

3 главных недостатка FSD после 3 лет использования

Евгений Паромов, Evrone

Поговорим о фронтенде

- 6 лет опыта во Front-end

Поговорим о фронтенде

- 6 лет опыта во Front-end
- Автор Youtube канала 11к человек

Поговорим о фронтенде

- 6 лет опыта во Front-end
- Автор Youtube канала 10к человек
- Основатель  evolution-community

Поговорим о фронтенде

- 6 лет опыта во Front-end
- Автор Youtube канала 10к человек
- Основатель  evolution-community
- Работаю в evrone

Поговорим о фронтенде

- 6 лет опыта во Front-end
- Автор Youtube канала 10к человек
- Основатель  evolution-community
- Работаю в evrone
- Заказная разработка: много разных проектов

Поговорим о фронтенде

- 6 лет опыта во Front-end
- Автор Youtube канала 10к человек
- Основатель  evolution-community
- Работаю в evrone
- Заказная разработка: много разных проектов
- 3 года я работаю с FSD

За это время я действительно
полюбил FSD

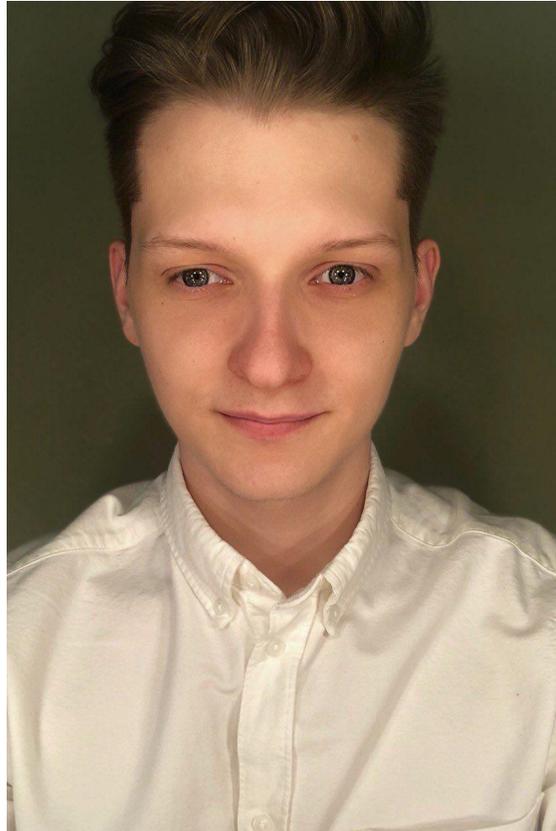
Но сегодня мы поговорим о
его **недостатках**

Как я познакомился с FSD

Как я познакомился с FSD

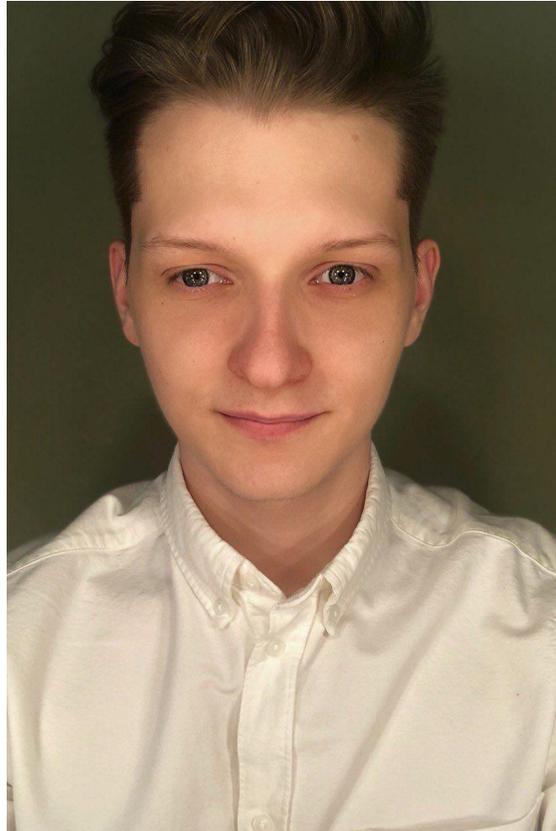


Как я познакомился с FSD



Тимлидил
в Outstaf

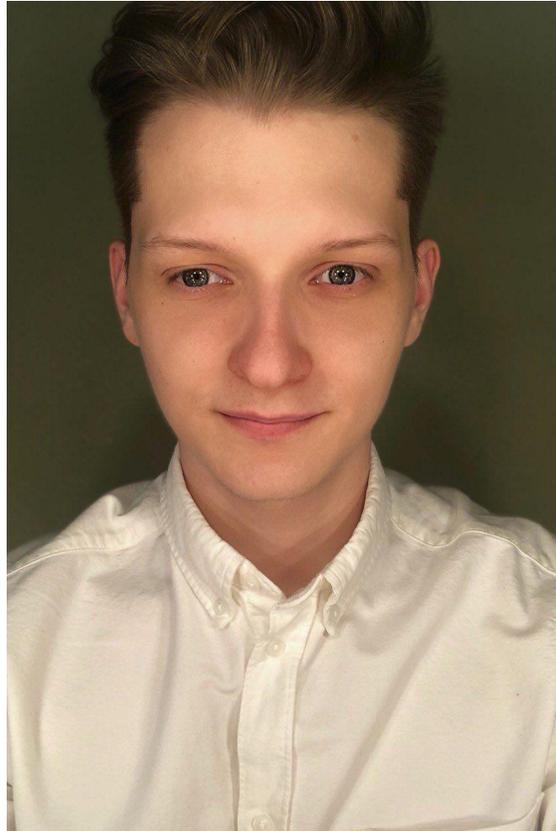
Как я познакомился с FSD



Тимлидил
в Outstaf

Clean
Architecture ❤️

Как я познакомился с FSD

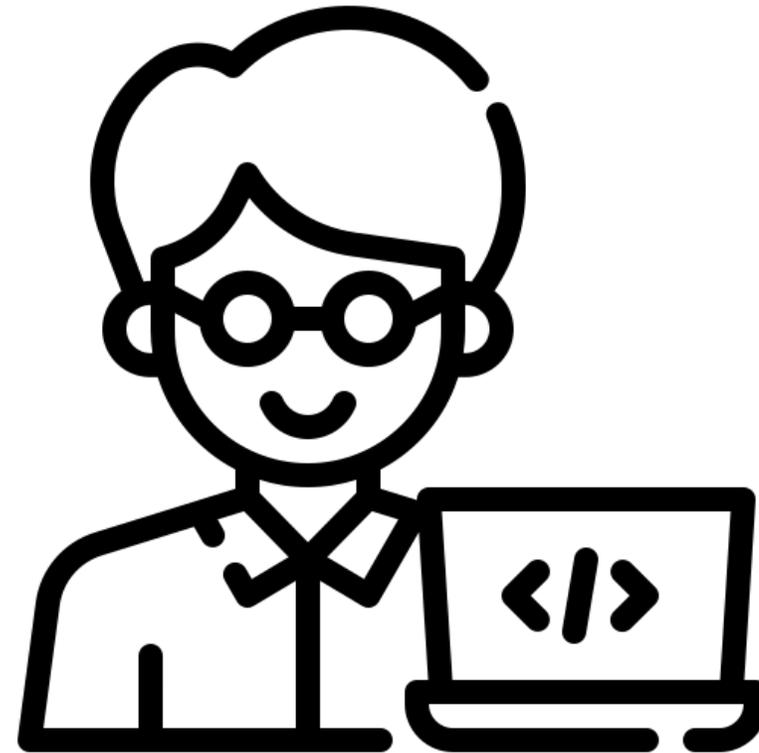
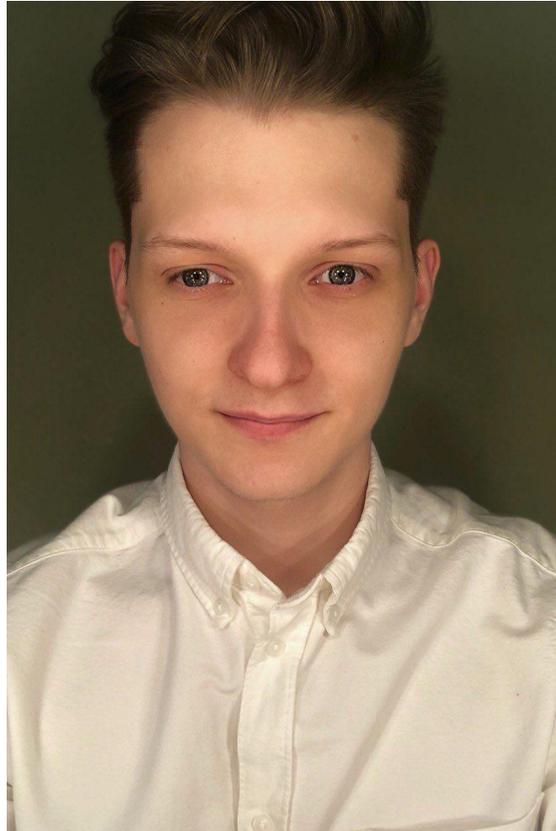


Тимлидил
в Outstaf

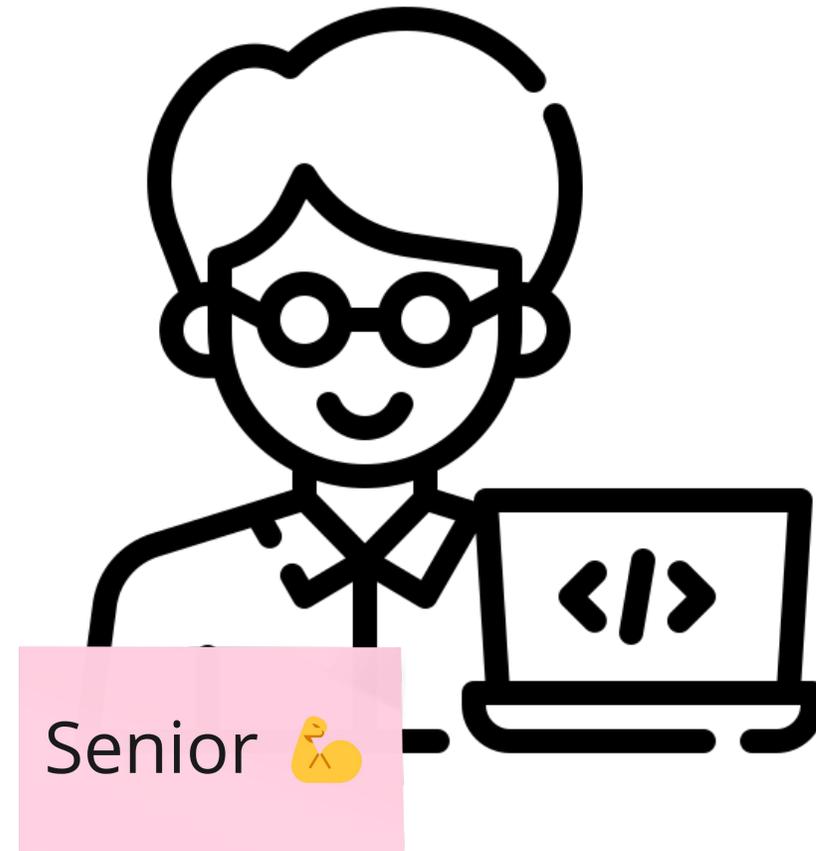
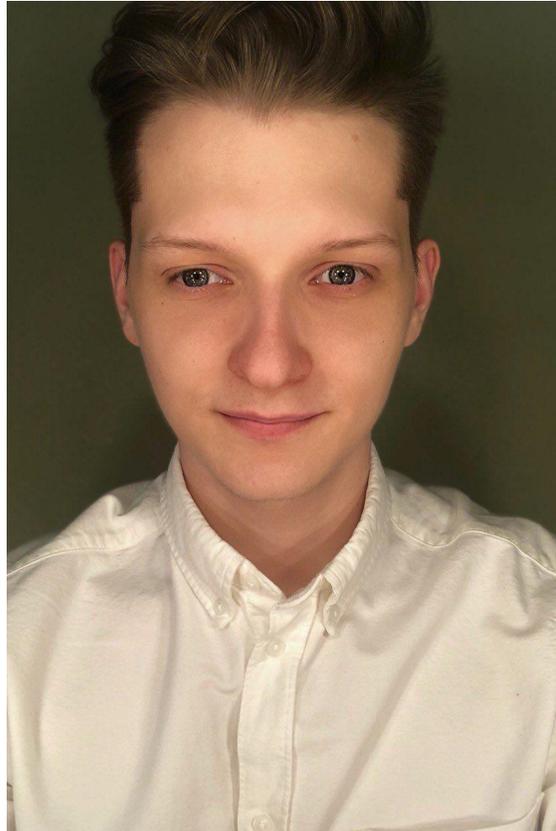
Clean
Architecture ❤️

DI + React
= 🤖

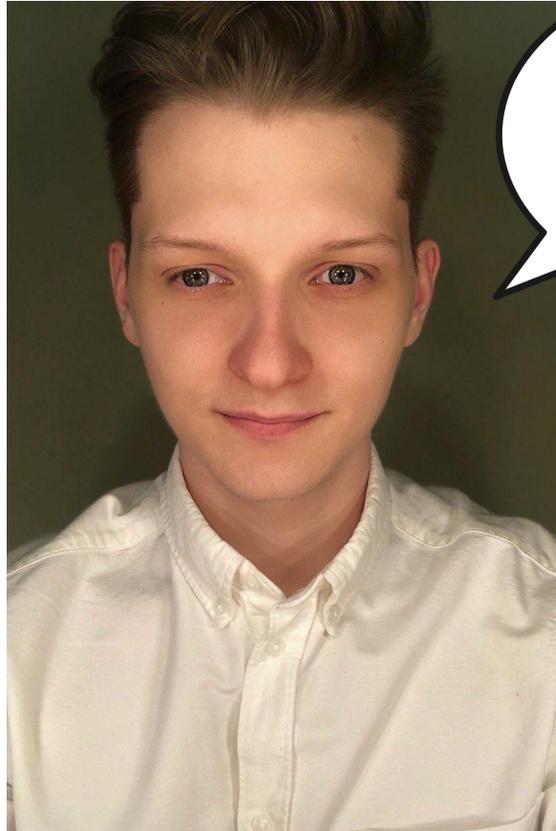
Как я познакомился с FSD



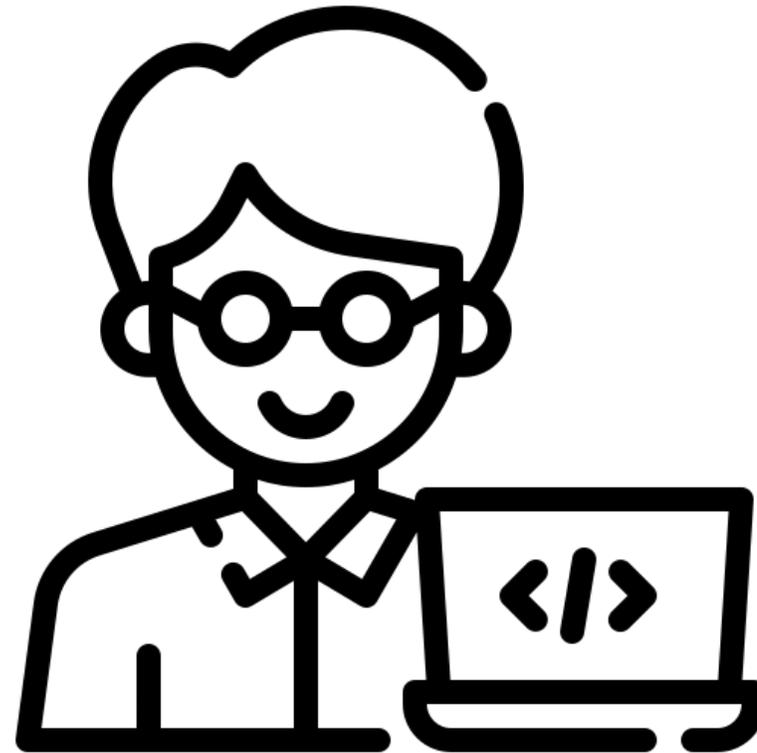
Как я познакомился с FSD



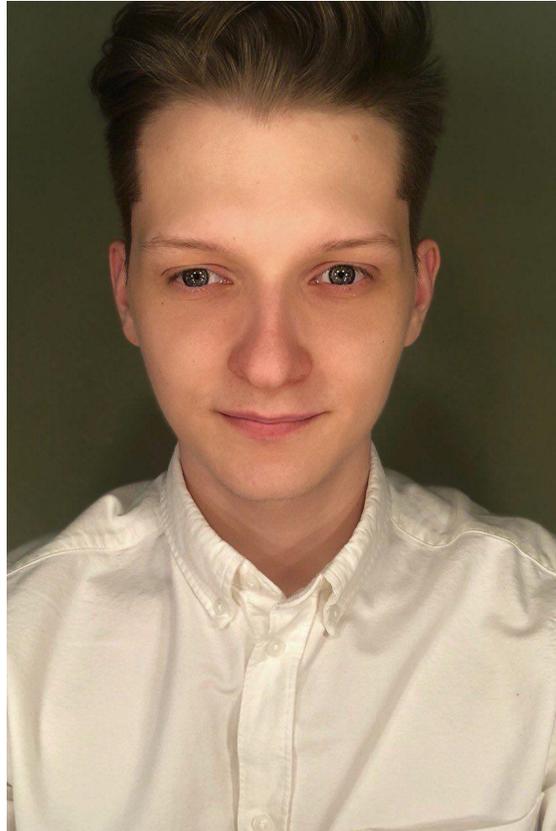
Как я познакомился с FSD



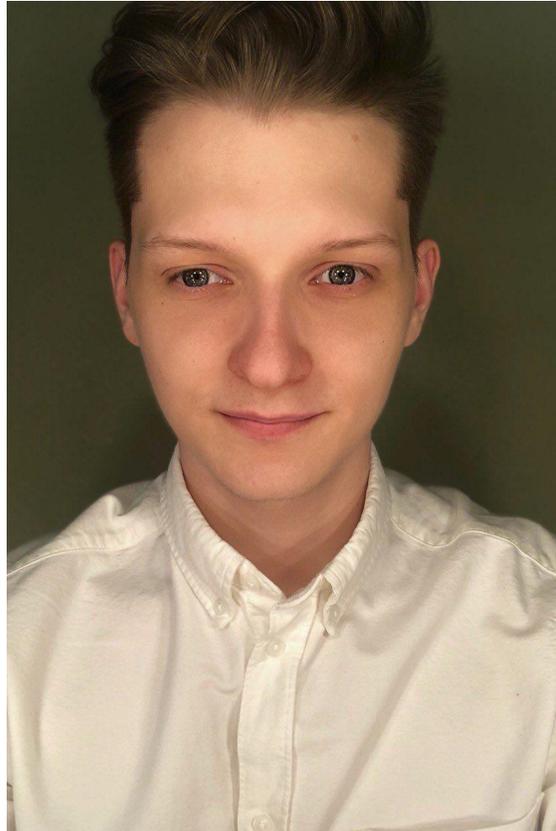
Вот **clean
architecture**
это да...



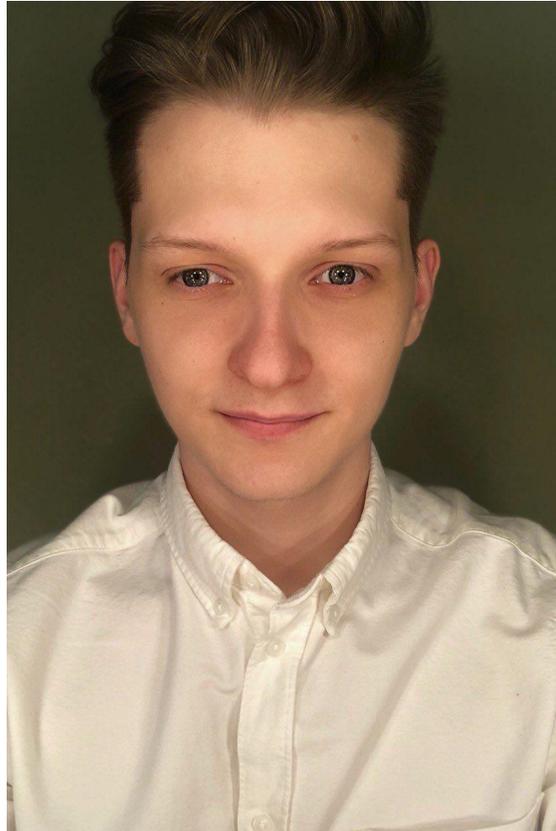
Как я познакомился с FSD



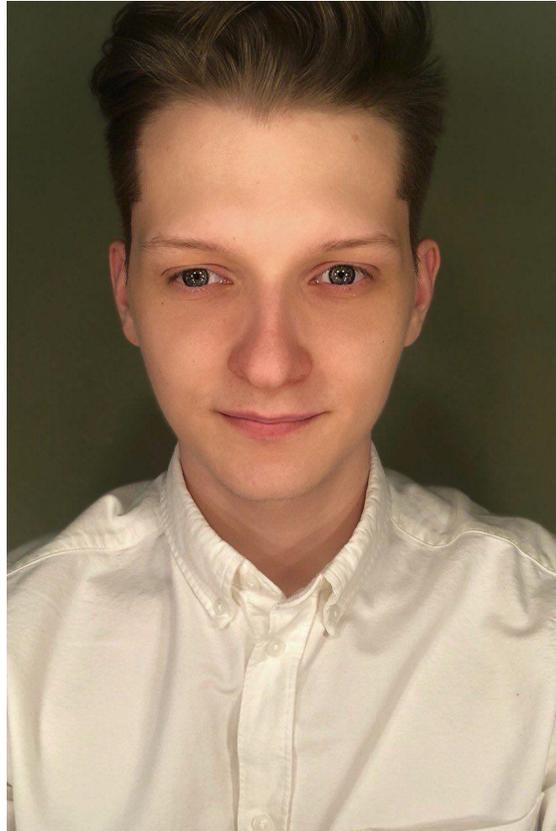
Сначала я пошёл в telegram



Сначала я пошёл в telegram



Потом пошёл в доку



Документация



Feature-Sliced Design (FSD) — это архитектурная методология для проектирования frontend-приложений. Проще говоря, это свод правил и соглашений по организации кода. Главная цель методологии — сделать проект понятным и структурированным, особенно в условиях регулярного изменения требований бизнеса.

