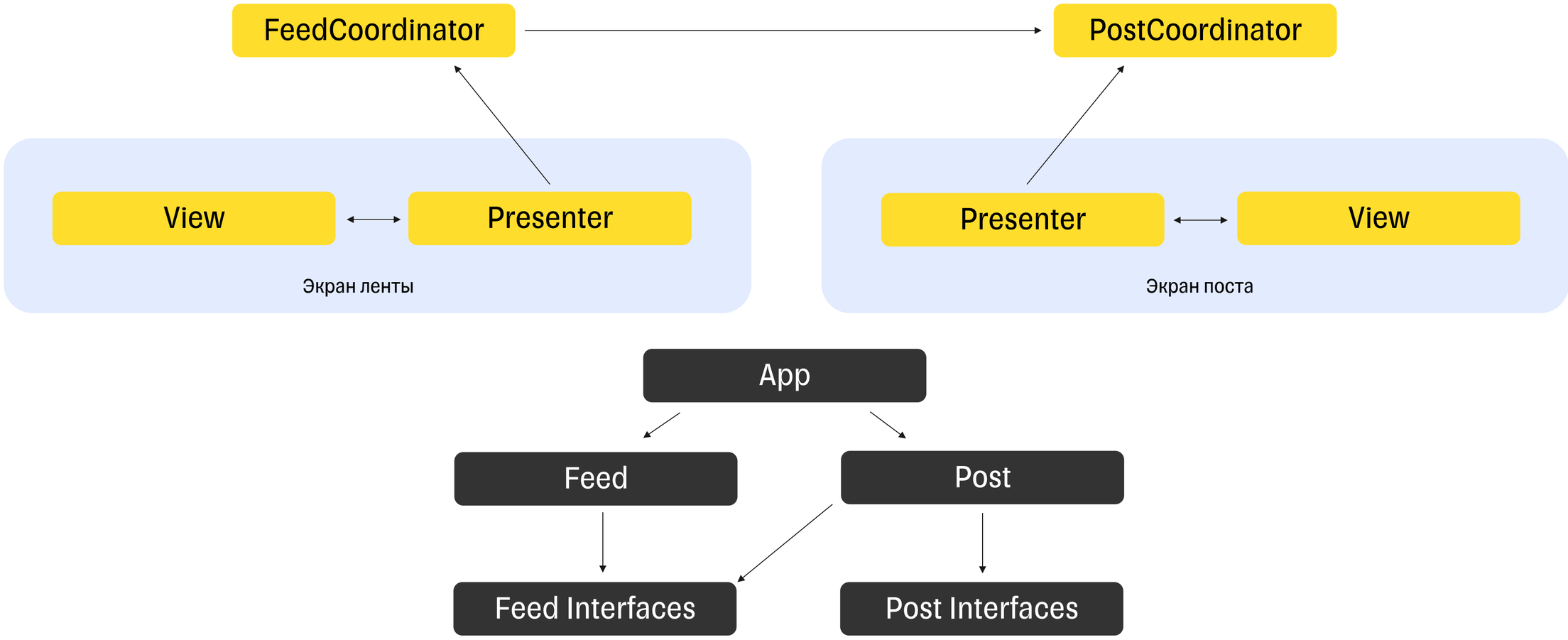


Как организовать навигацию?

Как организовать навигацию



Как организовать навигацию



План

1. Проблемы найма
2. Классический System Design
3. Архитектурная секция
4. **Система оценки**
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

Система оценки

На примере модуляризации

Senior

1. Предложил способ борьбы с горизонтальными связями
2. Обосновал инструмент для модуляризации
3. Обосновал типы линковки модулей

Middle+

1. Обосновал пользу от модуляризации (4-5 пунктов)
2. Выбрал способ модуляризации по слоям и доменам
3. Тесты модуляризированы аналогично коду, описан процесс работы с ними

Middle

1. Предложил способ модуляризации
2. Выбрал инструмент модуляризации
3. Не образовалось циклических зависимостей между модулями

План

1. Проблемы найма
2. Классический System Design
3. Архитектурная секция
4. Система оценки
5. **Какие навыки нужны в масштабном проекте**
6. Поведенческое интервью

Навыки

Перформанс	Авторизация	Безопасность	Архитектура
Надежность	Сеть	DI	Дизайн система
Навигация	Тестирование	CI/CD	UI
Персистенс	Инфраструктура	SDK	Модульность

Навыки

Перформанс	Авторизация	Безопасность	Архитектура
Надежность	Сеть	DI	Дизайн система
Навигация	Тестирование	CI/CD	UI
Персистенс	Инфраструктура	SDK	Модульность

