

Я Реклама

Glorious Monolith

Масштабируем приложение
без микрофронтендов

Максим Земсков
Ведущий разработчик интерфейсов, Яндекс



Что будет в докладе

01 Проблема масштабирования

02 Монолиты и микрофронтенды

03 Что такое Glorious Monolith

04 Принципы построения хорошего монолита

05 Выводы



спикер

nodge@yandex-team.ru

t.me/nodge74

linkedin.com/in/maxim-zemskov

Максим Земсков

Ведущий разработчик
интерфейсов в Яндекс

- В веб-разработке с 2010 года
- Frontend и Fullstack
- 9 лет в Яндексе
- Запускал новые сервисы и внедрял масштабные изменения в существующие
- Руководил инфраструктурной командой в Яндекс Директе
- Решил проблему масштабирования монолита

Frontend Яндекс Директа

22 года интенсивной
разработки

4 монолита
разных времен

50 разработчиков
интерфейсов

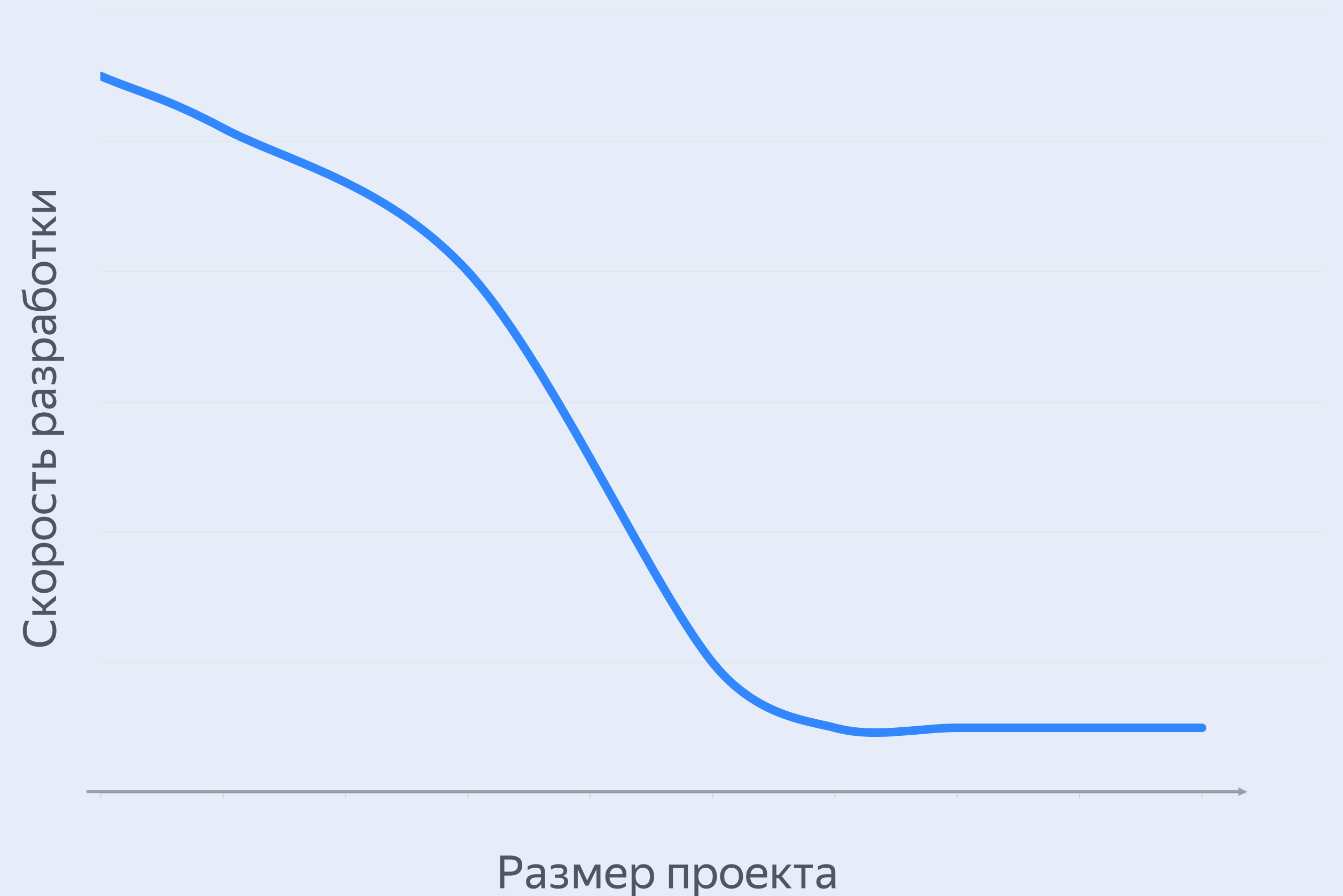
2 500 000
строк кода

>100
сложных страниц

О проблеме масштабирования

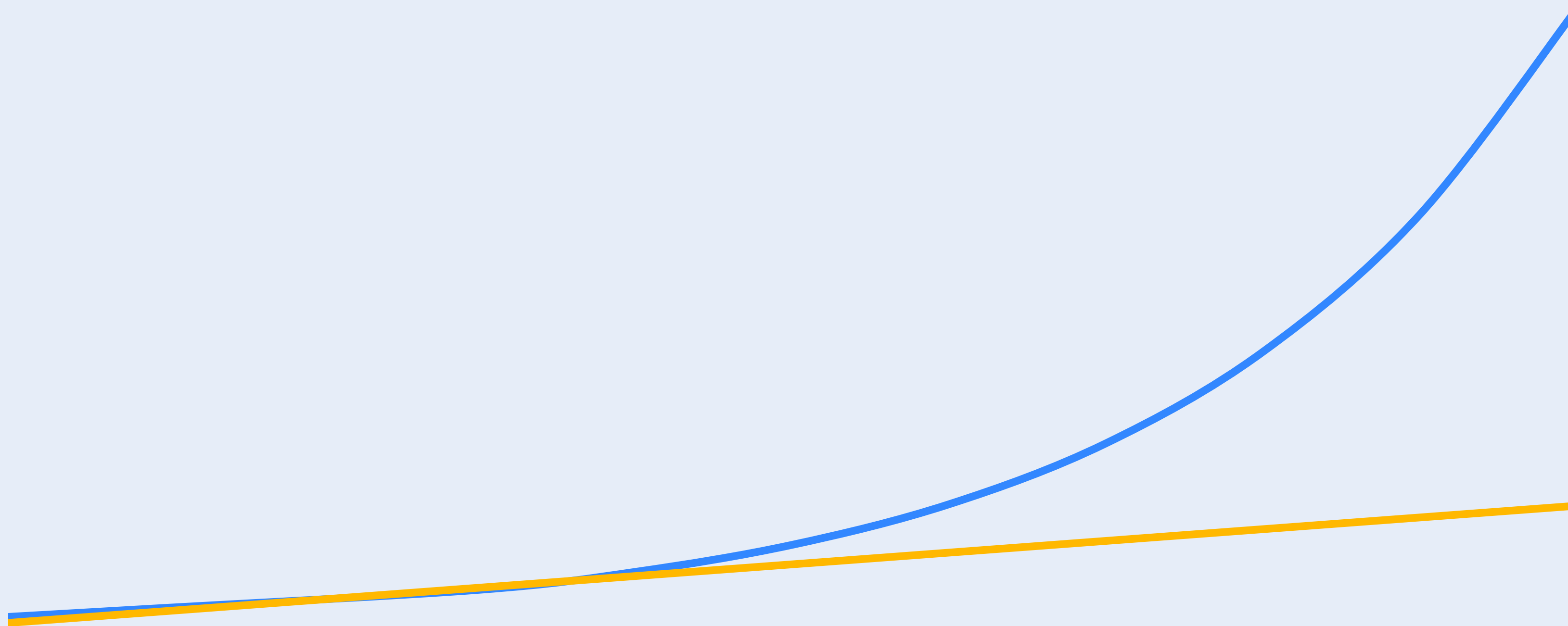
О проблеме масштабирования

- Проект перестает помещаться в голову разработчика
- Страшно вносить изменения
- Неожиданные баги при рефакторинге
- Сложно писать новый код
- Новые разработчики долго погружаются в проект
- Появляется куча решений для одних и тех же задач