И что я там увидел

ОСНОВЫ FSD

ОСНОВЫ FSD

- FSD - Архитектурная методология

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?

Архитектура

Архитектура

- Набор важнейших решений по организации системы

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
 - Определяем требования

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
 - Определяем требования
 - Принимаем решения

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
 - Определяем требования
 - Принимаем решения
 - Смотрим как оно работает

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике
 - Определение требований – сложно

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике
 - Определение требований – сложно
 - Принятие решений – еще сложнее

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике
 - Определение требований – сложно
 - Принятие решений – еще сложнее
 - Делаем по наитию



Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике
- Архитектура есть в любом проекте!

Архитектура

- Набор важнейших решений по организации системы
- В идеальном мире
- На практике
- Архитектура есть в любом проекте!
- Большой комок грязи – это тоже архитектура



Архитектурная методология

Архитектурная методология

- Архитектура – это сложно

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
 - Понятный

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
 - Понятный
 - Надёжный

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
 - Понятный
 - Надёжный
 - Переиспользуемый

Архитектурная методология

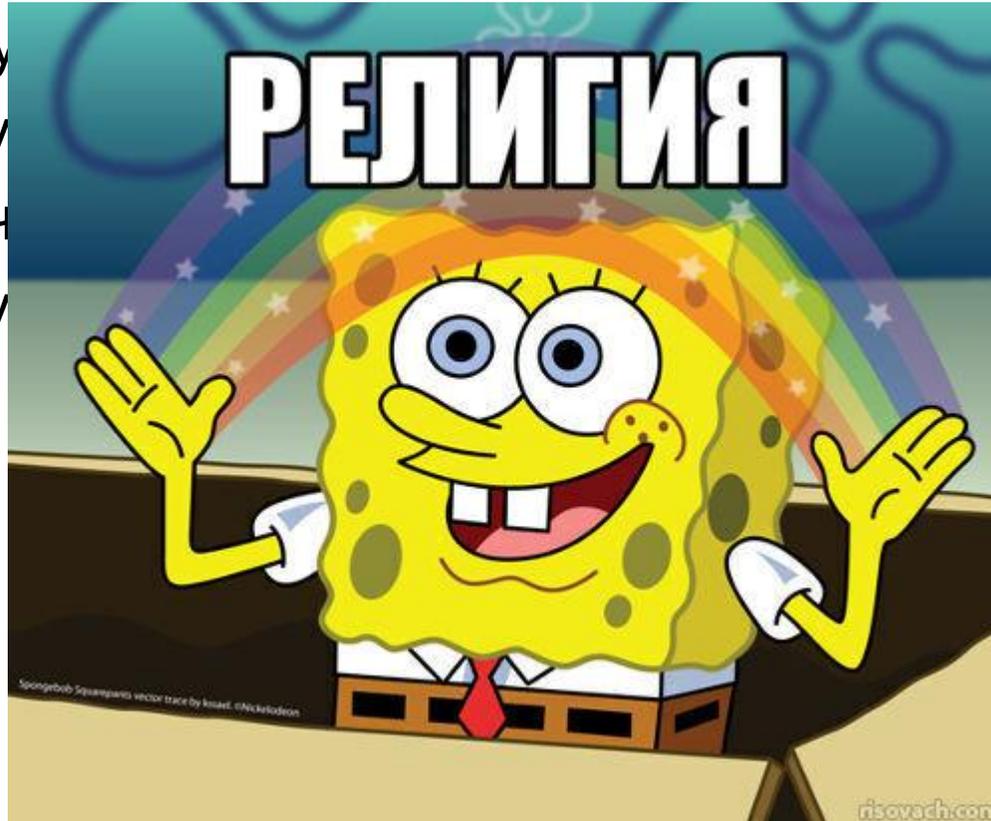
- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ

Архитектурная методология

- Архитекту
- Существу
- Как получ
- Существу



ий”

архитектора?

Архитектурная методология

- Архитекту
- Существу
- Как получ
- Существу



ий”

архитектора?

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
 - **Простой** способ, сделать **достаточно** хорошо

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
 - **Простой** способ, сделать **достаточно** хорошо
 - Как эвристические алгоритмы

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
 - **Простой** способ, сделать **достаточно** хорошо
 - Как эвристические алгоритмы
 - Примеры: SRP, DIP, DRY

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
 - **Простой** способ, сделать **достаточно** хорошо
 - Как эвристические алгоритмы
 - Примеры: SRP, DIP, DRY
 - Мы можем даже не понимать, как оно работает!

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
- Архитектурная методология - набор эвристик

Архитектурная методология

- Архитектура – это сложно
- Существует набор “стандартных требований”
- Как получить хорошую Архитектуру, не привлекая архитектора?
- Существует проверенный веками способ
- Эвристика
- Архитектурная методология - набор эвристик
- Набор достаточно хороших архитектурных решений

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений
- FSD сконцентрирован на поддерживаемом коде

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений
- FSD сконцентрирован на поддерживаемом коде
 - Понятность

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений
- FSD сконцентрирован на поддерживаемом коде
 - Понятность
 - Надежность

ОСНОВЫ FSD

- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений
- FSD сконцентрирован на поддерживаемом коде
 - Понятность
 - Надежность
 - Переиспользуемость

ОСНОВЫ FSD

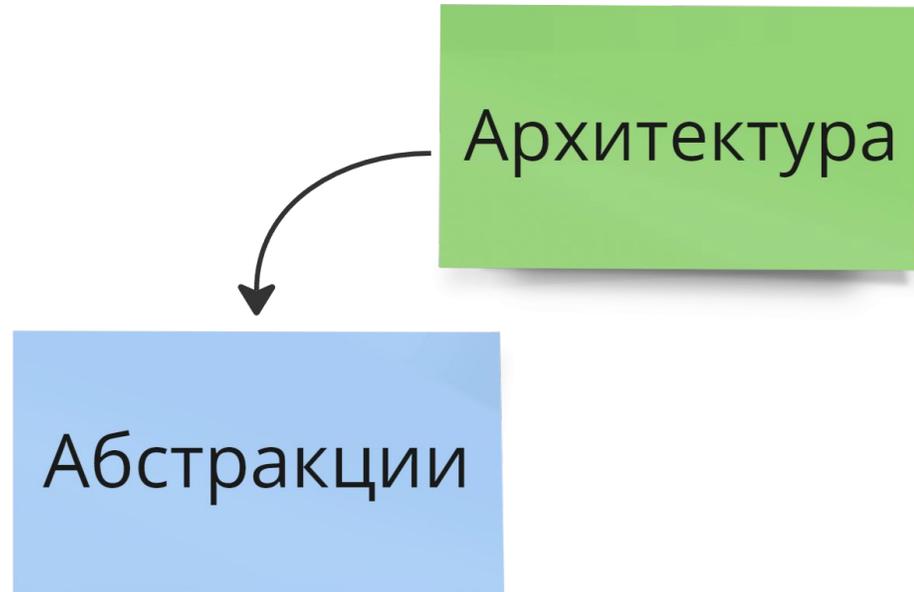
- FSD - Архитектурная методология
- В чём разница между архитектурой и методологией?
- FSD - набор **достаточно** хороших архитектурных решений
- FSD сконцентрирован на поддерживаемом коде
 - Понятность
 - Надежность
 - Переиспользуемость
- Давайте повторим, архитектурные решения FSD