Навыки

Перформанс	Авторизация	Безопасность	Архитектура
Надежность	Сеть	DI	Дизайн система
Навигация	Тестирование	CI/CD	UI
Персистенс	Инфраструктура	SDK	Модульность

ТИНЬКОФФ

Виды тестирования

Белый, черный, серый ящик

Модульные, Интеграционные, End2End

Логика, UI, Перформанс, Snapshot

Инструменты

XCTest

Нативный фреймворк тестирования от Apple

SnapshotTesting

Библиотека для написания snapshot тестов

KIF

Библиотека для написания UI тестов

Appium

Инфраструктура для написания кроссплатформенных тестов

EarlGrey

Библиотека для написания UI тестов

Когда запускать тесты

На Merge Request

Перед релизом SDK или приложения

Ночной прогон

Соотношение тестов

Пирамида

Больше всего модульных тестов, меньше всего end2end

Мороженка

Больше всего end2end, меньше всего модульных тестов

Ромб

Больше всего интеграционных, остальных меньше

Столб

Соотношение примерно одинаковое

Решать типовые проблемы

Flacky тесты

Долгий прогон тестов

Автоматизация моков/стабов

ТИНЬКОФФ


Я ОРАКУЛ, Я ОТВЕЧУ НА ЛЮБОЙ ВОПРОС
Как оптимизировать приложение
НА ЛЮБОЙ, КРОМЕ ЭТОГО

Перформанс персистенса

Batching, faulting

И как они представлены во фреймворках персистенса

Архитектура стека CoreData

Как эффективно выстроить чтение, запись, мерджинг контекстов

Нишевые оптимизации

Например, убрать UndoManager в CoreData

Денормализация

Стандартный прием из теории баз данных

Перформанс scroll или коллекции

Color Blending

Offscreen Rendering

Ресайзинг картинок

Оптимизация стадии layout

Оптимизация на уровне приложения

Старт приложения

Размер приложения

Хранение данных

Использование батареи

Профилирование

Hitches

Memory leaks

Time profiler

Мониторинг

MetricKit

Firebase

ТИНЬКОФФ

Сетевые решения

Websocket

GraphQL

Long polling

MQTT

Оптимизации

Бинарный формат сериализации

HTTP 3

ETag

Безопасность

- SSL-pinning, сертификаты
- Передача чувствительной информации
- Авторизация, аутентификация и идентификация

ТИНЬКОФФ

Хранение данных

Как кешировать изображения

Как хранить чувствительные данные

Обосновать CoreData, Realm, SQLite или файлы

ТИНЬКОФФ

SDK

- Публичный API SDK
- SemVer, Binary/Module Stability
- Как собирать и публиковать