Сложность в типичном монолите

— Сложность — Размер проекта



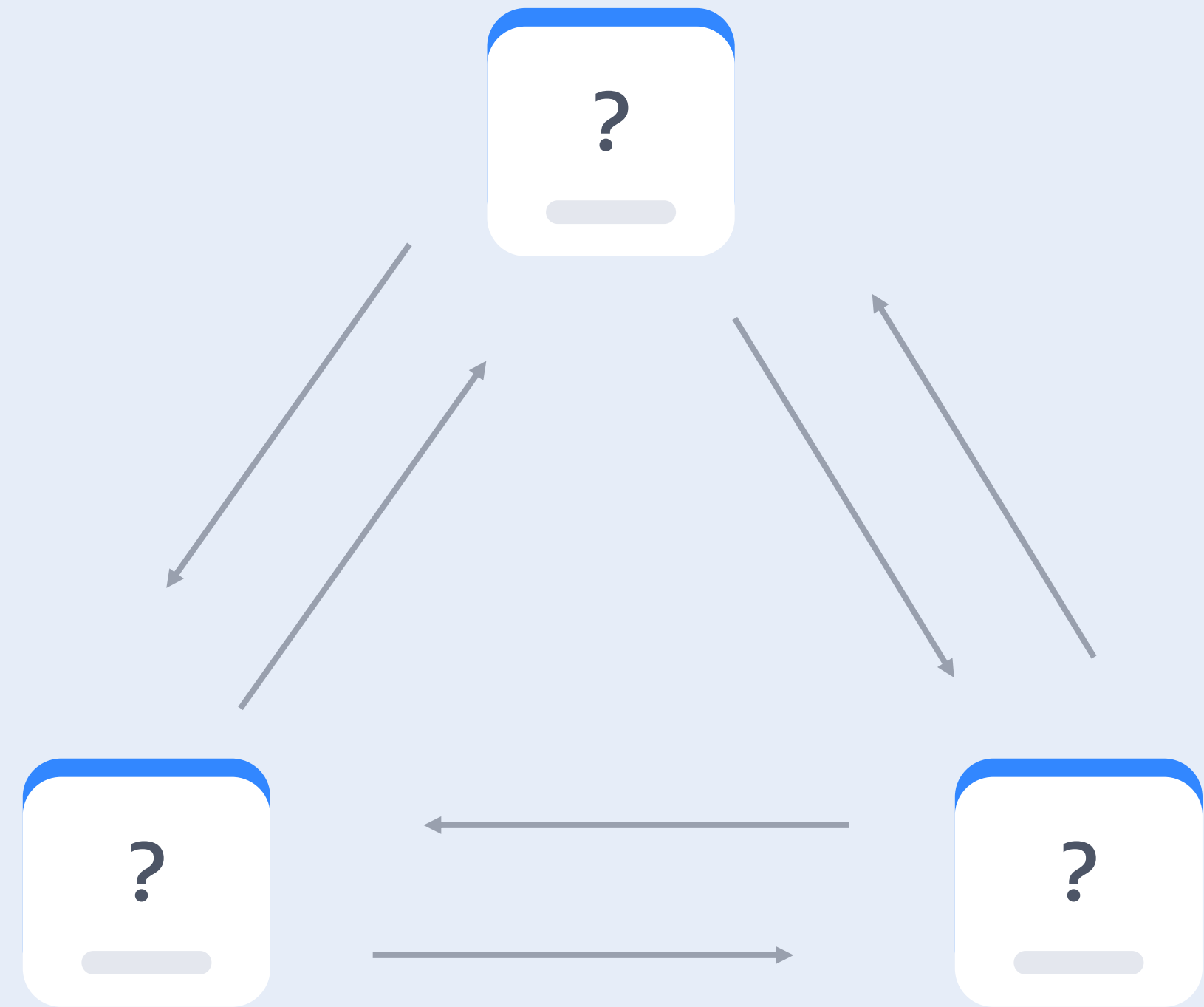
Откуда берется сложность

Объем кода × Связи (High Coupling)