ОСНОВЫ FSD

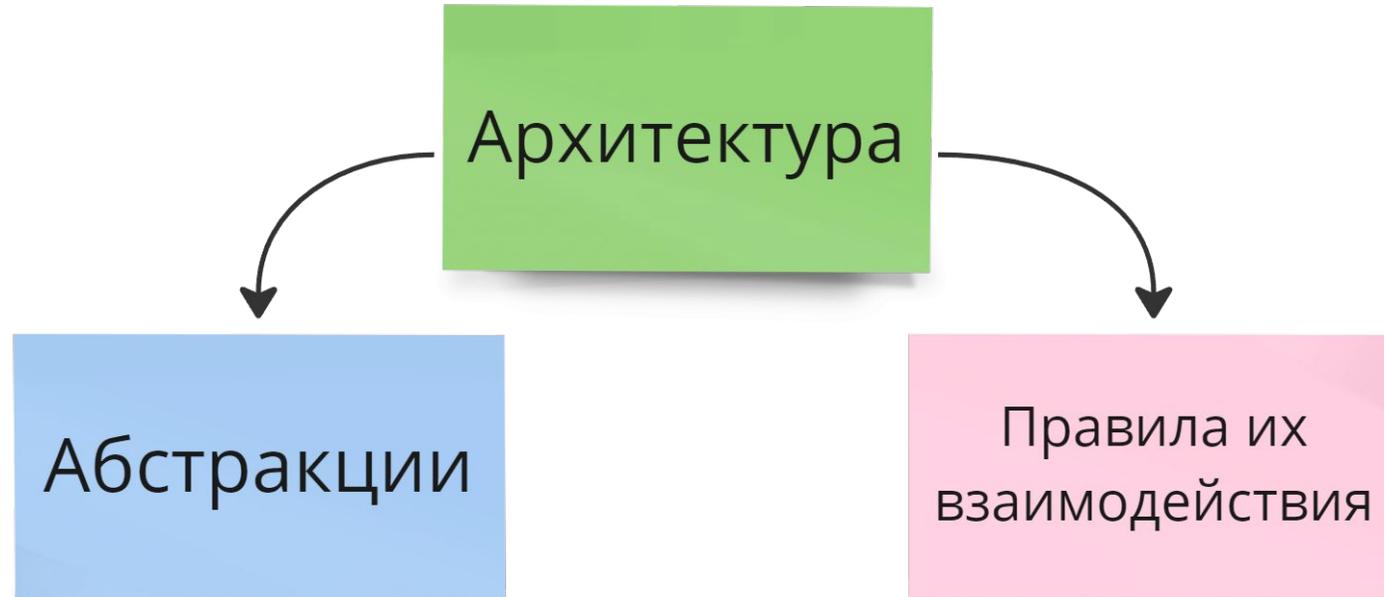


Архитектура

ОСНОВЫ FSD

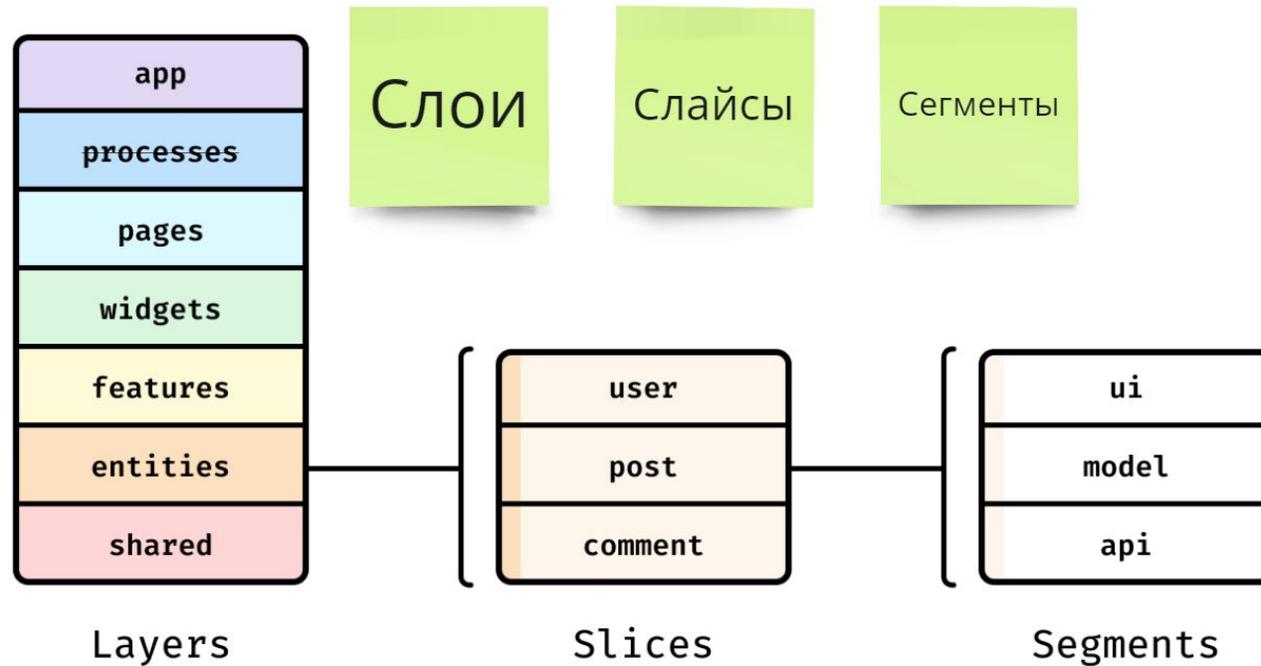


ОСНОВЫ FSD

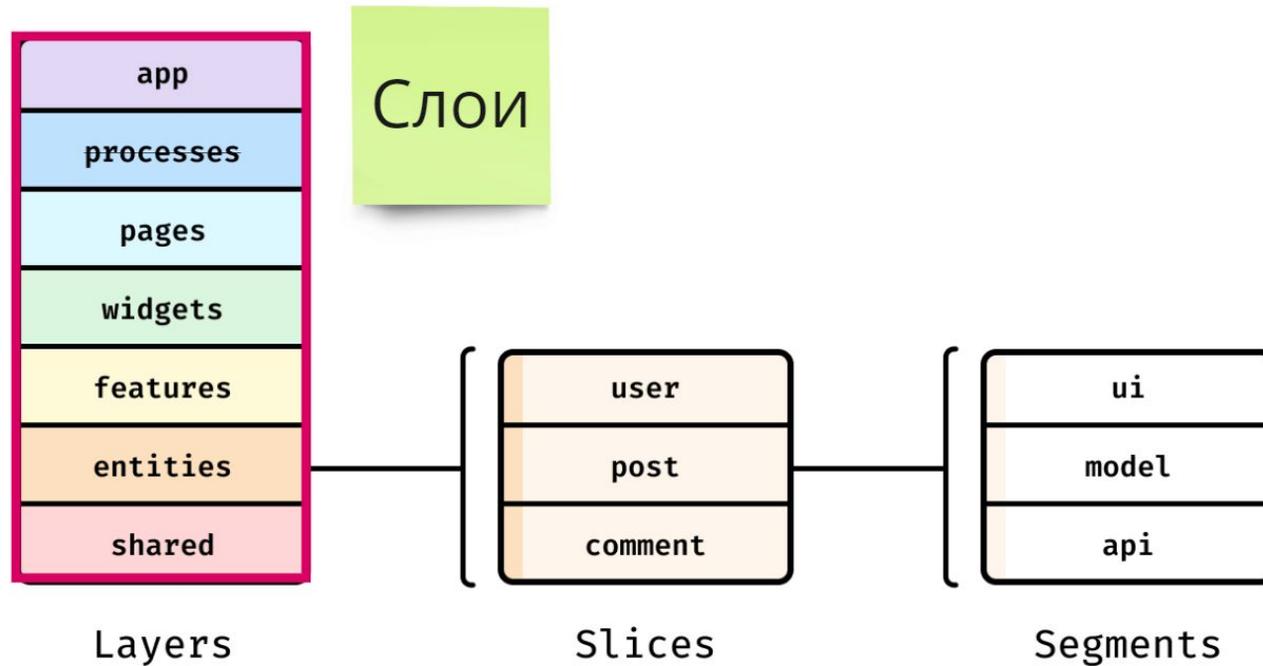


Абстракции в FSD

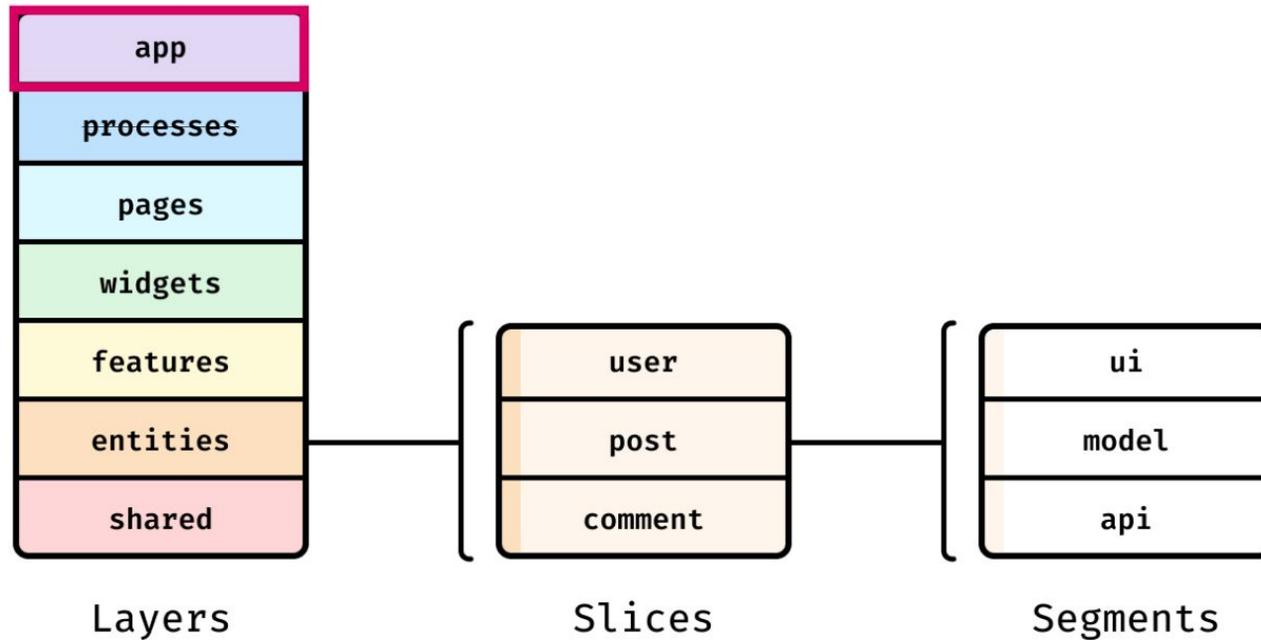
Абстракции в FSD



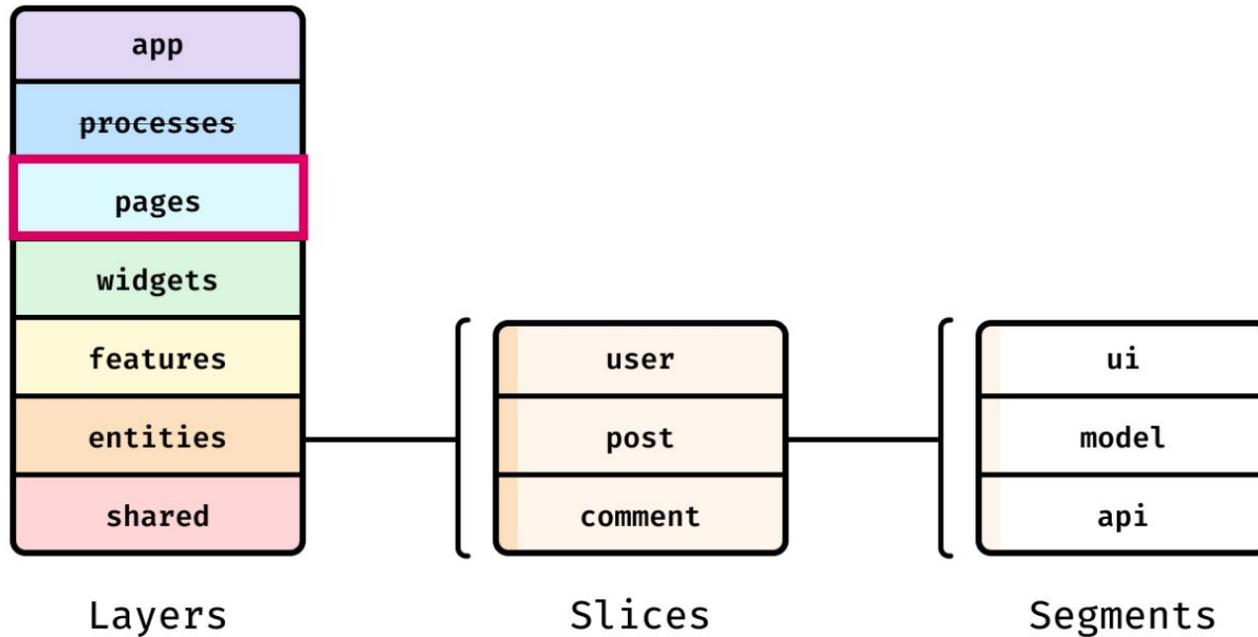
Абстракции в FSD



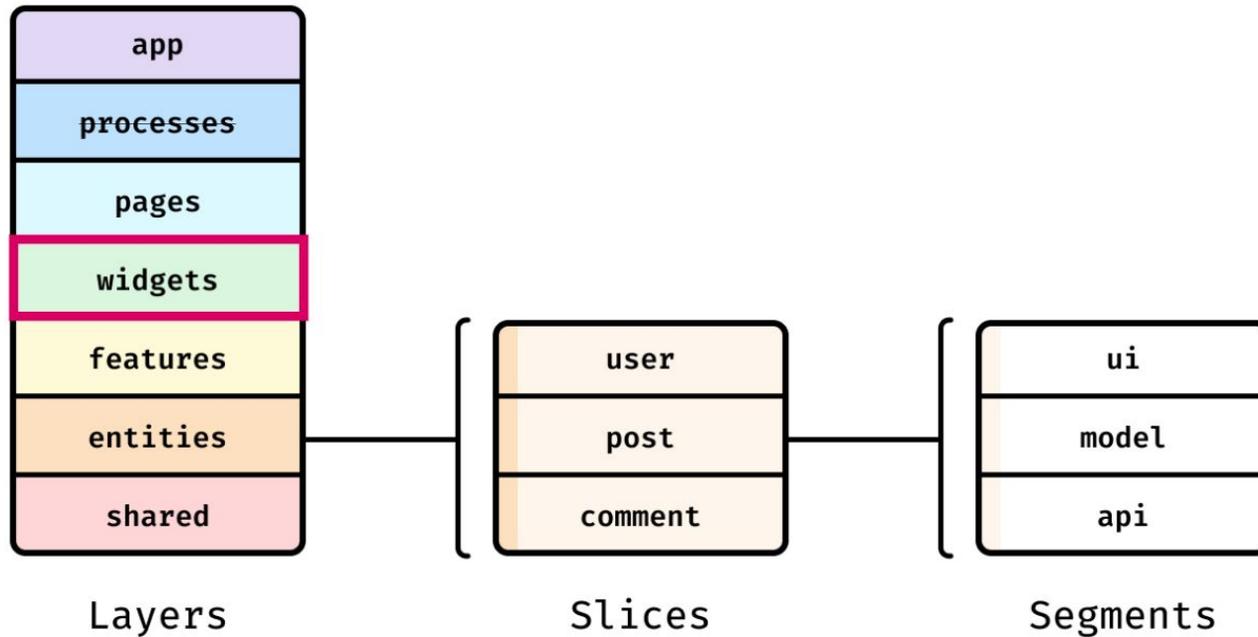
Абстракции в FSD



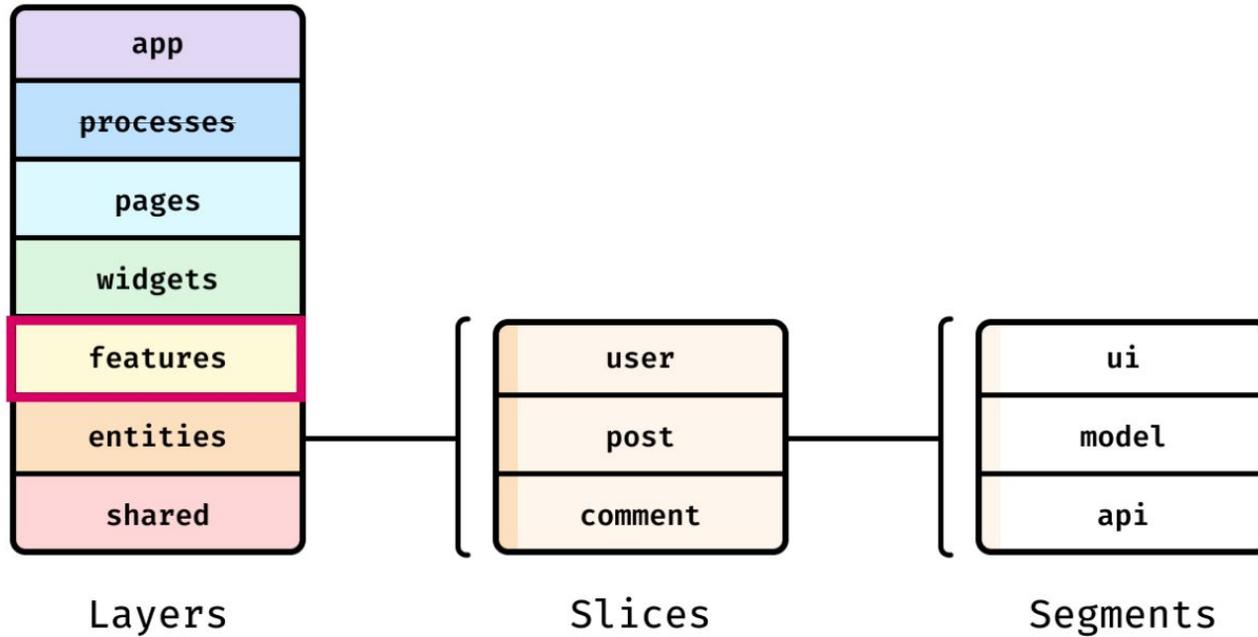
Абстракции в FSD



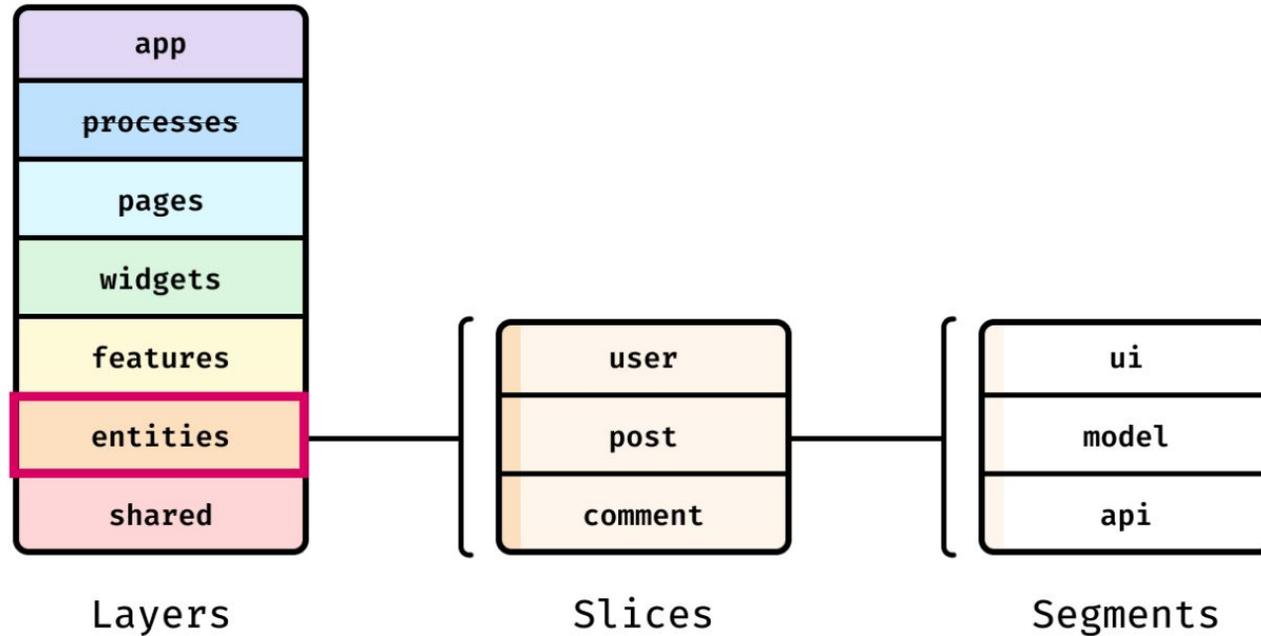
Абстракции в FSD



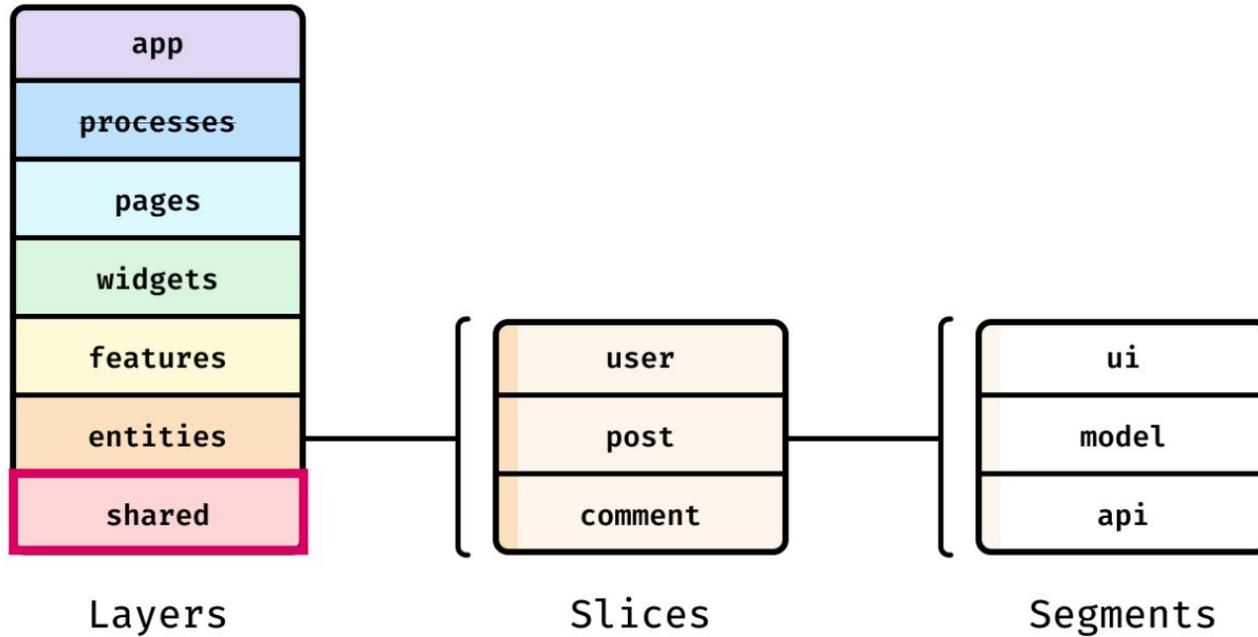
Абстракции в FSD



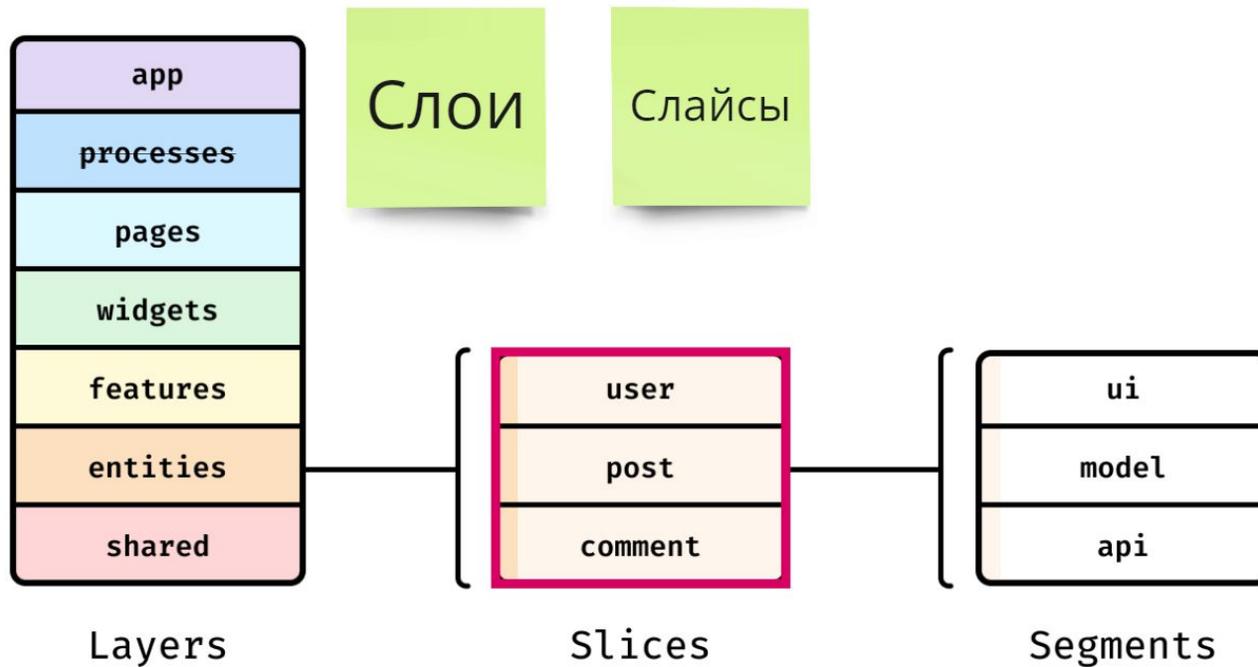
Абстракции в FSD



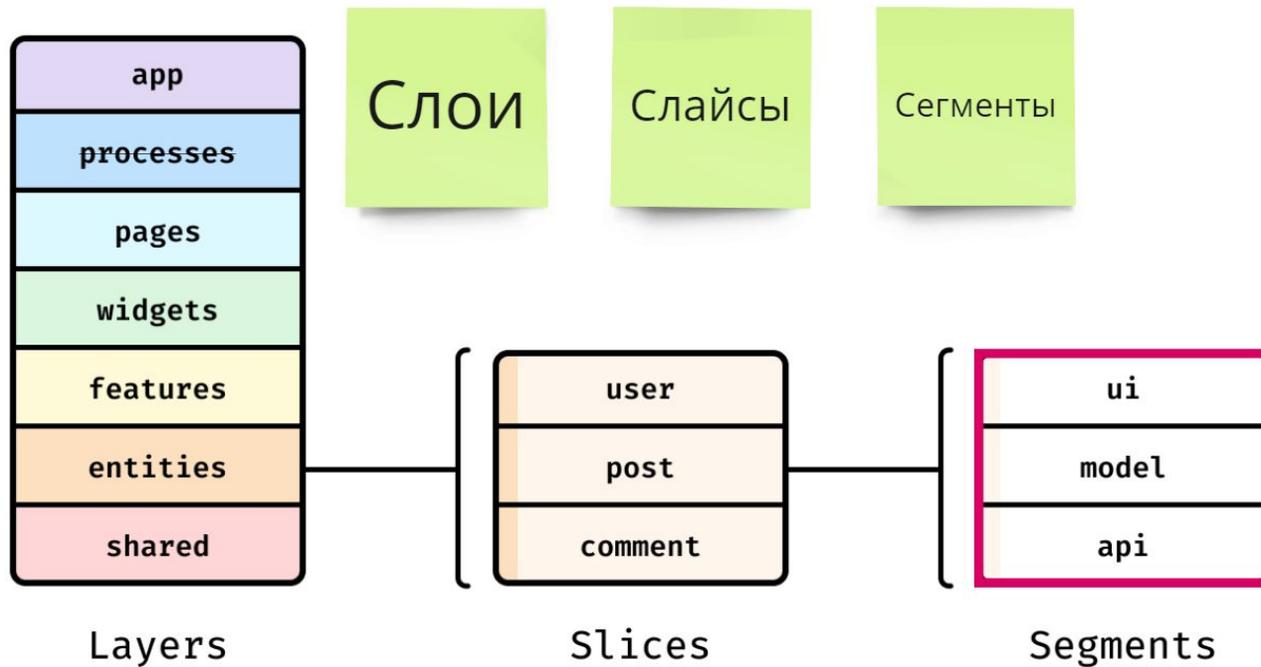
Абстракции в FSD



Абстракции в FSD

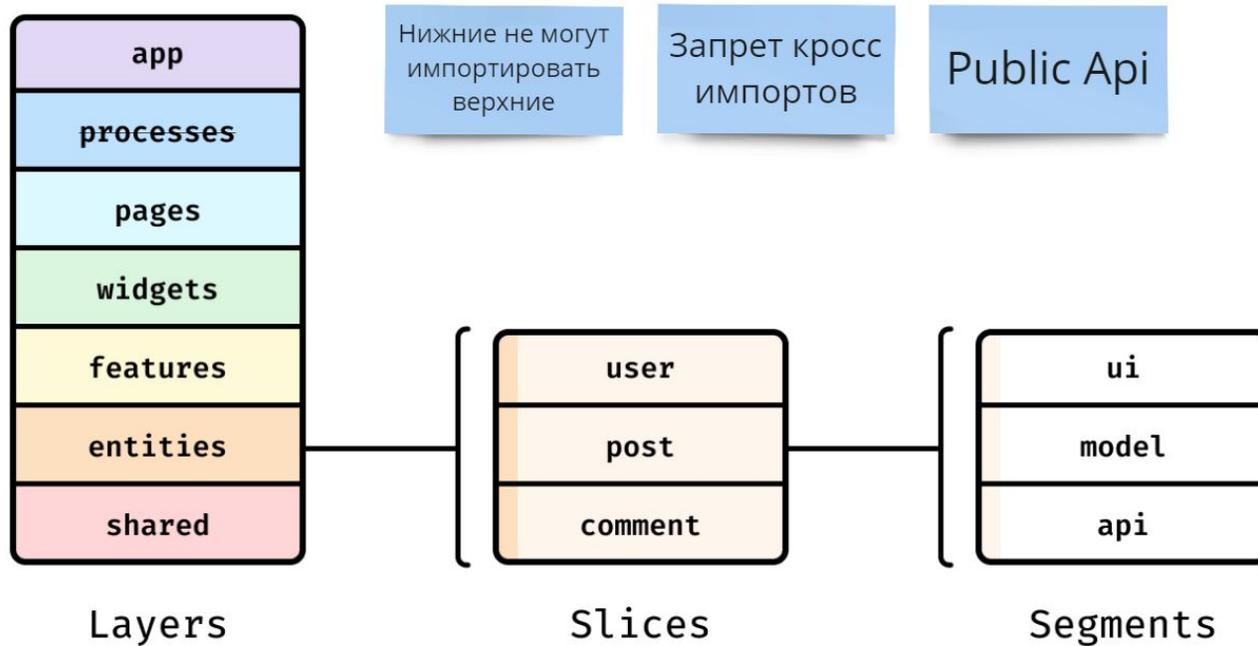


Абстракции в FSD

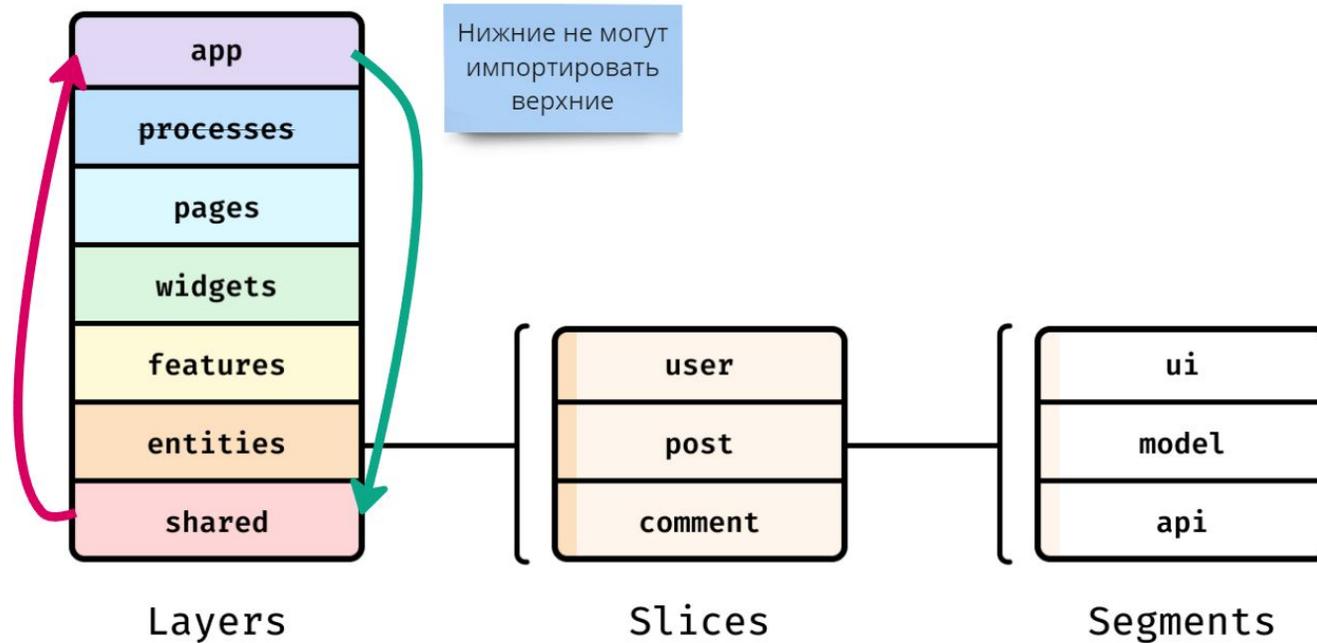


Правила в FSD

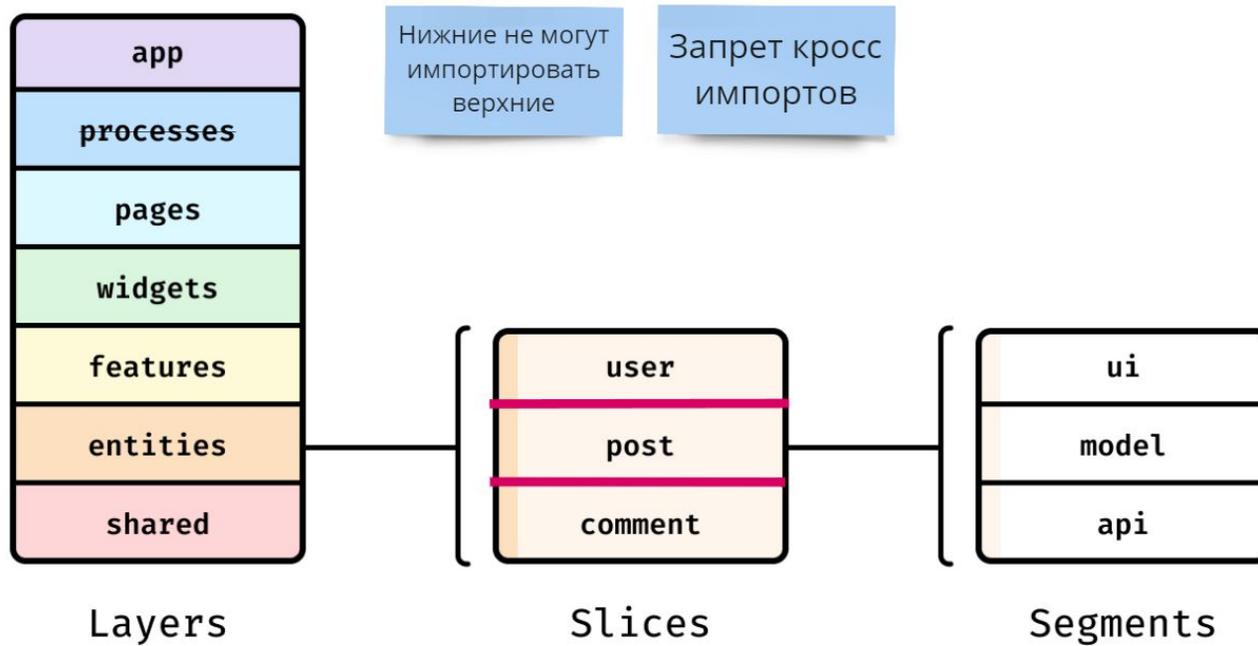
Правила в FSD



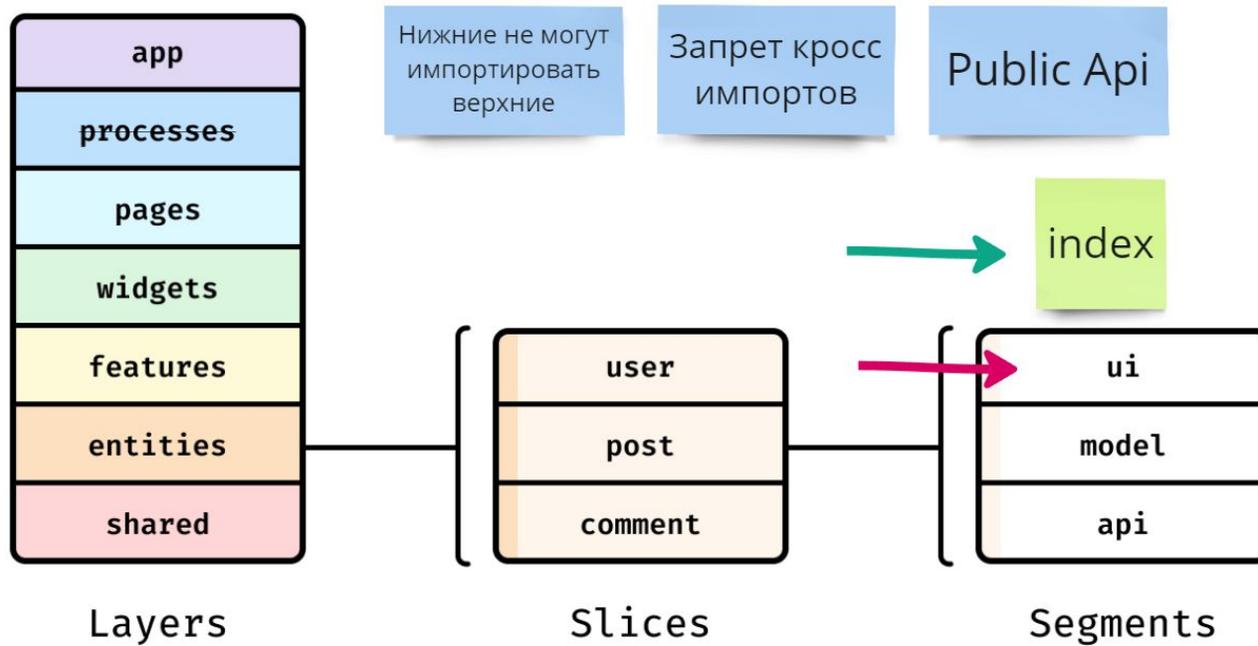
Правила в FSD



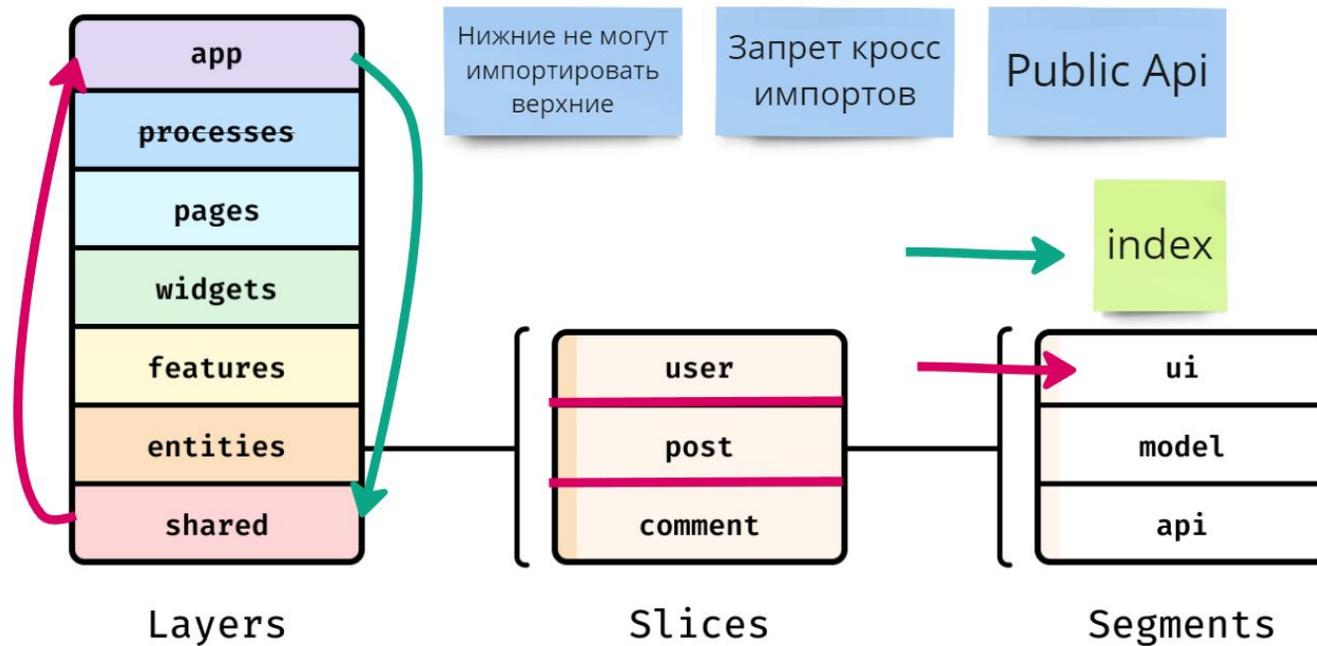
Правила в FSD



Правила в FSD



Правила в FSD



Почему я полюбил FSD

Почему я люблю FSD

- Миддл с FSD напишет лучше

Почему я люблю FSD

- Миддл с FSD напишет лучше
- Достаточно прост

Почему я люблю FSD

- Миддл с FSD напишет лучше
- Достаточно прост
- Адаптирован под Front-end

Почему я люблю FSD

- Миддл с FSD напишет лучше
- Достаточно прост
- Адаптирован под Front-end
- Популяризировал тему архитектуры во Front-end

Почему я люблю FSD

- Миддл с FSD напишет лучше
- Достаточно прост
- Адаптирован под Front-end
- Популяризировал тему архитектуры во Front-end
- Вынес знания об архитектуре на уровень “стека”

Почему я люблю FSD

- Миддл с FSD напишет лучше
- Достаточно прост
- Адаптирован под Front-end
- Популяризировал тему архитектуры во Front-end
- Вынес знания об архитектуре на уровень “стека”
- Поэтому я начал использовать FSD во всех проектах

Разочарование

Почему я разочаровался

Почему я разочаровался

- FSD - набор **достаточно хороших** решений

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
 - Оказывается, есть что-то кроме components

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
 - Оказывается, есть что-то кроме components
 - Можно отделить инфраструктуру от прикладного кода

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
 - Оказывается, есть что-то кроме components
 - Можно отделить инфраструктуру от прикладного кода
 - Можно разделять код по бизнес функциям