ТИНЬКОФФ


Ты красишь кнопки
Я проектирую дизайн систему
Мы на разных уровнях

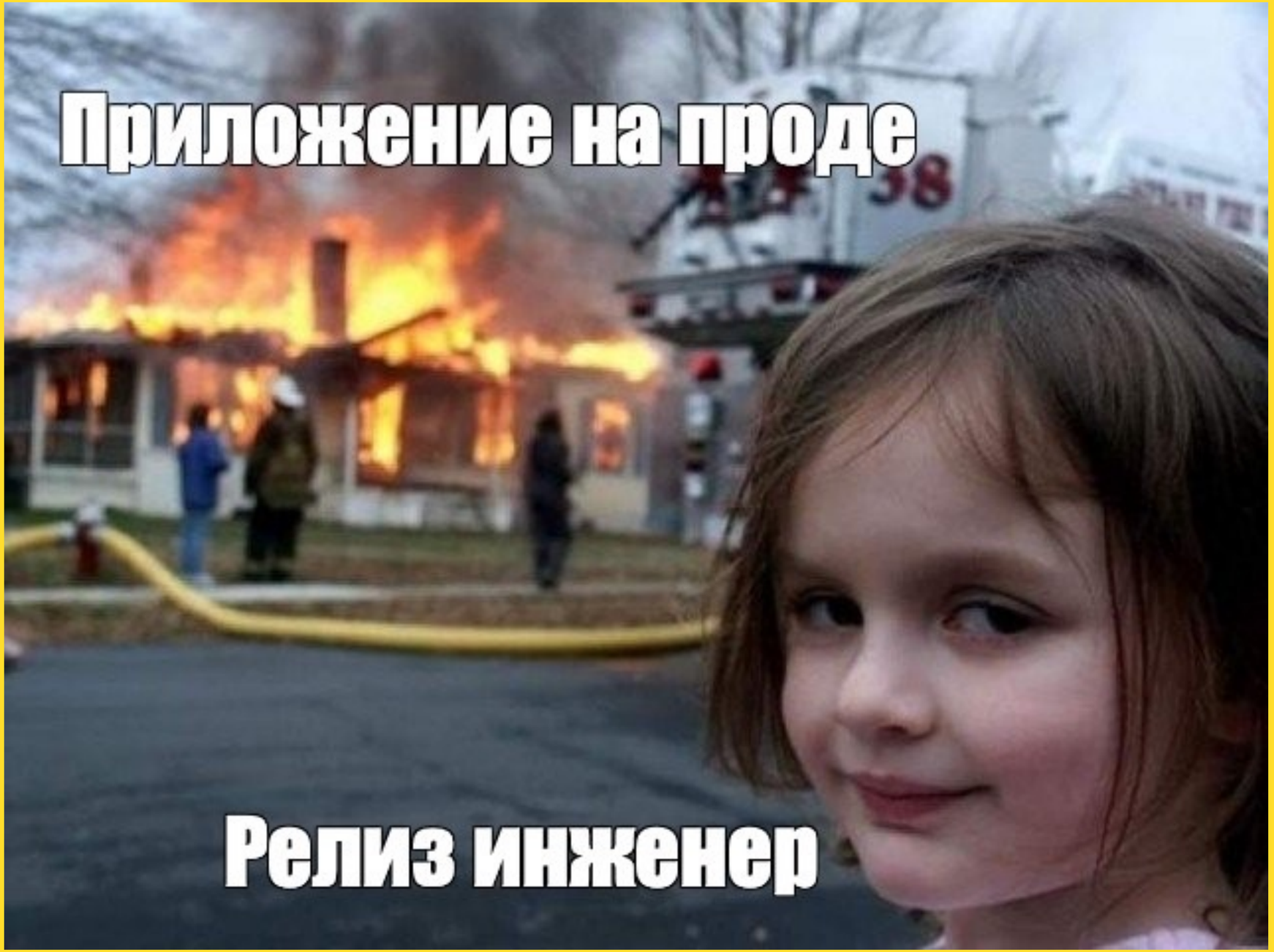
UI решения

Как устроена дизайн система,
из каких частей она состоит

DiffableDataSource

UICollectionLayout/LazyGrid

PinLayout, Texture, YogaKit


Приложение на проде
Релиз инженер

Надежность

Как следить за надежностью всего приложения

Как следить за надежностью отдельной фичи

Какие точки отказа стоит мониторить

Какие инструменты для этого понадобятся

Самопроверка для любого решения

Масштабирование

Как решение работает для 10+ команд или 10+ разработчиков

Параллельность

Насколько решение позволяет людям работать независимо

Тестируемость

Насколько просто тестировать сущности

Экосистемность

Насколько легко переиспользовать в нескольких приложениях

Best practices

SOLID, Clean Architecture, отсутствие GodObject и т.д.

План

1. Проблемы найма
2. Классический System Design
3. Архитектурная секция
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. **Поведенческое интервью**

Leadership Principles

We use our Leadership Principles every day, whether we're discussing ideas for new projects or deciding on the best way to solve a problem. It's just one of the things that makes Amazon peculiar.

ДНК Компании

01

Мы считаем компанию своей

Учимся принимать решения, работать сообща, брать ответственность и видеть, как наши действия влияют на цели компании

02

Мы работаем для клиента

Учимся создавать топовый клиентский сервис, выяснять потребности, собирать и учитывать обратную связь

03

Мы любим то, что делаем

Учимся быть проактивным, оставаться вдохновленным. Нам не все равно на результат нашей работы

04

Мы задаем тренды

Стараемся видеть тренды и использовать их во благо компании, мыслить нестандартно и анализировать опыт

05

Мы в одной лодке

Прокачиваем навык переговоров, поддерживаем коллег, налаживаем контакт и добиваемся целей на встречах

Мы любим то, что делаем

Чем он гордится больше всего?

Где он допустил ошибки и как бы их сейчас исправил?

Мы в одной лодке

Расскажи о конфликтах с коллегами, и как ты с ними их разрешал?

Расскажи, как ты помогаешь своим коллегам по работе?

Мотивация

Важно понимать, а не бежите ли вы от самих себя с одной работы на другую

Действительно ли вы заинтересованы дать что-то компании, а не только получить

Деньги это тоже нормальная мотивация, но надо правильно это объяснить

Leadership Principles

We use our Leadership Principles every day, whether we're discussing ideas for new projects or deciding on the best way to solve a problem. It's just one of the things that makes Amazon peculiar.

Итоги

1. Проблемы найма
2. Классический System Design
3. Архитектурная секция
4. Система оценки
5. Какие навыки нужны в масштабном проекте
6. Поведенческое интервью

ТИНЬКОФФ

Спасибо!