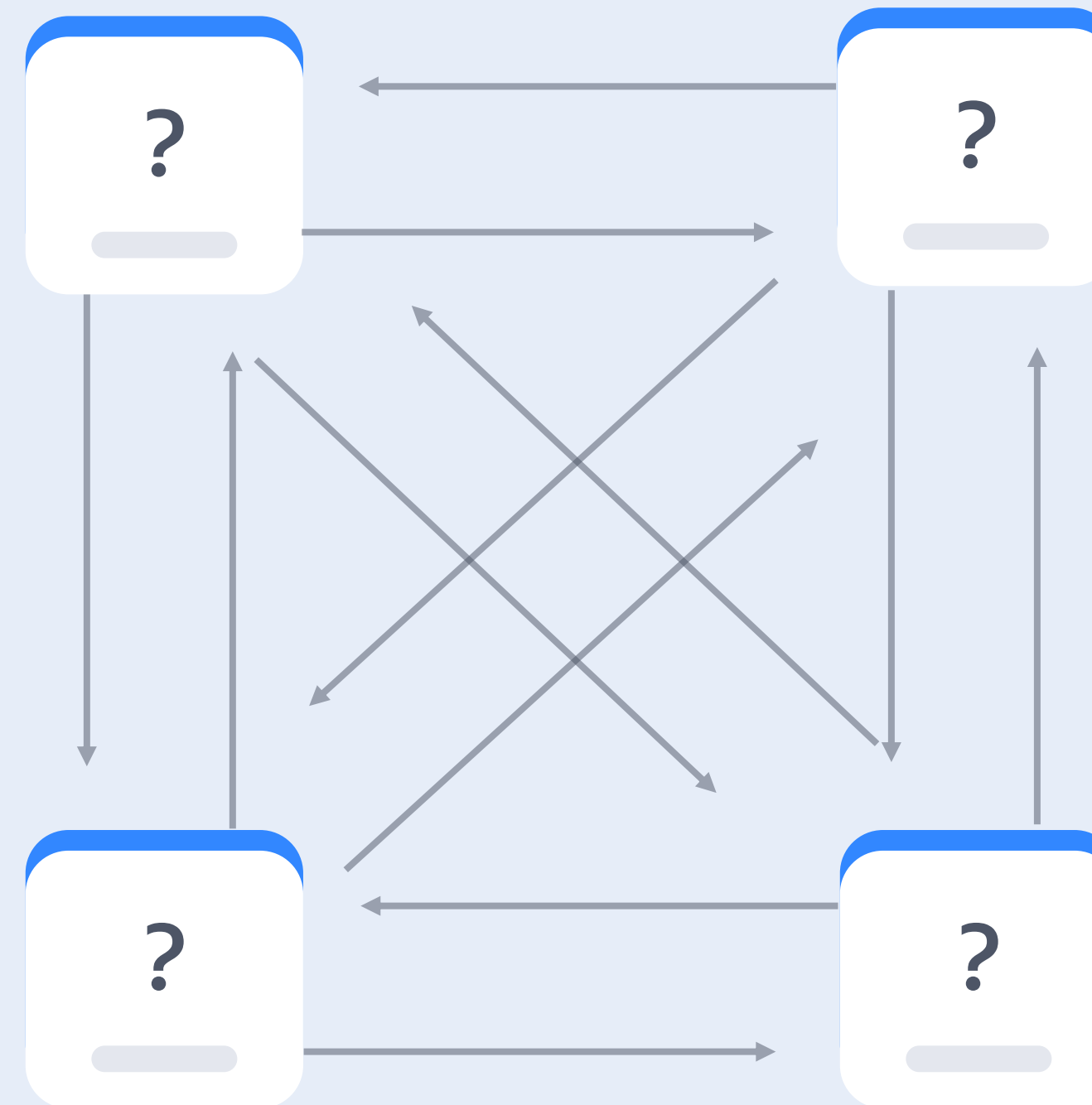
Откуда берется сложность

Объем кода × Связи (High Coupling)



Откуда берется сложность

Объем кода × Связи (High Coupling)



Откуда берется сложность

Объем кода × Связи (High Coupling)



imgflip.com



Добавим микрофронтенд?