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
 - Оказывается, есть что-то кроме components
 - Можно отделить инфраструктуру от прикладного кода
 - Можно разделять код по бизнес функциям
 - Можно ограничивать связи

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
 - Оказывается, есть что-то кроме components
 - Можно отделить инфраструктуру от прикладного кода
 - Можно разделять код по бизнес функциям
 - Можно ограничивать связи
 - Можно скрывать подробности реализации

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
- Но что, если ты знаешь базу Архитектуры?

Почему я разочаровался

- FSD - набор **достаточно хороших** решений
- Самый мощный эффект – когда ты ноль в Архитектуре
- Но что, если ты знаешь базу Архитектуры?
- Становятся очевидны некоторые проблемы

История одного проекта

История одного проекта

- Проект: команда 5 чел 3+ года разработки

История одного проекта

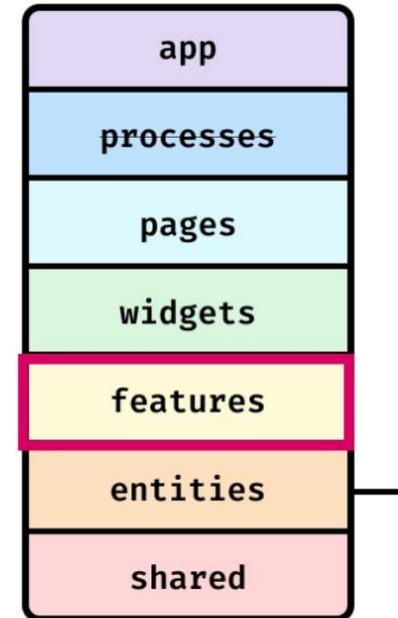
- Проект: команда 5 чел 3+ года разработки
- Первый раз, когда FSD внедрял не я

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
 - Много маленьких **фич** (100+)



Layers

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
 - Много маленьких **фич** (100+)
 - Частые изменения в shared вызывали баги

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
 - Много маленьких **фич** (100+)
 - Частые изменения в shared вызывали баги
 - Связанная функциональность размазана по слоям

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
 - Много маленьких **фич** (100+)
 - Частые изменения в shared вызывали баги
 - Связанная функциональность размазана по слоям
 - Сложно было понять источник данных

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
- В результате я решил провести исследование

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
- В результате я решил провести исследование
 - Это системные проблемы FSD?

История одного проекта

- Проект: Команда 5 чел 3+ года разработки
- Первый раз когда FSD внедрял не я
- Встретил следующие проблемы
- В результате я решил провести исследование
 - Это системные проблемы FSD?
 - И можно ли их исправить?

3 главные проблемы FSD

На что я буду опираться

На что я буду опираться

- Clean Architecture

На что я буду опираться

- Clean Architecture
 - Тоже Архитектурный паттерн

На что я буду опираться

- Clean Architecture
 - Тоже Архитектурный паттерн
 - Но есть книга, из которой можно узнать “Почему”

На что я буду опираться

- Clean Architecture
- GRASP

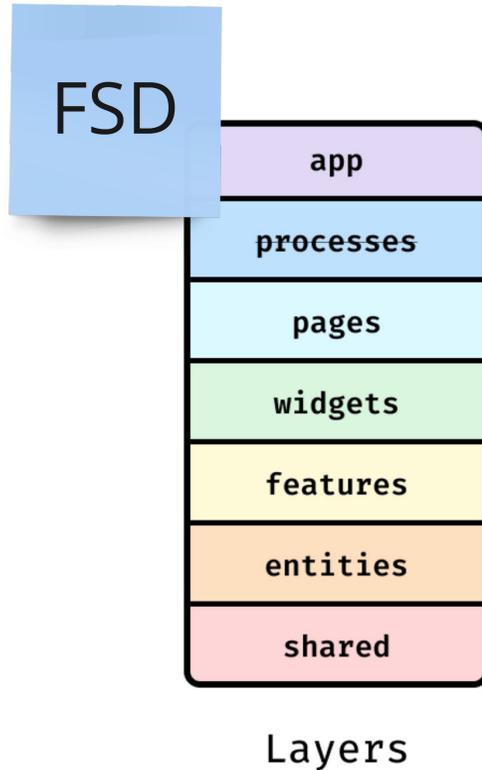
На что я буду опираться

- Clean Architecture
- GRASP
- Личный опыт

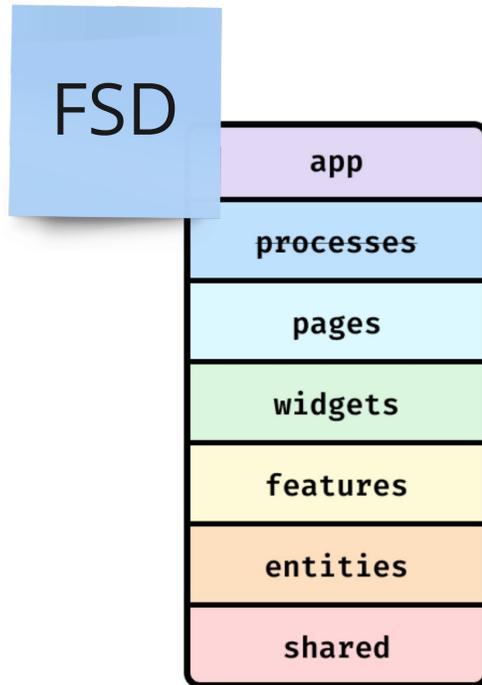
1. Слишком изменчивый **Shared**

Сравним FSD и Clean Architecture

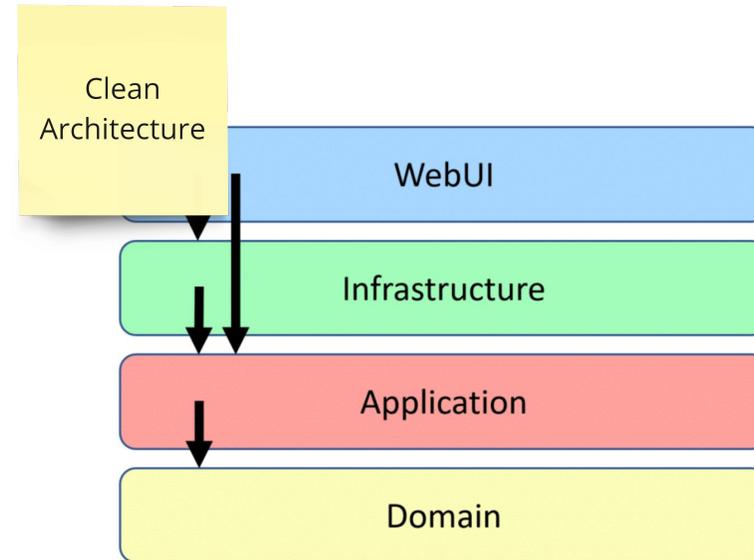
Сравним FSD и Clean Architecture



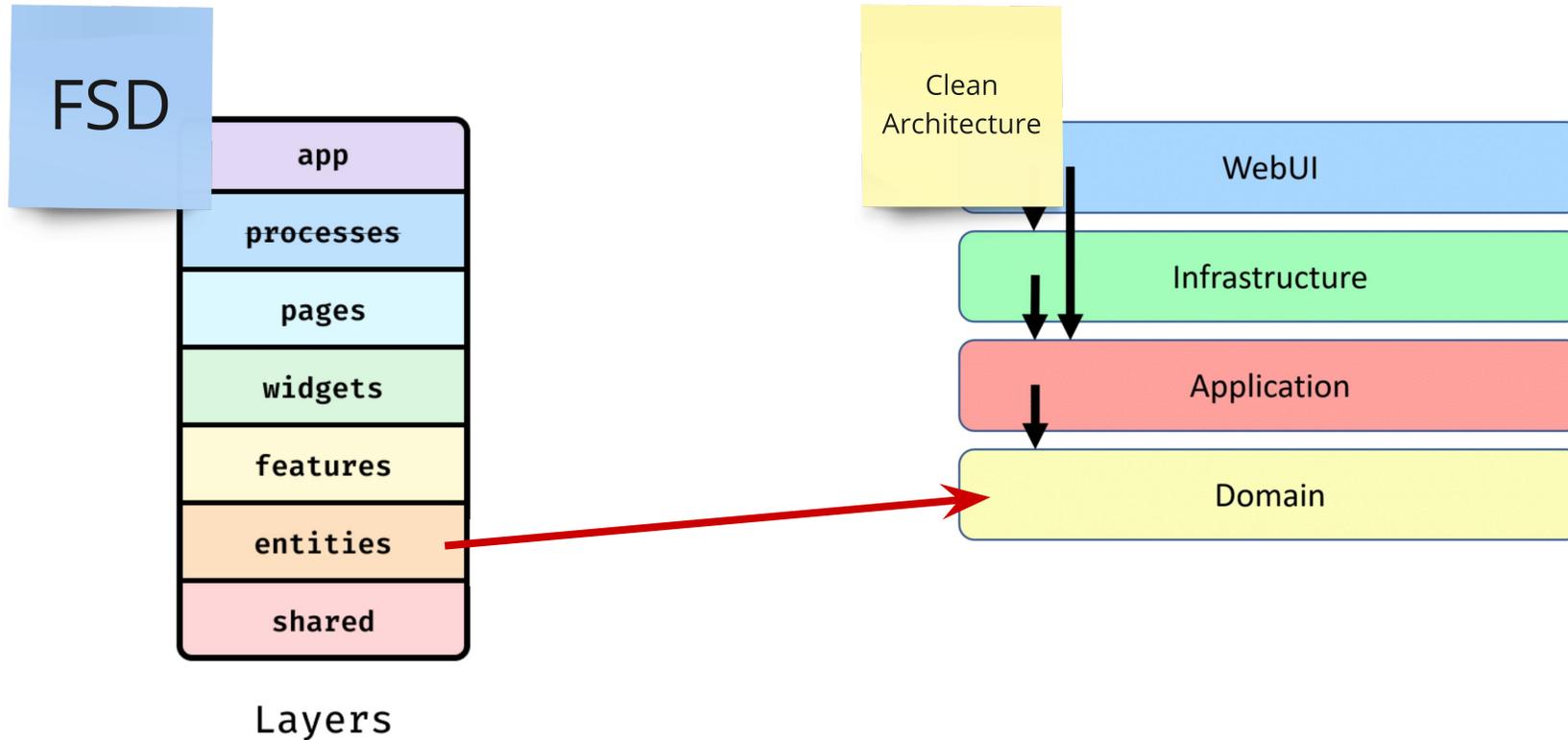
Сравним FSD и Clean Architecture



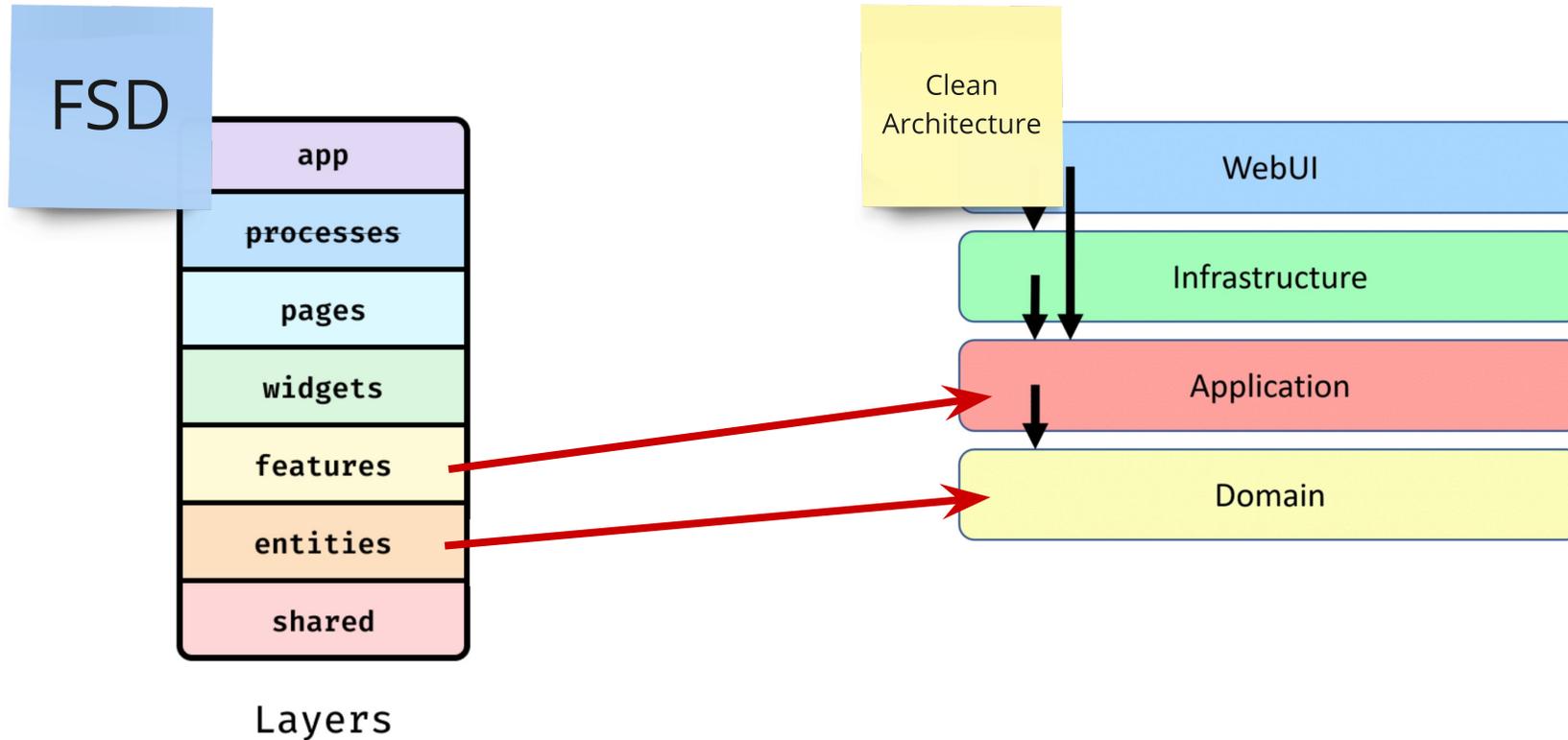
Layers



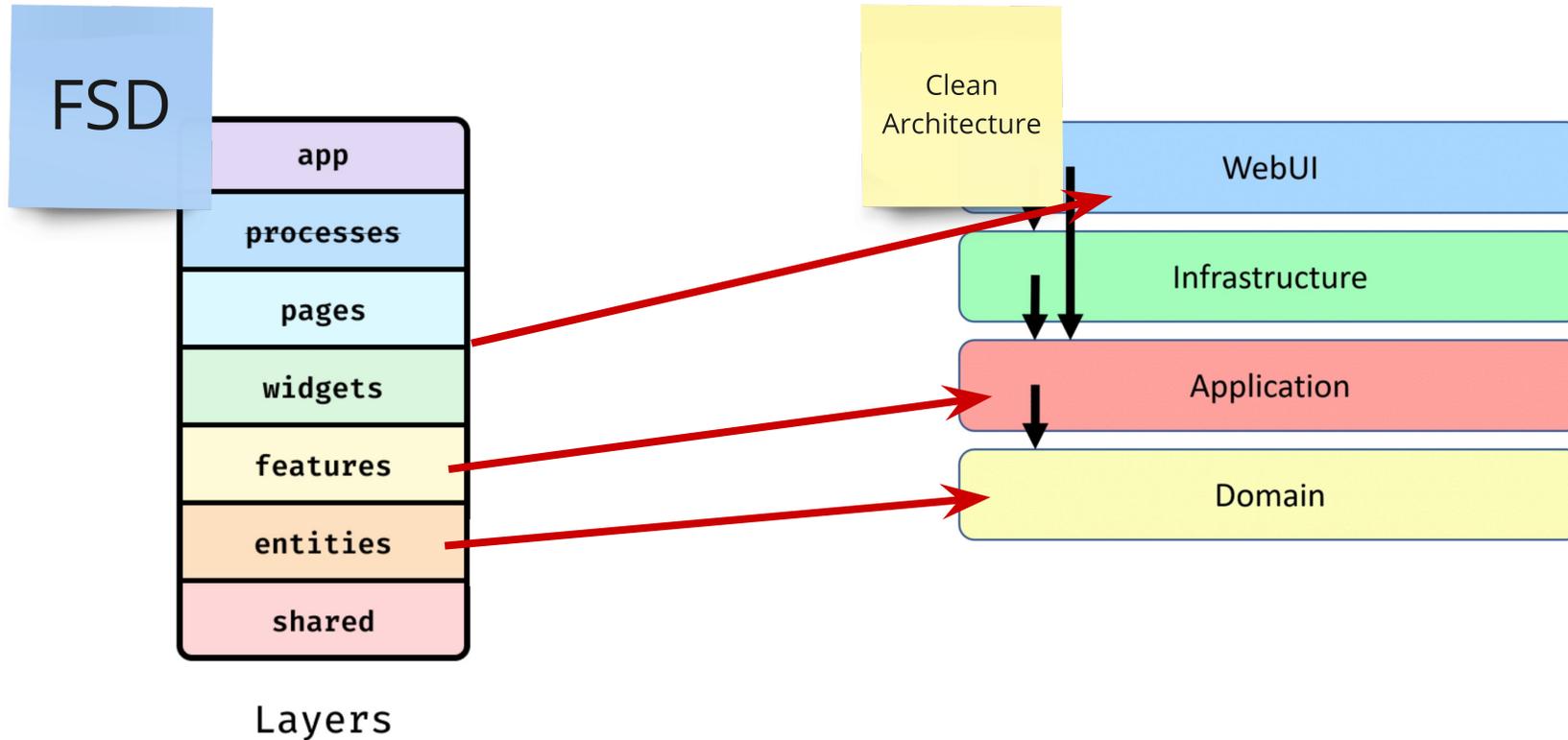
Сравним FSD и Clean Architecture



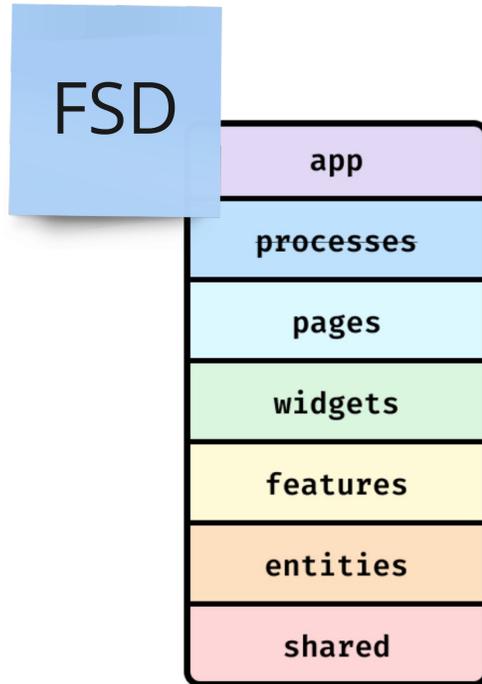
Сравним FSD и Clean Architecture



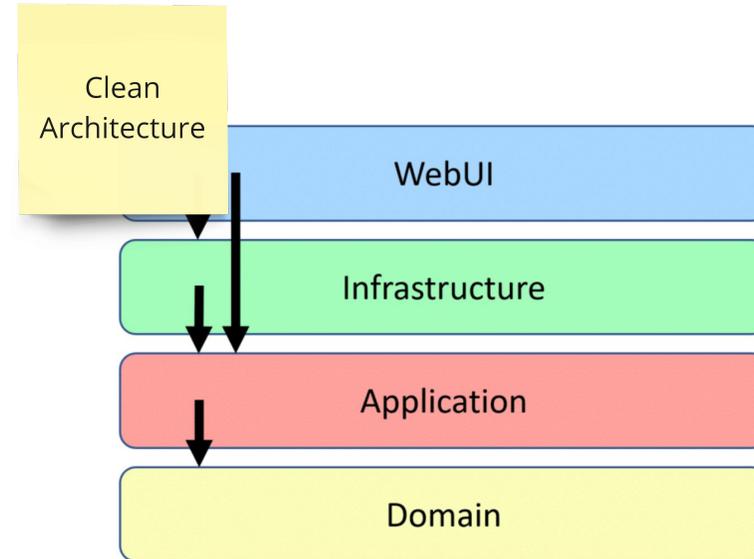
Сравним FSD и Clean Architecture



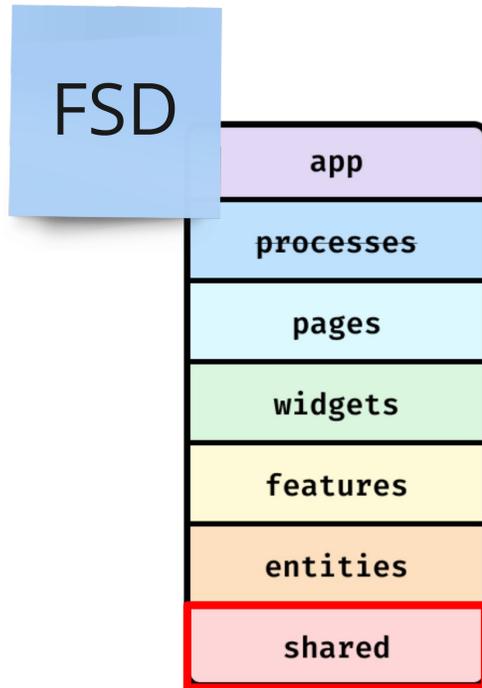
Важное отличие



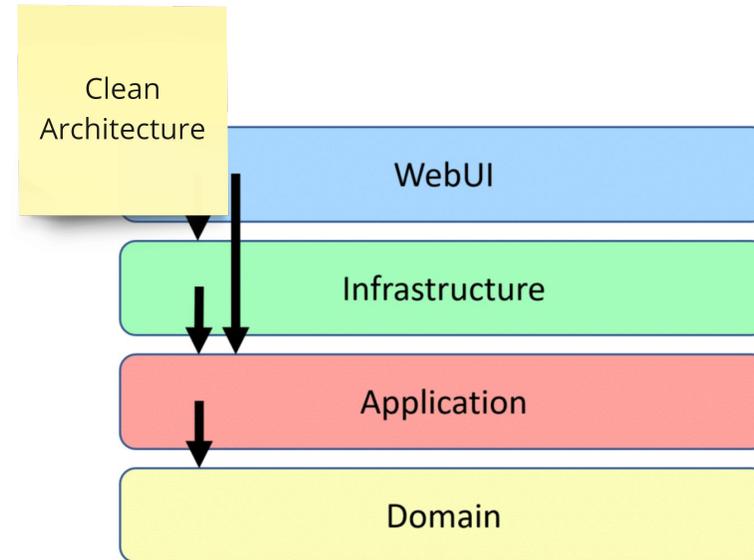
Layers



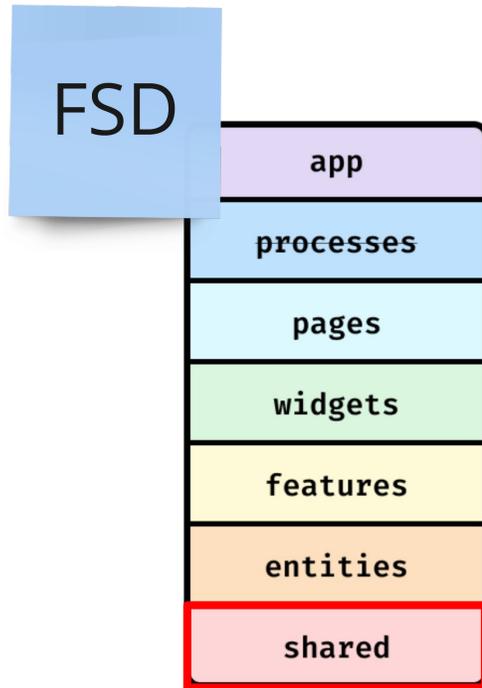
Важное отличие



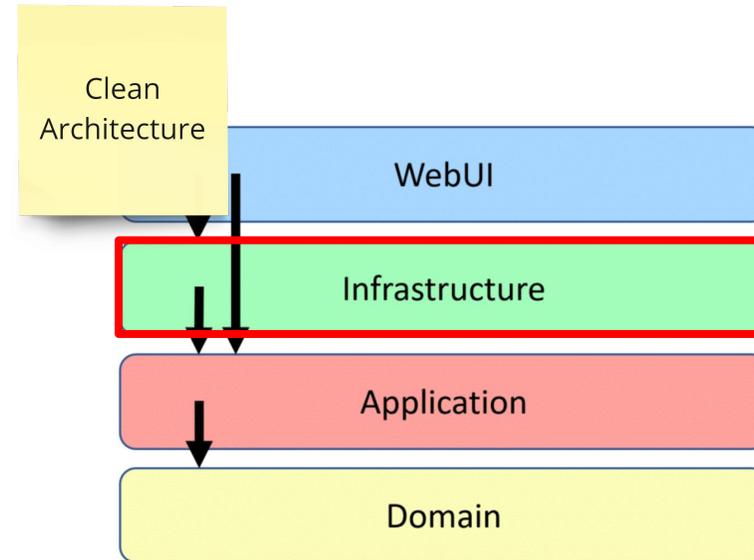
Layers



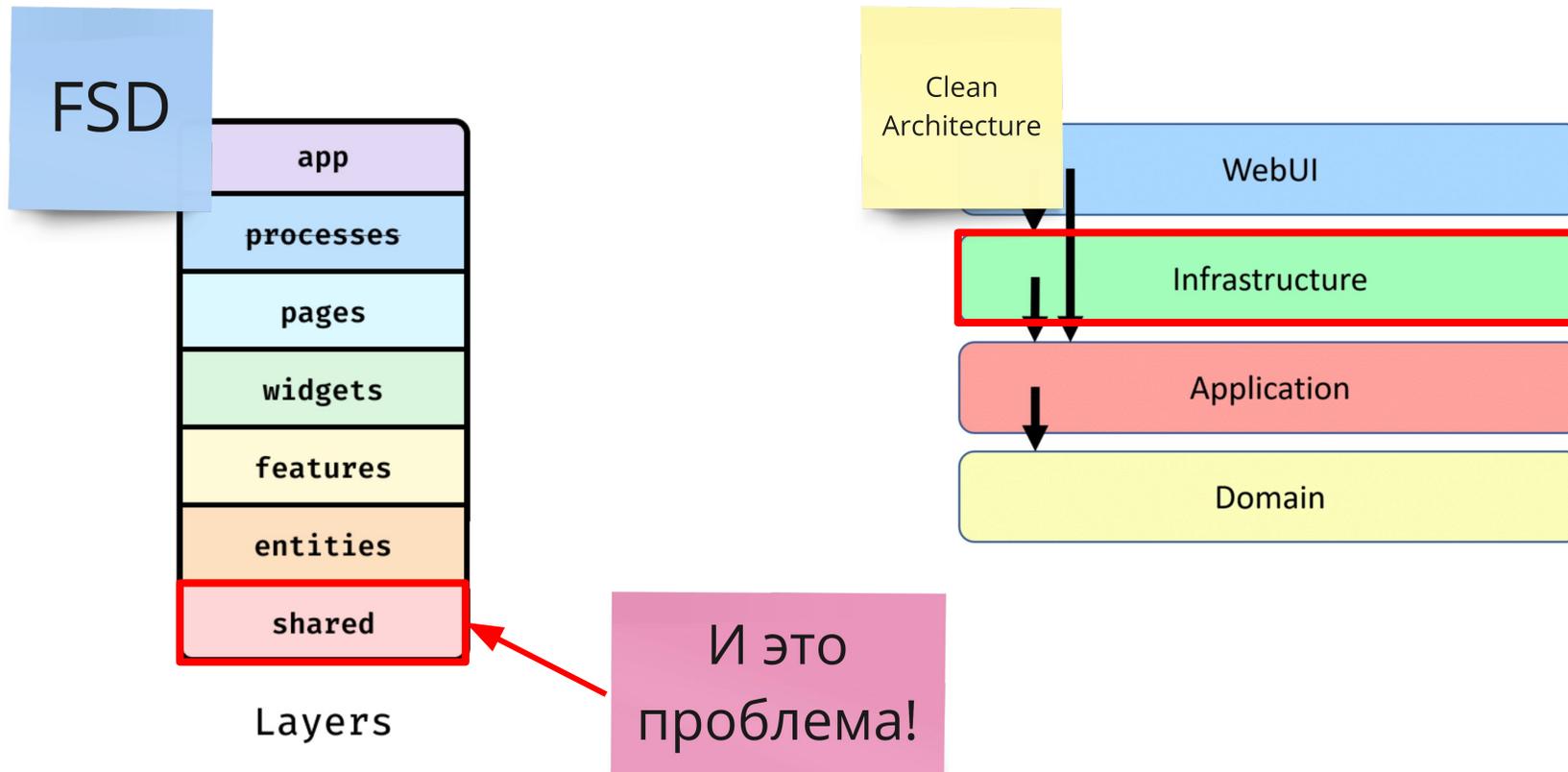
Важное отличие



Layers



Важное отличие



Stable Abstractions Principle (SAP)

Принцип устойчивых абстракций

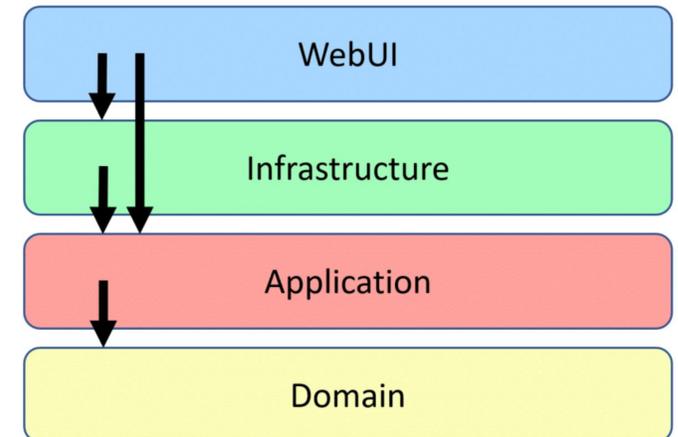
- Зависимости должны быть направлены в сторону увеличения

Абстрактности

Принцип устойчивых абстракций

- Зависимости должны быть направлены в сторону увеличения

Абстрактности

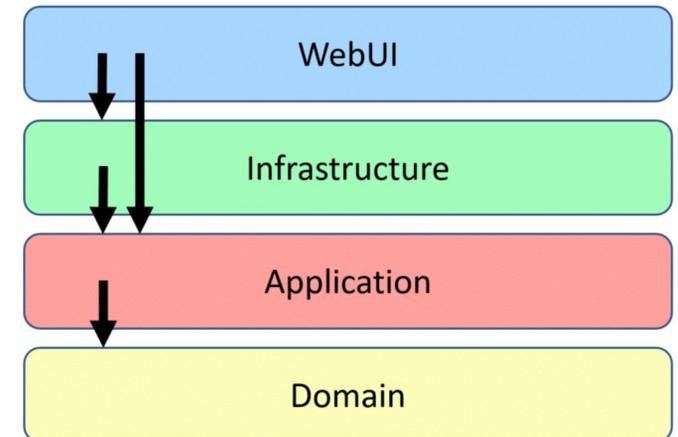


Принцип устойчивых абстракций

- Зависимости должны быть направлены в сторону увеличения

Абстрактности

- Это главная идея Clean Architecture

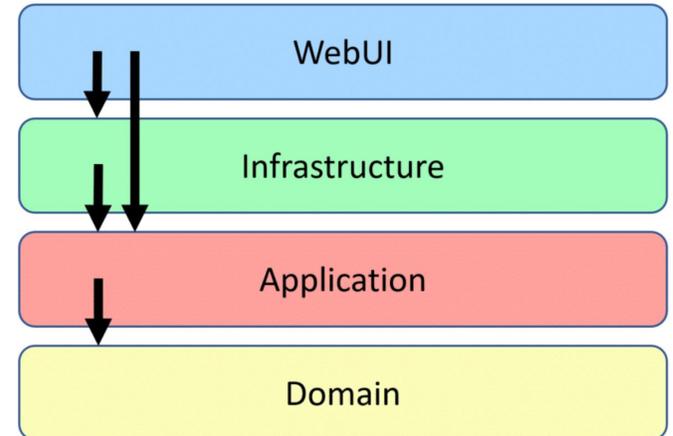


Принцип устойчивых абстракций

- Зависимости должны быть направлены в сторону увеличения

Абстрактности

- Это главная идея Clean Architecture
- Которую FSD не взял



Shared - это боль!

Зона боли

- Это термин из Clean Architecture



Layers

Зона боли

- Это термин из Clean Architecture
- Все зависимости идут в **shared**



Layers

Зона боли

- Это термин из Clean Architecture
- Все зависимости идут в **shared**
- shared очень конкретный
 - uikit
 - api
 - config



Layers

Зона боли

- Это термин из Clean Architecture
- Все зависимости идут в **shared**
- shared очень конкретный
- Часто меняется



Layers

Зона боли

- Это термин из Clean Architecture
- Все зависимости идут в **shared**
- shared очень конкретный
- Часто меняется
- **shared в зоне боли**



Layers

Зона боли

- Это термин из Clean Architecture
- Все зависимости идут в **shared**
- shared очень конкретный
- Часто меняется
- **shared в зоне боли**
- Надежность 🥲



Layers

СИМПТОМЫ, ЧТО ВЫ В ЗОНЕ БОЛИ

СИМПТОМЫ, ЧТО ВЫ В ЗОНЕ БОЛИ

- ButtonV1, ButtonV2, ButtonV3

СИМПТОМЫ, ЧТО ВЫ В ЗОНЕ БОЛИ

- ButtonV1, ButtonV2, ButtonV3
- Нужен глобальный рефакторинг

СИМПТОМЫ, ЧТО ВЫ В ЗОНЕ БОЛИ

- ButtonV1, ButtonV2, ButtonV3
- Нужен глобальный рефакторинг
- Нужно больше фасадов

И как это решить?

И как это решить

- Dependency Inversion

И как это решить

- Dependency Inversion
- DI во Front-end не прижился

И как это решить

- Dependency Inversion
- DI во Front-end не прижился
- Без DI никак, можно только облегчить...

3 проблемы FSD

1. Слишком изменчивый **shared**

2. **Entities** не позволяют
описать модель бизнеса

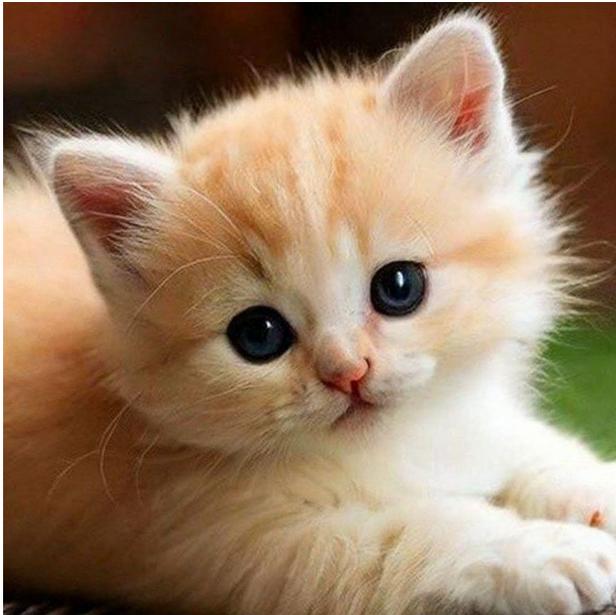
Что такое модель?

Что такое модель?

- Упрощенное представление реального объекта

Что такое модель?

- Упрощенное представление реального объекта



```
1  type Cat = {  
2      color: string;  
3      age: number;  
4  }  
5  
6  const shouldMimimi = (cat: Cat): boolean => {  
7      return cat.color === 'white' && cat.age < 0.25  
8  }
```

Что такое модель?

- Упрощенное представление реального объекта
- Все архитектурные шаблоны отделяют “модель” бизнеса

Что такое модель?

- Упрощенное представление реального объекта
- Все архитектурные шаблоны отделяют “модель” бизнеса
- Чаще всего – это самая важная информация

Что такое модель?

- Упрощенное представление реального объекта
- Все архитектурные шаблоны отделяют “модель” бизнеса
- Чаще всего – это самая важная информация
- Выделение модели помогает:

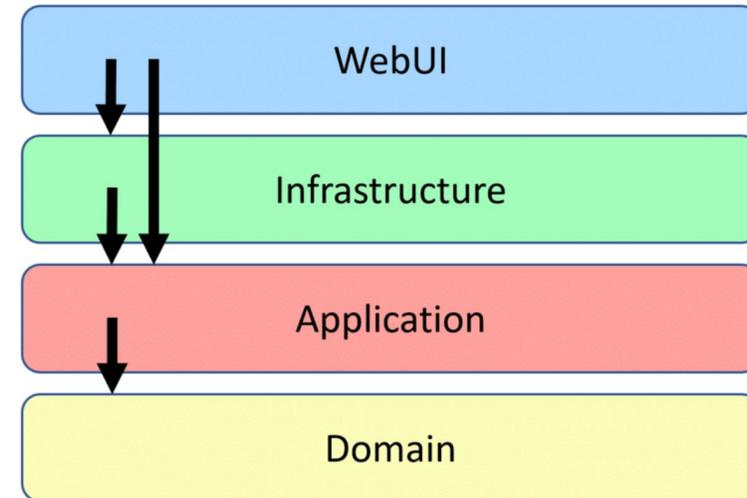
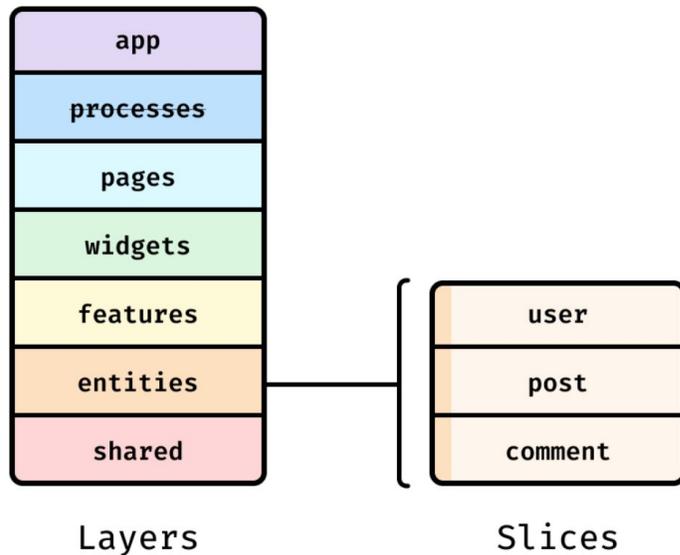
Что такое модель?

- Упрощенное представление реального объекта
- Все архитектурные шаблоны отделяют “модель” бизнеса
- Чаще всего – это самая важная информация
- Выделение модели помогает:
 - Понятнее

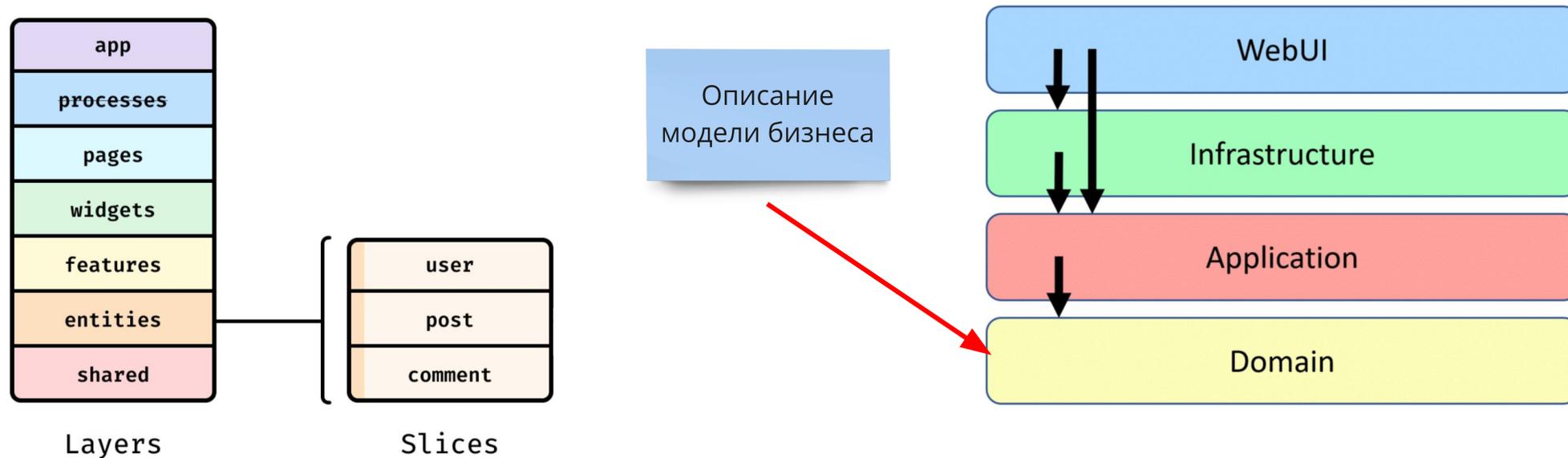
Что такое модель?

- Упрощенное представление реального объекта
- Все архитектурные шаблоны отделяют “модель” бизнеса
- Чаще всего – это самая важная информация
- Выделение модели помогает:
 - Понятнее
 - Надёжнее

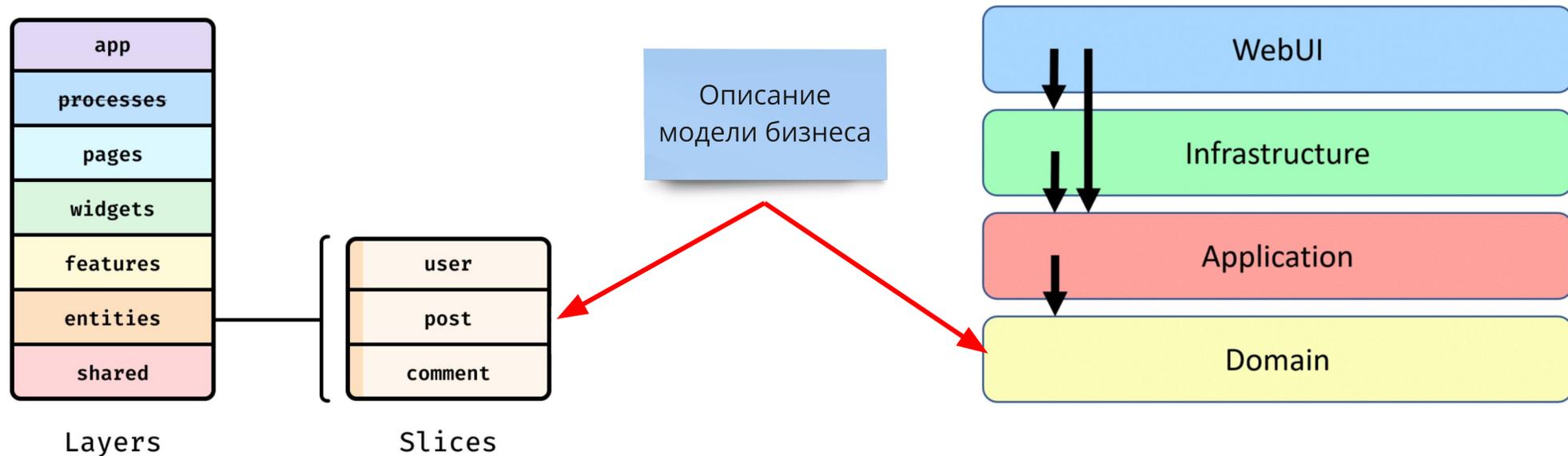
Продолжаем сравнение с Clean Architecture



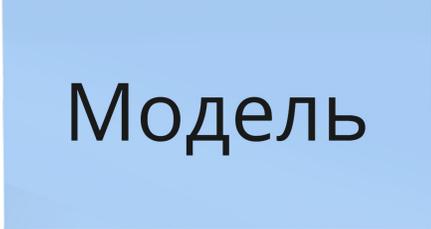
Продолжаем сравнение с Clean Architecture



Продолжаем сравнение с Clean Architecture

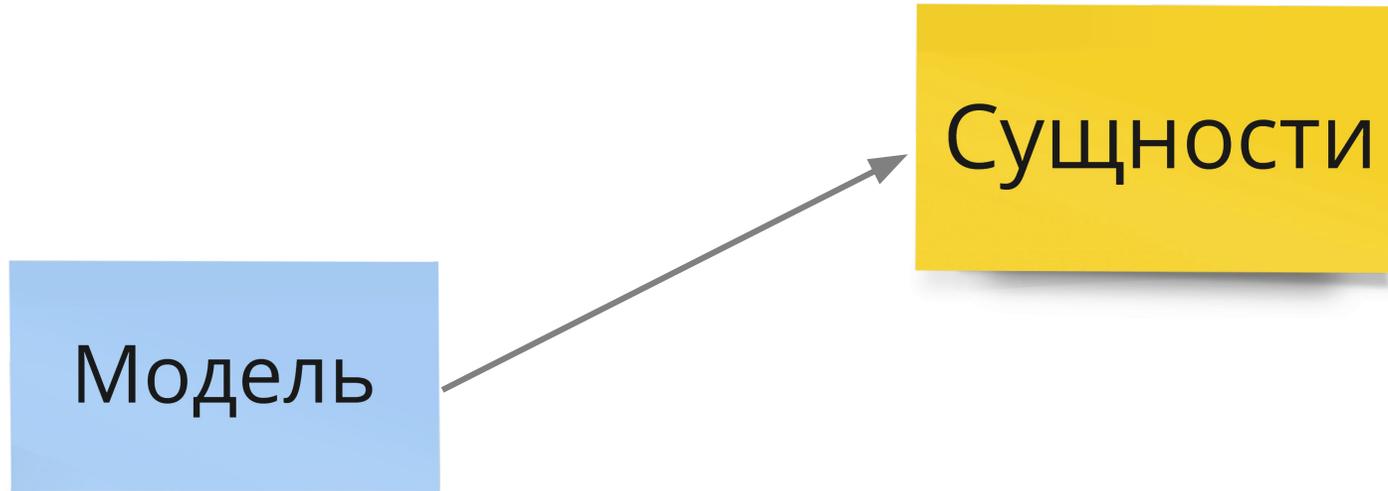


Но есть проблема

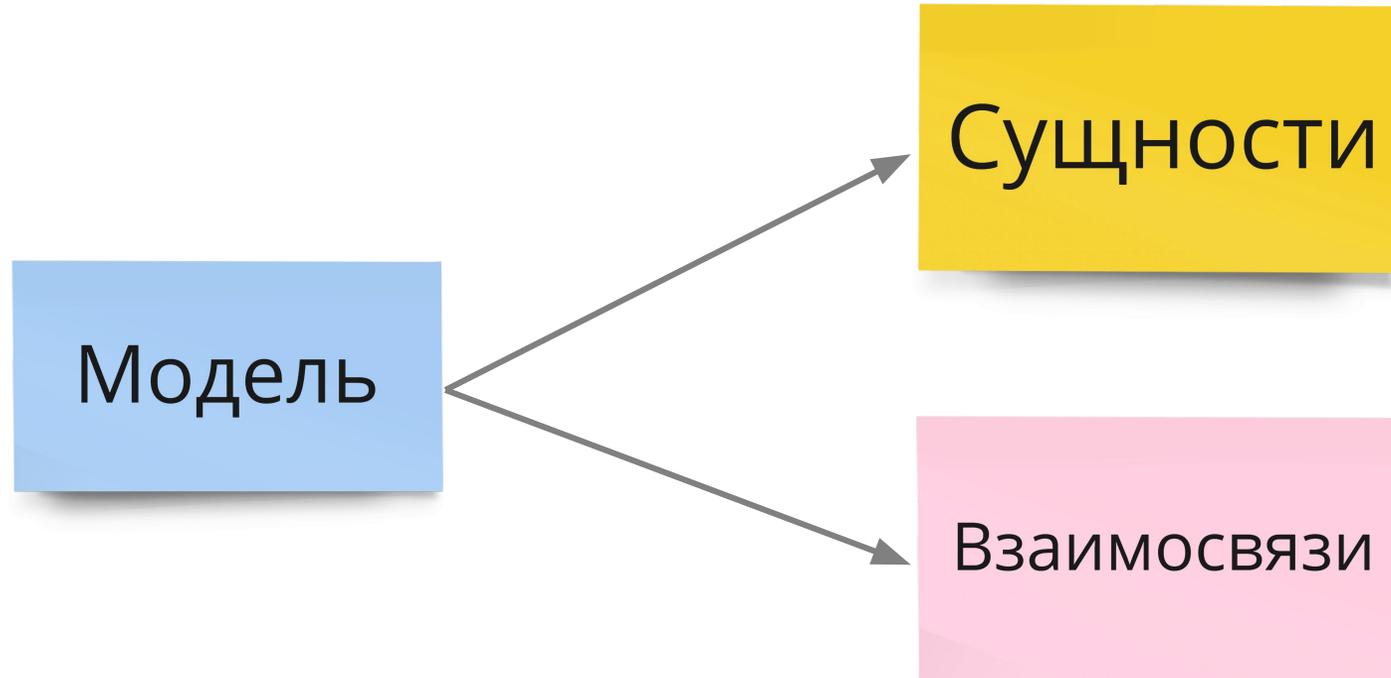


Модель

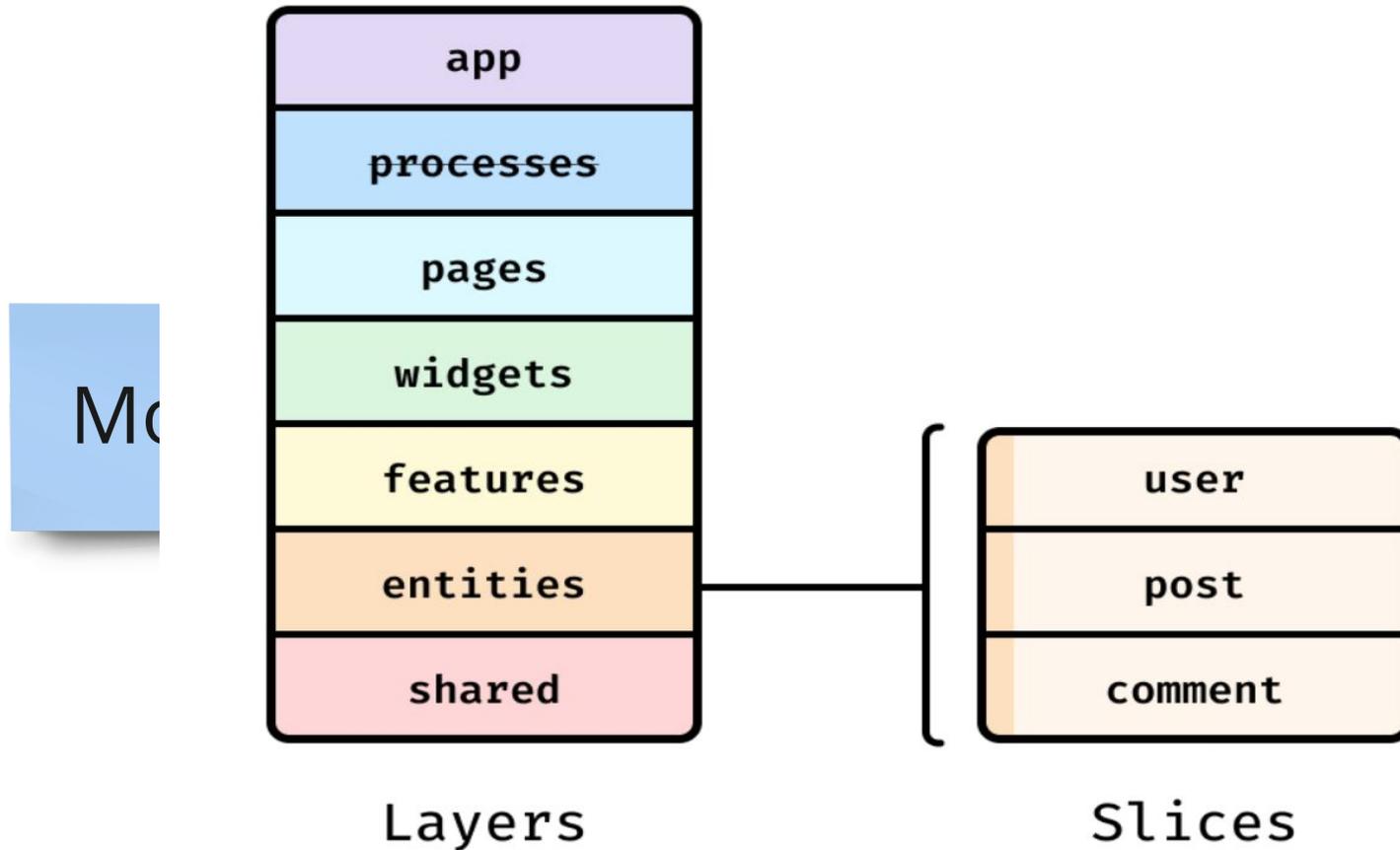
Но есть проблема



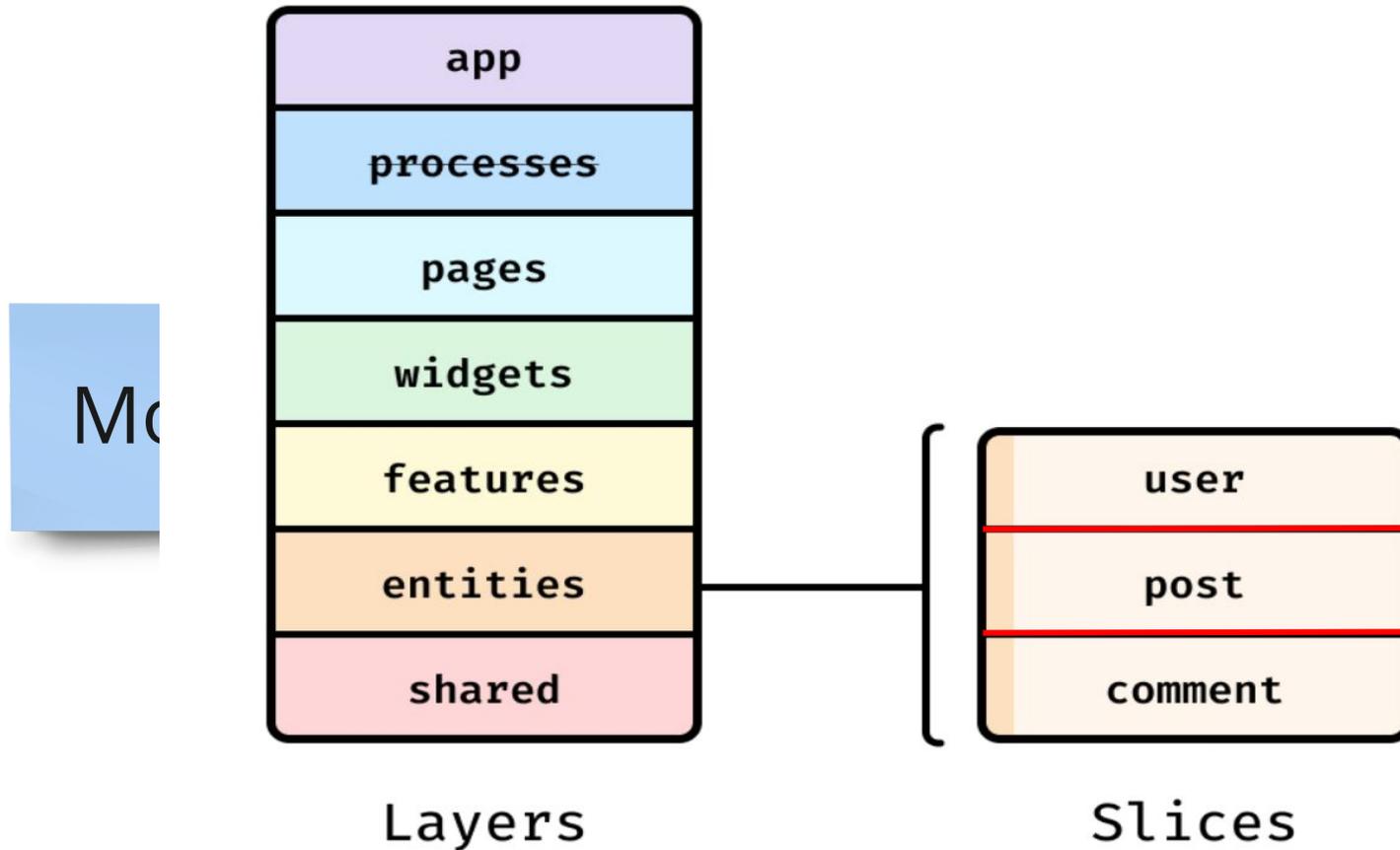
Но есть проблема



Но есть проблема



Но есть проблема



К чему это приводит?

К чему это приводит?

- У людей когнитивный диссонанс

К чему это приводит?

- У людей когнитивный диссонанс
- Entities - описание бизнес сущностей

К чему это приводит?

- У людей когнитивный диссонанс
- Entities - описание бизнес сущностей
- Но бизнес сущности **всегда связаны**

К чему это приводит?

- У людей когнитивный диссонанс
- Entities - описание бизнес сущностей
- Но бизнес сущности **всегда связаны**
- FSD вынуждает нас провести границу **между плотно связанными модулями**

Как FSD предлагает это решать?

Как FSD предлагает это решать?

- Связи между сущностями по id

Как FSD предлагает это решать?

- Связи между сущностями по id

```
1  type User = {  
2      id: string;  
3  }
```

```
1  type Post = {  
2      id: string  
3      userId: string  
4  }
```

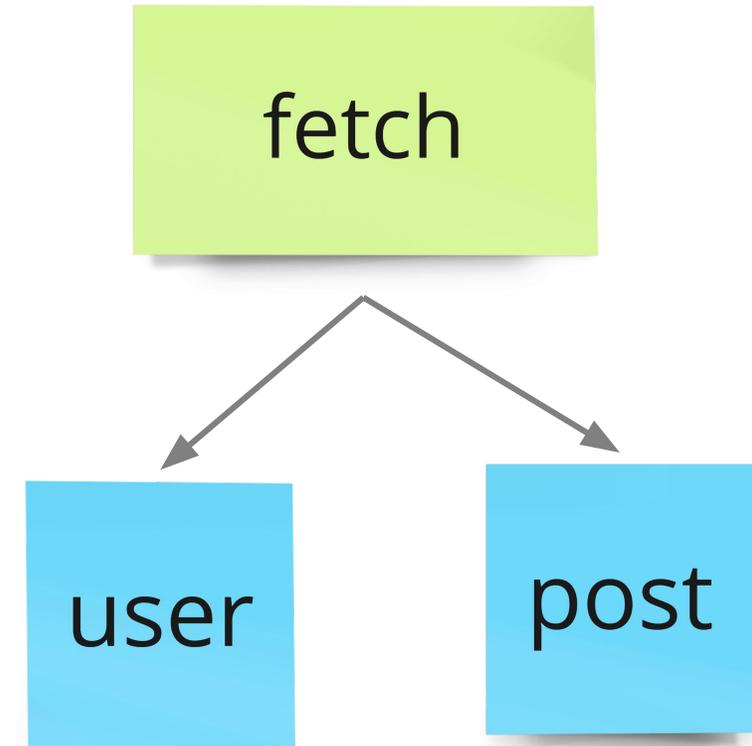
Как FSD предлагает это решать?

- Связи между сущностями по id
- Но что делать, если с бека приходит вот это?

```
1  type Post = {  
2      id: string  
3      user: User  
4  }  
5  
6  type User = {  
7      id: string;  
8  }
```

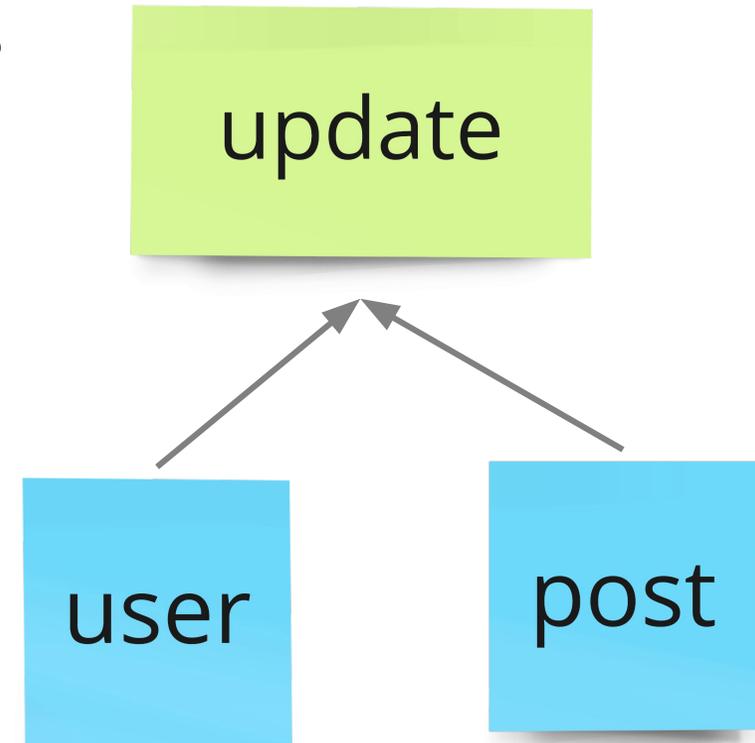
Как FSD предлагает это решать?

- Связи между сущностями по id
- Но что делать если с бека приходит вот это?
- Извольте сделать нормализацию



Как FSD предлагает это решать?

- Связи между сущностями по id
- Но что делать если с бека приходит вот это?
- Извольте сделать нормализацию



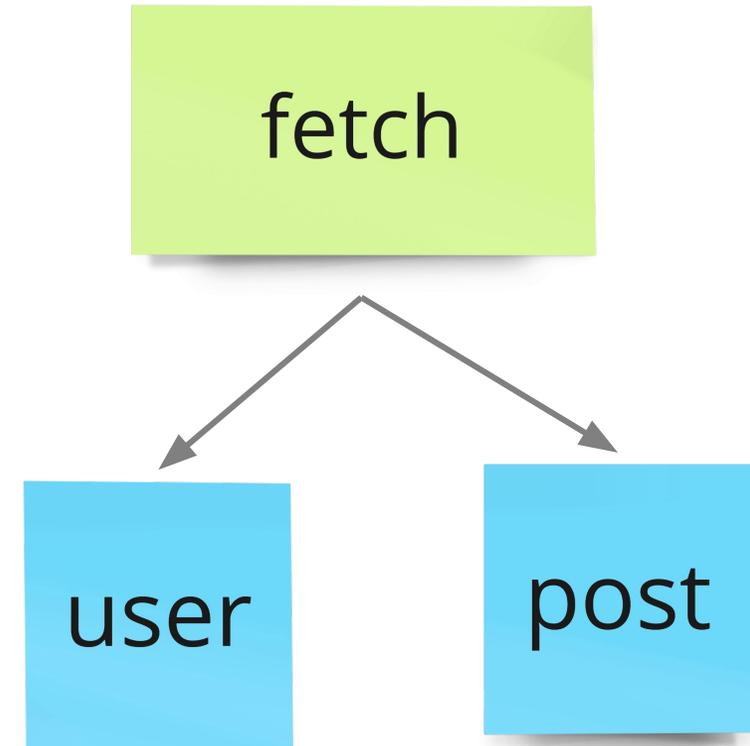
Это приводит к большой проблеме:

Это приводит к большой проблеме:

- Важнейшие связи между бизнес сущностями становятся **неявными**

Это приводит к большой проблеме:

- Важнейшие связи между бизнес сущностями становятся **неявными**
- Это просто много кода →



А может просто разрешить связи
между сущностями?

А может просто разрешить связи между сущностями?

- В Entities есть ui

А может просто разрешить связи между сущностями?

- В Entities есть ui
- Появляется много “циклов”

А может просто разрешить связи между сущностями?

- В Entities есть ui
- Появляется много “циклов”
- Происходит “сваливание” всего кода в entities

3 Проблемы FSD

1. Слишком изменчивый **shared**
2. **Entities** не позволяют описать бизнес модель

3. Widgets + Features + Entities =
high coupling

Я начну с масла



ToDoList

Я начну с масла



ToDoList

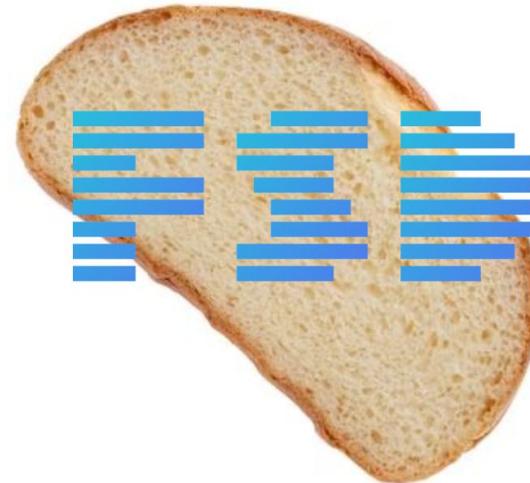


DRY

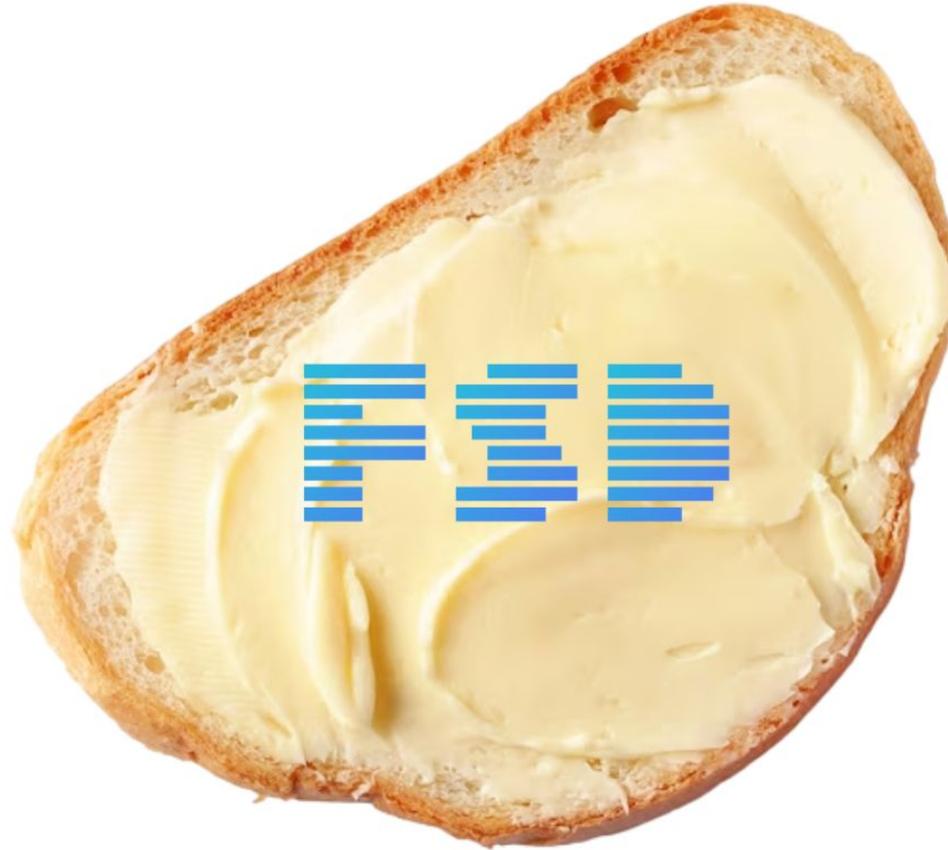
Я начну с масла



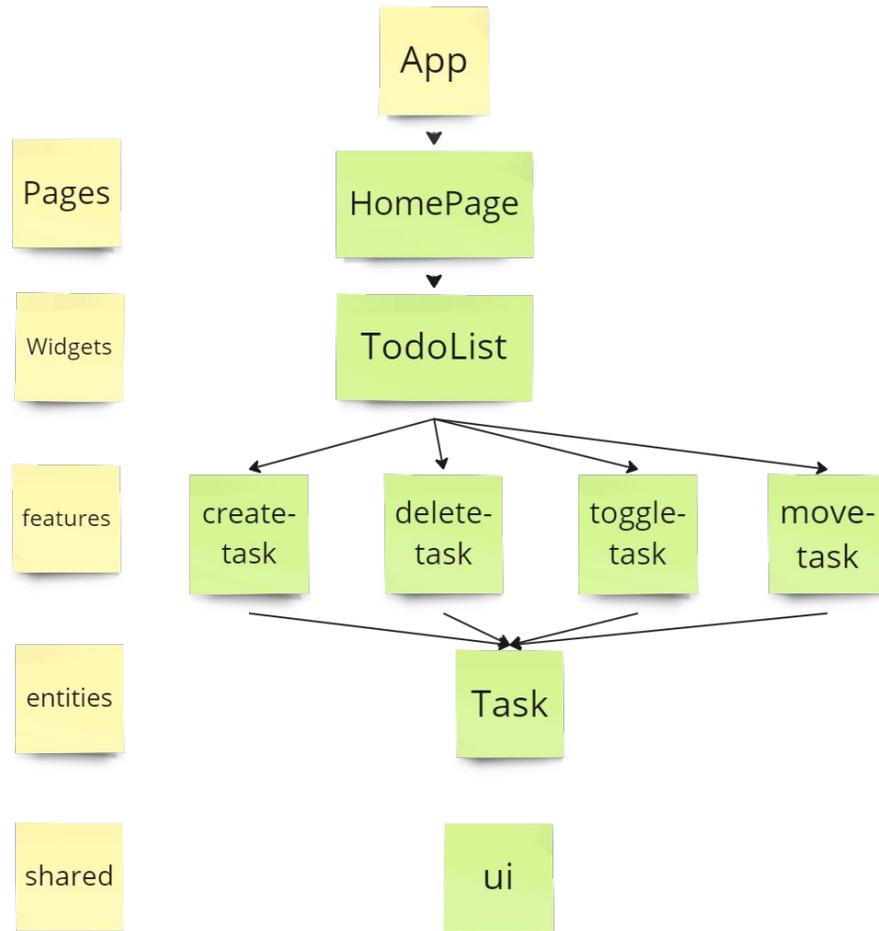
ToDoList



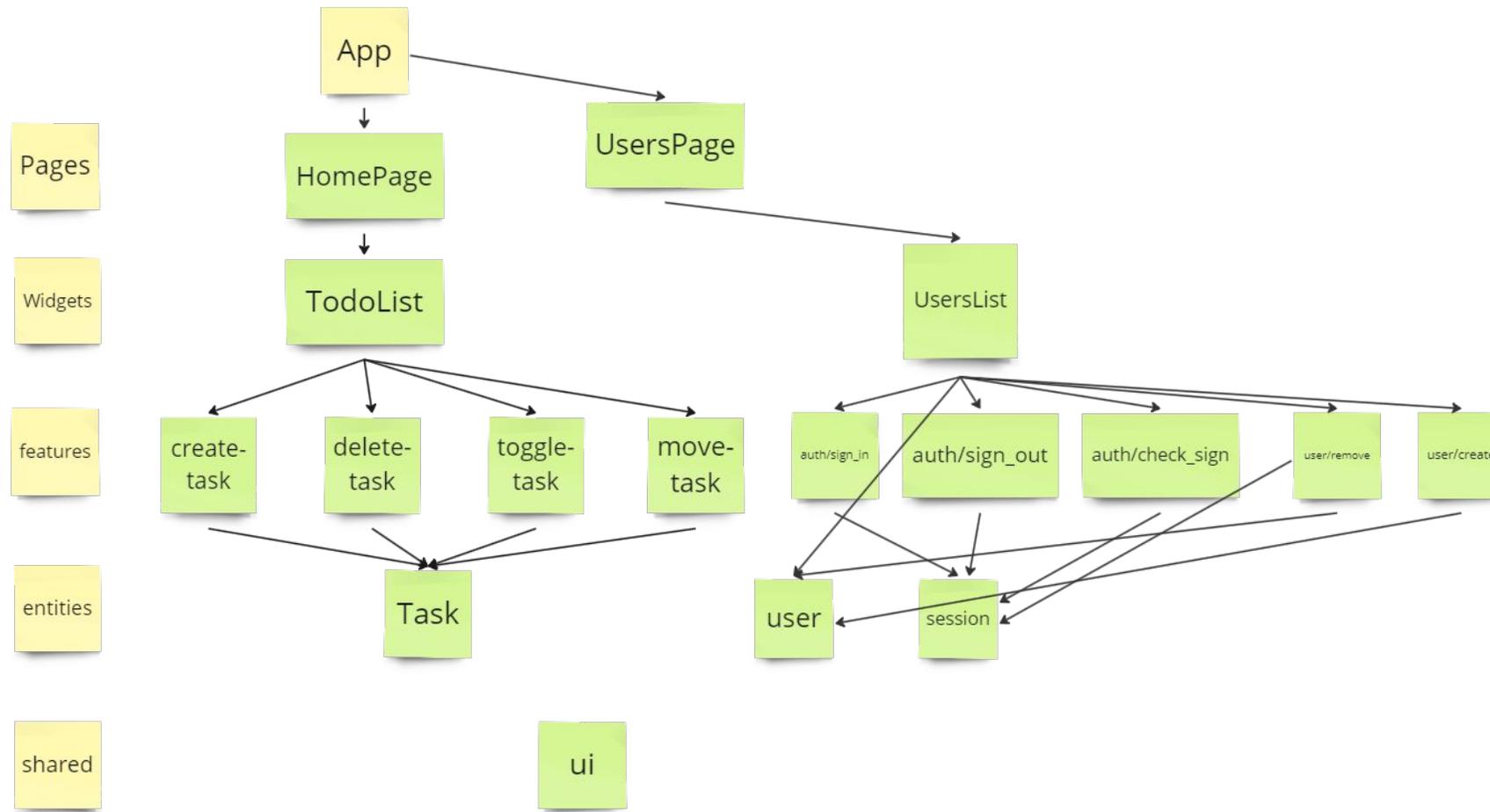
Я начну с масла



Пример декомпозиции по FSD



Пример декомпозиции по FSD





Ты должен был уменьшить сложность

А не увеличить её

Low coupling, High cohesion

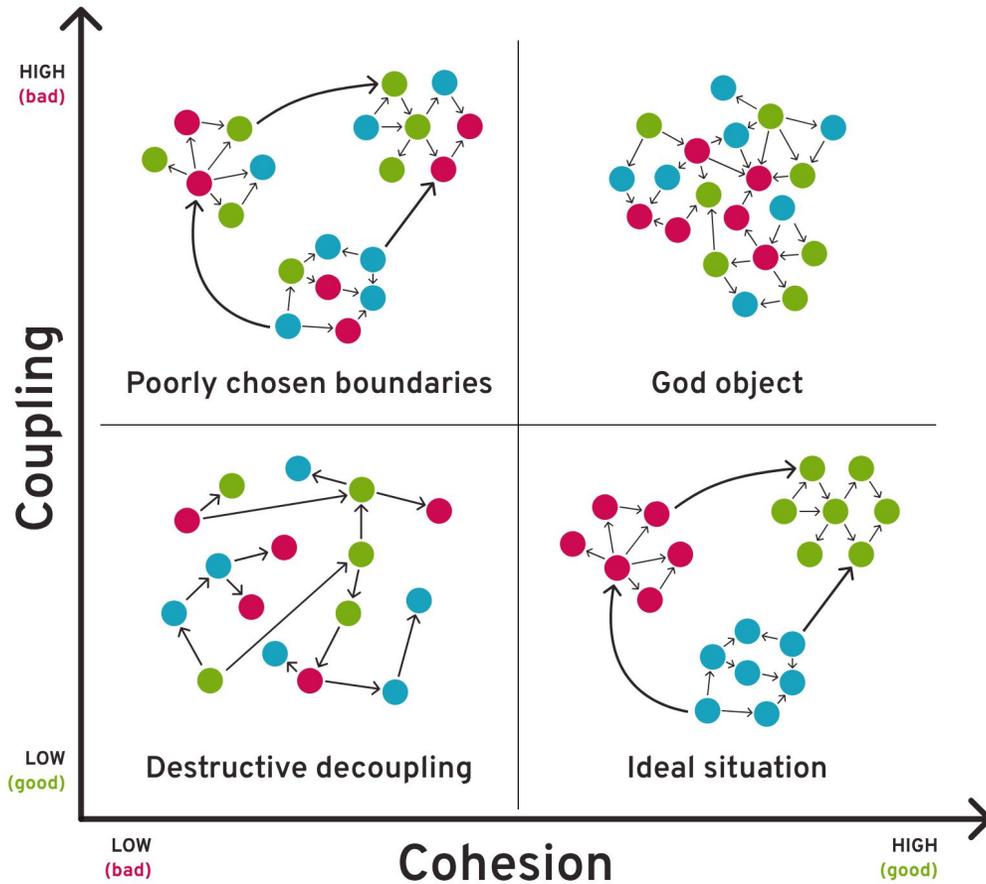
Low coupling, High cohesion

- Low coupling : Мало связей

Low coupling, High cohesion

- Low coupling : Мало связей
- High cohesion : Высокая смысловая связанность

Low coupling, High cohesion



зая СВЯЗАННОСТЬ

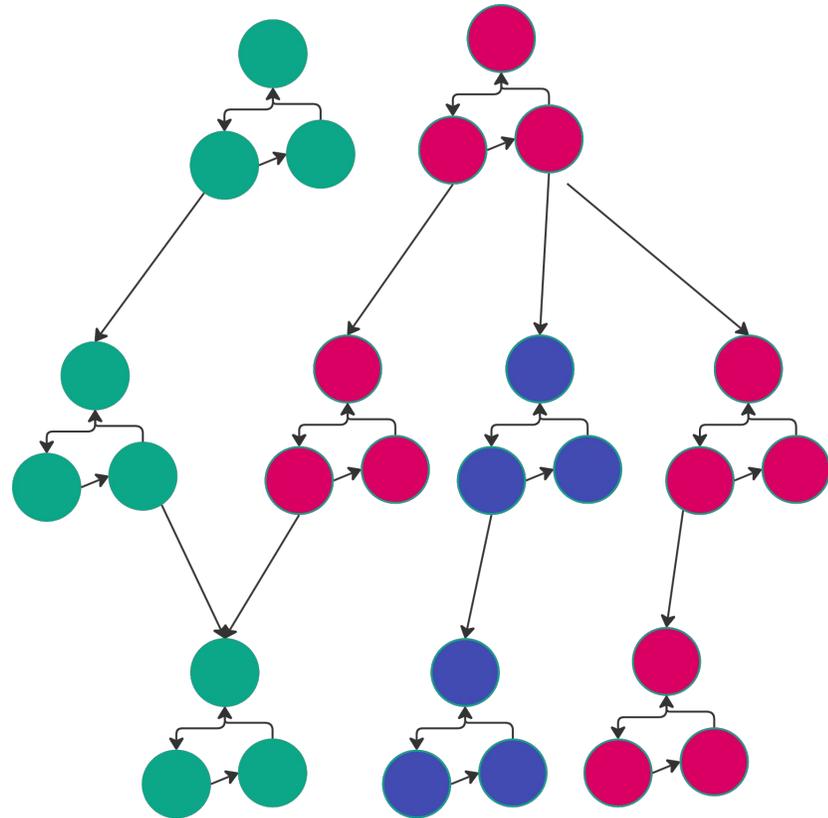
Low coupling, High cohesion

- Low coupling : Мало связей
- High cohesion : Высокая смысловая связанность
- С High cohesion **Всё хорошо**

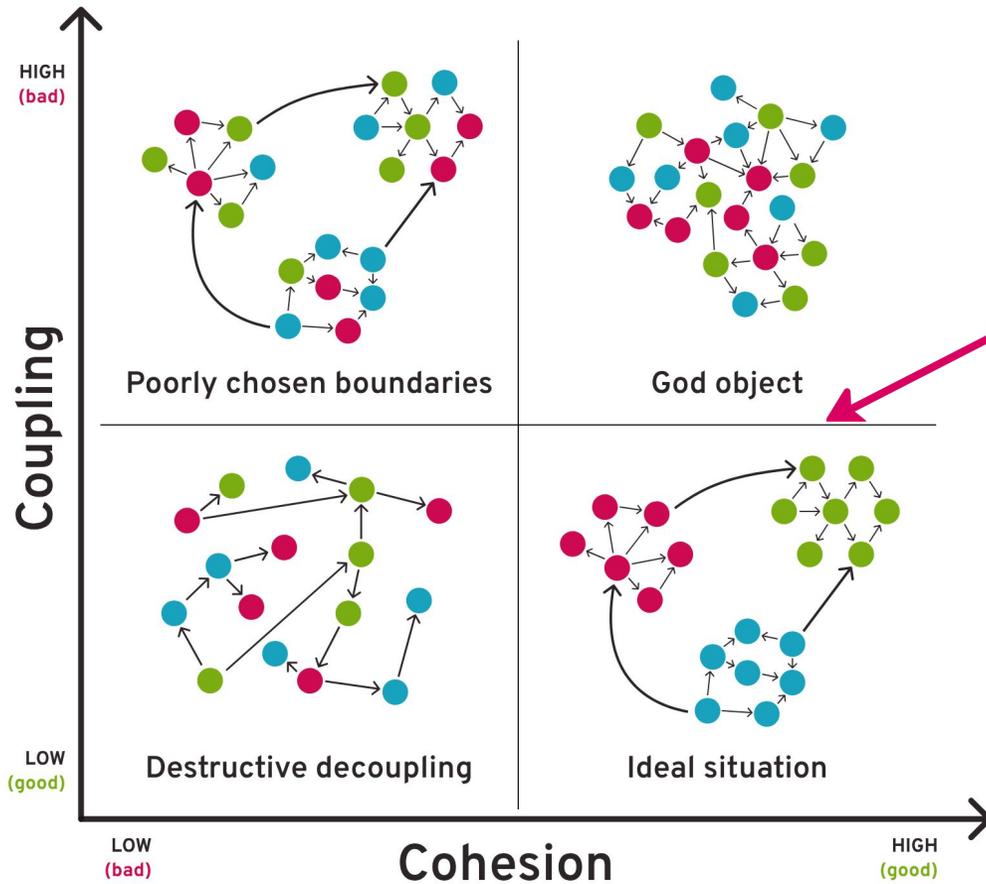
Low coupling, High cohesion

- Low coupling : Мало связей
- High cohesion : Высокая смысловая связанность
- С High cohesion **Всё хорошо**
- С Low coupling Есть проблемы

Low coupling, High cohesion



Low coupling, High cohesion



Вот
здесь!

Почему так получается?

Почему так получается?

- У FSD есть **однозначные** критерии декомпозиции

Почему так получается?

- У FSD есть однозначные критерии декомпозиции
- Entities: Слова из бизнеса

Почему так получается?

- У FSD есть однозначные критерии декомпозиции
- Entities: Слова из бизнеса
- Features: Действия пользователя над сущностями

Почему так получается?

- У FSD есть однозначные критерии декомпозиции
- Entities: Слова из бизнеса
- Features: Действия пользователя над сущностями
- Widgets: Объединение Entities и Features

Почему так получается?

- У FSD есть однозначные критерии декомпозиции
- Entities: Слова из бизнеса
- Features: Действия пользователя над сущностями
- Widgets: Объединение Entities и Features
- Слишком большое количество модулей

Почему так получается?

- У FSD есть однозначные критерии декомпозиции
- Entities: Слова из бизнеса
- Features: Действия пользователя над сущностями
- Widgets: Объединение Entities и Features
- Слишком большое количество модулей
- При декомпозиции мы **не учитывали связи**

Декомпозиция должна
учитывать СВЯЗИ

В этом месте в FSD есть прогресс

В этом месте в FSD есть прогресс

- В доке написано не декомпозировать pages без переиспользования

В этом месте в FSD есть прогресс

- В доке написано не декомпозировать pages без переиспользования
- Я считаю это в корне неверным подходом

В этом месте в FSD есть прогресс

- В доке написано не декомпозировать pages без переиспользования
- Я считаю это в корне неверным подходом
- **Полностью обесценивает смысл слоев**

3 Проблемы FSD

1. Слишком изменчивый **shared**
2. **Entities** не позволяют описать модель
3. Widgets + Features + Entities = **high coupling**

Как это решаю я?

Какие изменения я вношу в FSD

Какие изменения я вношу в FSD

1. Я оставляю слои и их порядок

Какие изменения я вношу в FSD

1. Я оставляю слои и их порядок
2. Меняю смысл слоя features entities

Какие изменения я вношу в FSD

1. Я оставляю слои и их порядок
2. Меняю смысл слоя features entities
3. Добавляю иерархию сегментов внутри слайсов

Какие изменения я вношу в FSD

1. Я оставляю слои и их порядок
2. Меняю смысл слоя features entities
3. Добавляю иерархию сегментов внутри слайсов
4. DI для связи между фичами

Как я меняю смысл слов

Как я меняю смысл слов

- В базовом FSD

Как я меняю смысл слоев

- В базовом FSD
- Entities – бизнес объекты

Как я меняю смысл слов

- В базовом FSD
- Entities – бизнес объекты
- Features – действия пользователя

Как я меняю смысл слоев

- В базовом FSD
- Entities – бизнес объекты
- Features – действия пользователя
- Критерии декомпозиции не учитывают связи

Как я меняю смысл слов

- В моей версии

Как я меняю смысл слов

- В моей версии
- Entities – группа связанных сущностей
 - User: Session, User, Profile
 - Board: Board, Rectangle, Line

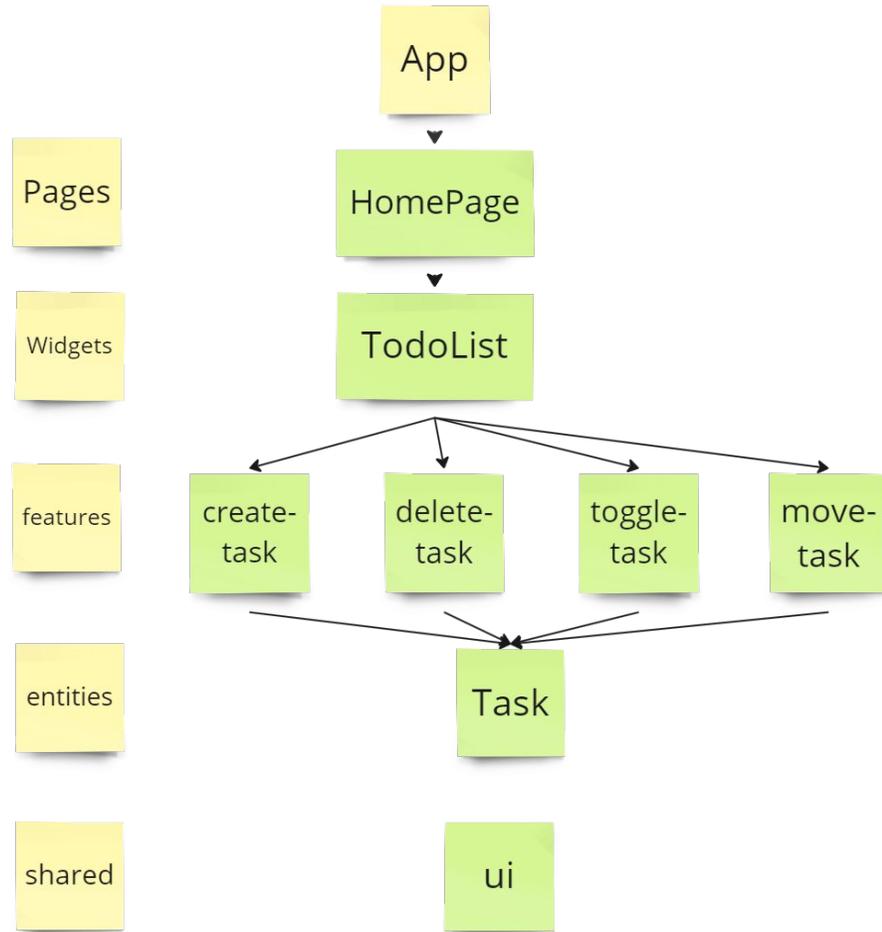
Как я меняю смысл слов

- В моей версии
- Entities – группа связанных сущностей
 - User: Session, User, Profile
 - Board: Board, Rectangle, Line
- Features – независимые блоки функциональности
 - TodoList
 - BoardEditor

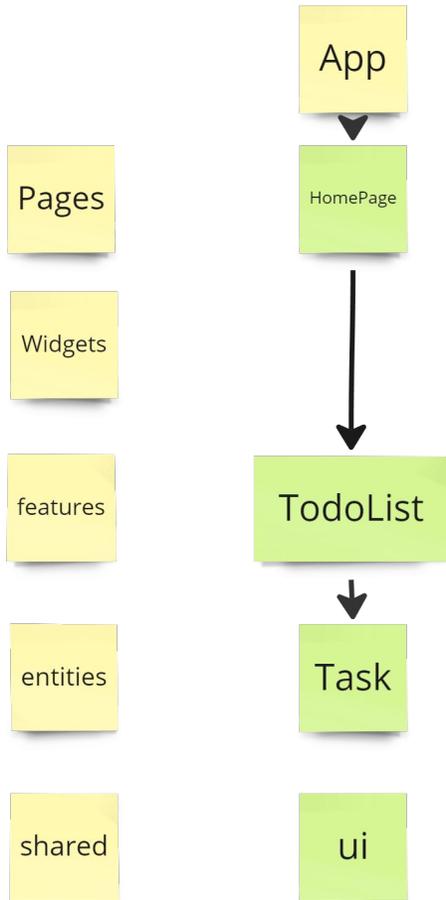
Как я меняю смысл слоев

- В моей версии
- Entities – группа связанных сущностей
 - User: Session, User, Profile
 - Board: Board, Rectangle, Line
- Features – независимые блоки функциональности
 - TodoList
 - BoardEditor
- Главный критерий: Low coupling/High cohesion

Как я меняю смысл слоев



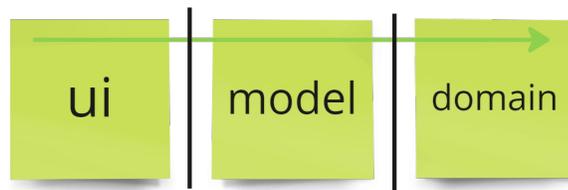
Как я меняю смысл слоев



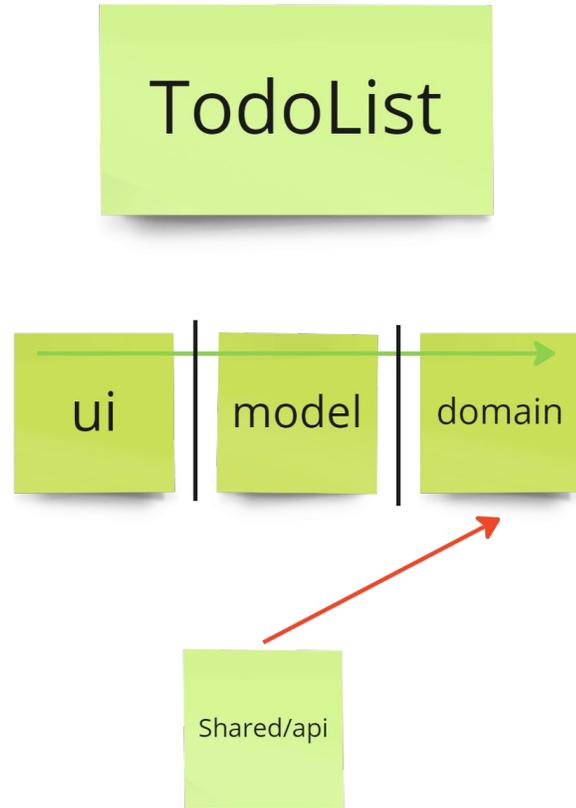
Добавляю иерархию сегментов

Добавляю иерархию сегментов

TodoList



Добавляю иерархию сегментов



Как использую DI

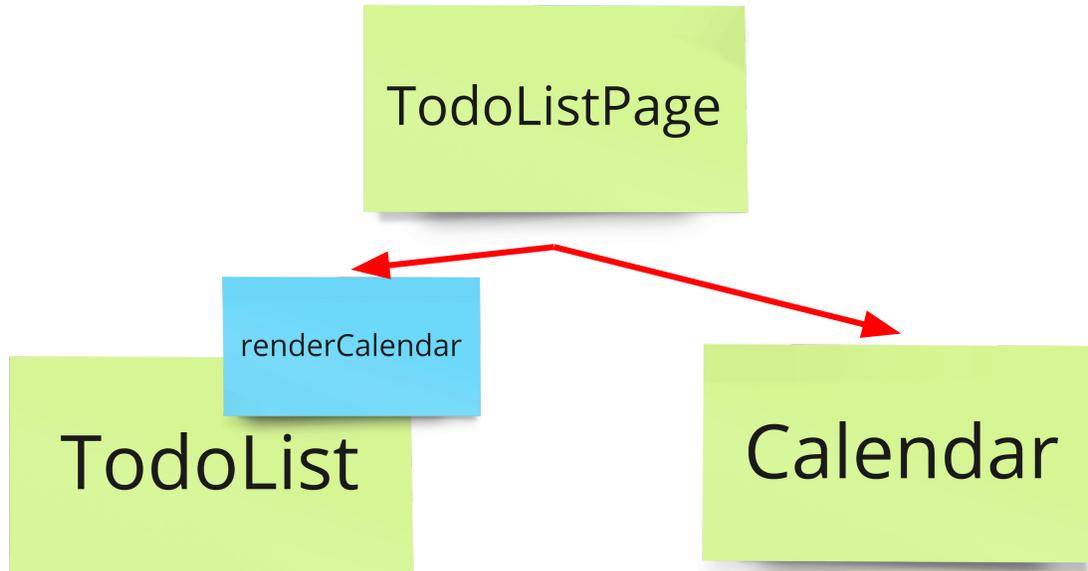
ToDoList

Calendar

Как использую DI



Как использую DI



ИТОГИ

✓ 3 Проблемы FSD

- Слишком изменчивый **shared**
- **Entities** не позволяют описать модель
- Widgets + Features + Entities = **high coupling**

ИТОГИ

- ✓ 3 Проблемы FSD
- ✓ Возможные решения
 - **Большие** Entities и Features
 - Правила импортов для сегментов
 - DI для связи между фичами

А до сих пор ли у меня FSD?

А нужно ли цепляться за FSD?



КОНТАКТЫ

Tg: @paromovevg

Youtube: @paromovevg