Монолит

MF

MF

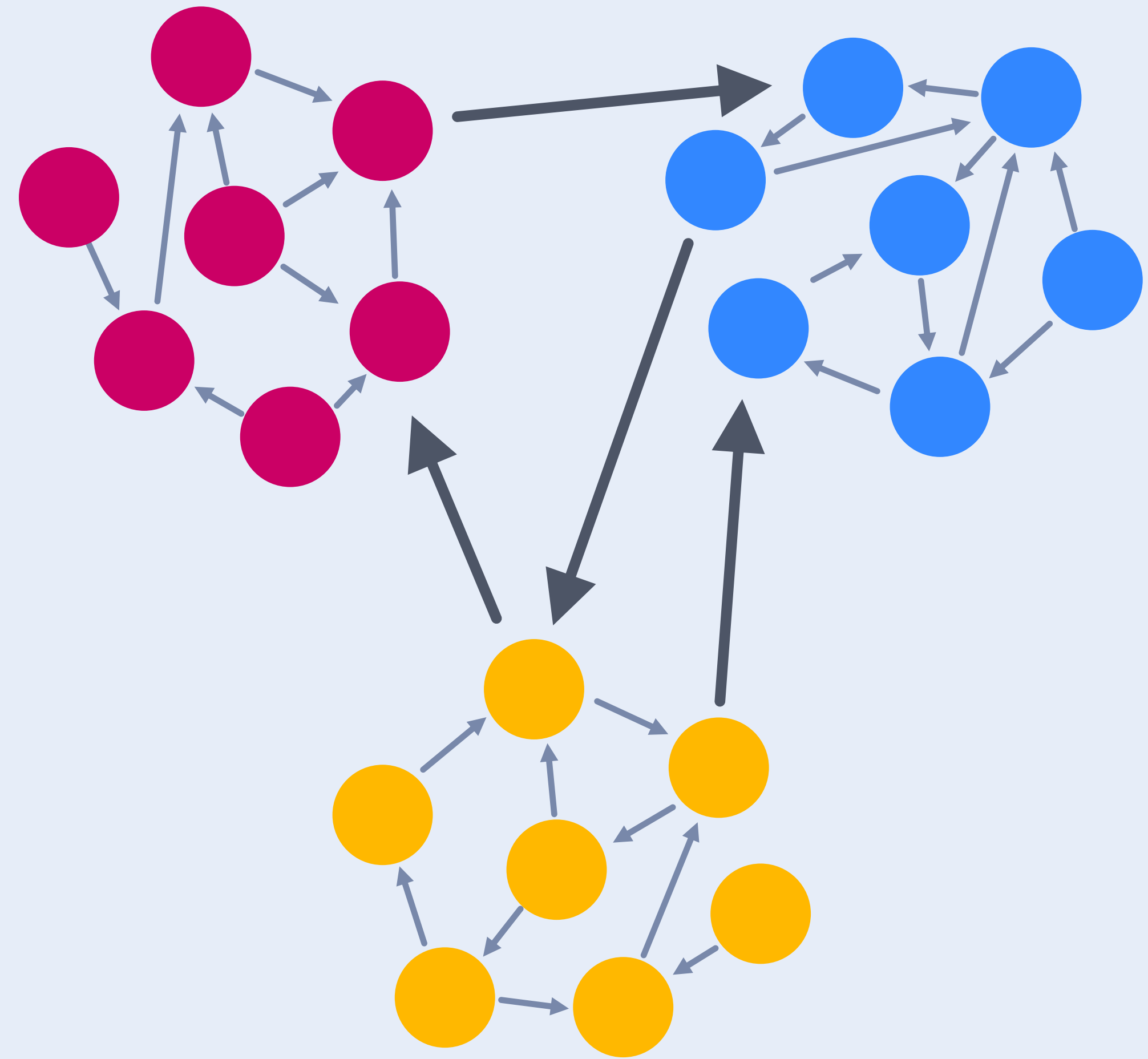
MF

MF

MF

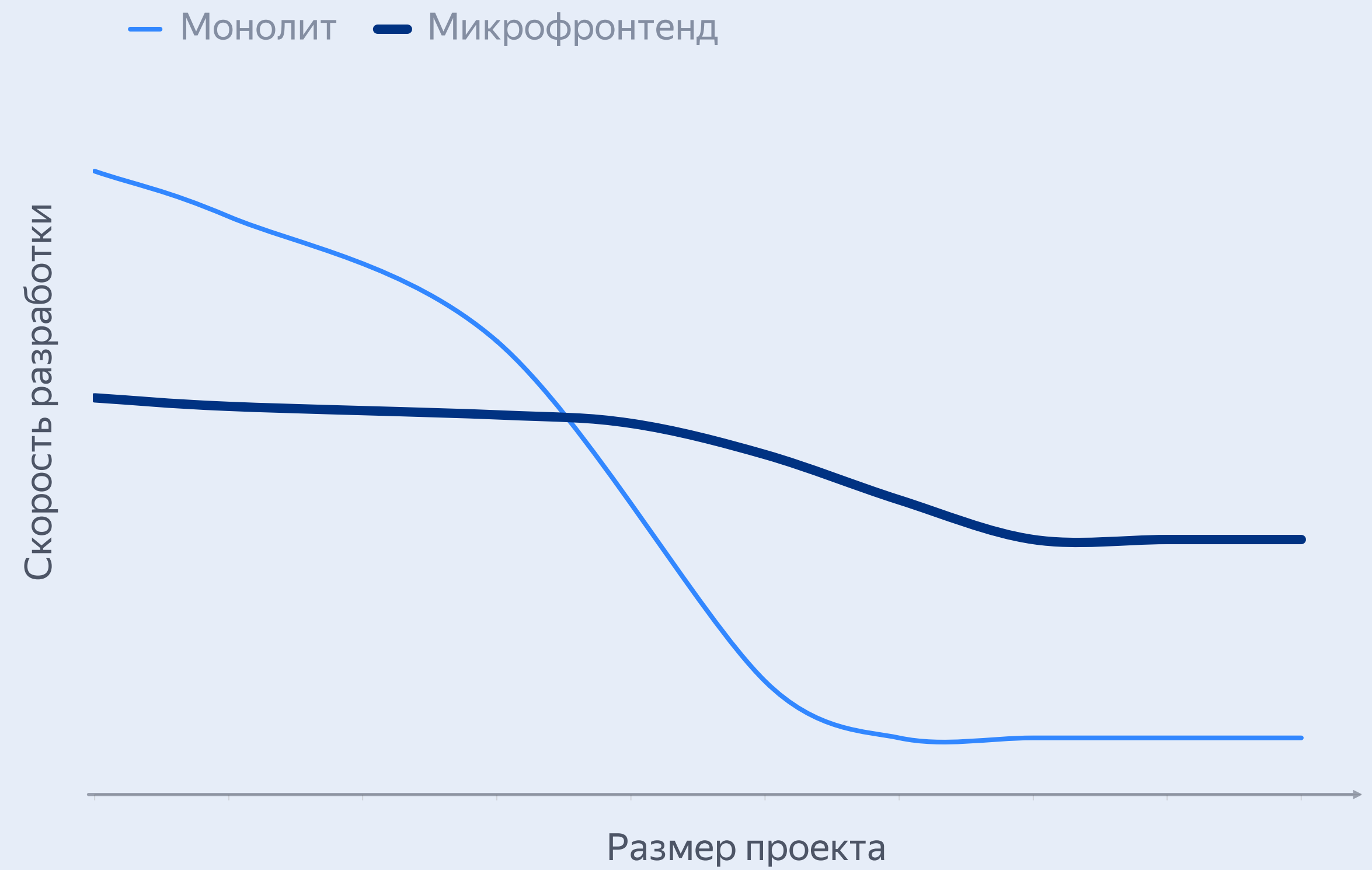
Плюсы микрофронтенда

- ✓ Low Coupling



Плюсы микрофронтенда

- ✓ Low Coupling
- ✓ Масштабируемость



Плюсы микрофронтенда

- ✓ Low Coupling
- ✓ Масштабируемость
- ✓ Разделение ответственности по командам

Отдельные правила и процессы для каждой команды

Раздельный релизный цикл

— Любой проект лучше с микрофронтендом?

— Нет 😊

Последствия микрофронтенда

- ⊗ Сложнее инфраструктура
- ⊗ Сложнее переиспользовать код

монорепозиторий

публикация пакетов

npm

rollup

webpack

module federation

turbopack

vite

Последствия микрофронтенда

- ⊗ Сложнее инфраструктура
- ⊗ Сложнее переиспользовать код
- ⊗ Для SSR нужны микросервисы

монорепозиторий

rollup

webpack

трейсинг

микросервисы

npm

мониторинг

turbopack

скорость приложения

публикация пакетов
эксплуатация

module federation

vite

Последствия микрофронтенда

- ⊗ Сложнее инфраструктура
- ⊗ Сложнее переиспользовать код
- ⊗ Для SSR нужны микросервисы
- ⊗ Сложнее релизный процесс

монорепозиторий
управление версиями
npm
rollup
эксплуатация
публикация пакетов
мониторинг
webpack
module federation
трейсинг
turbopack
vite
микросервисы
скорость приложения
прямая совместимость
обратная совместимость

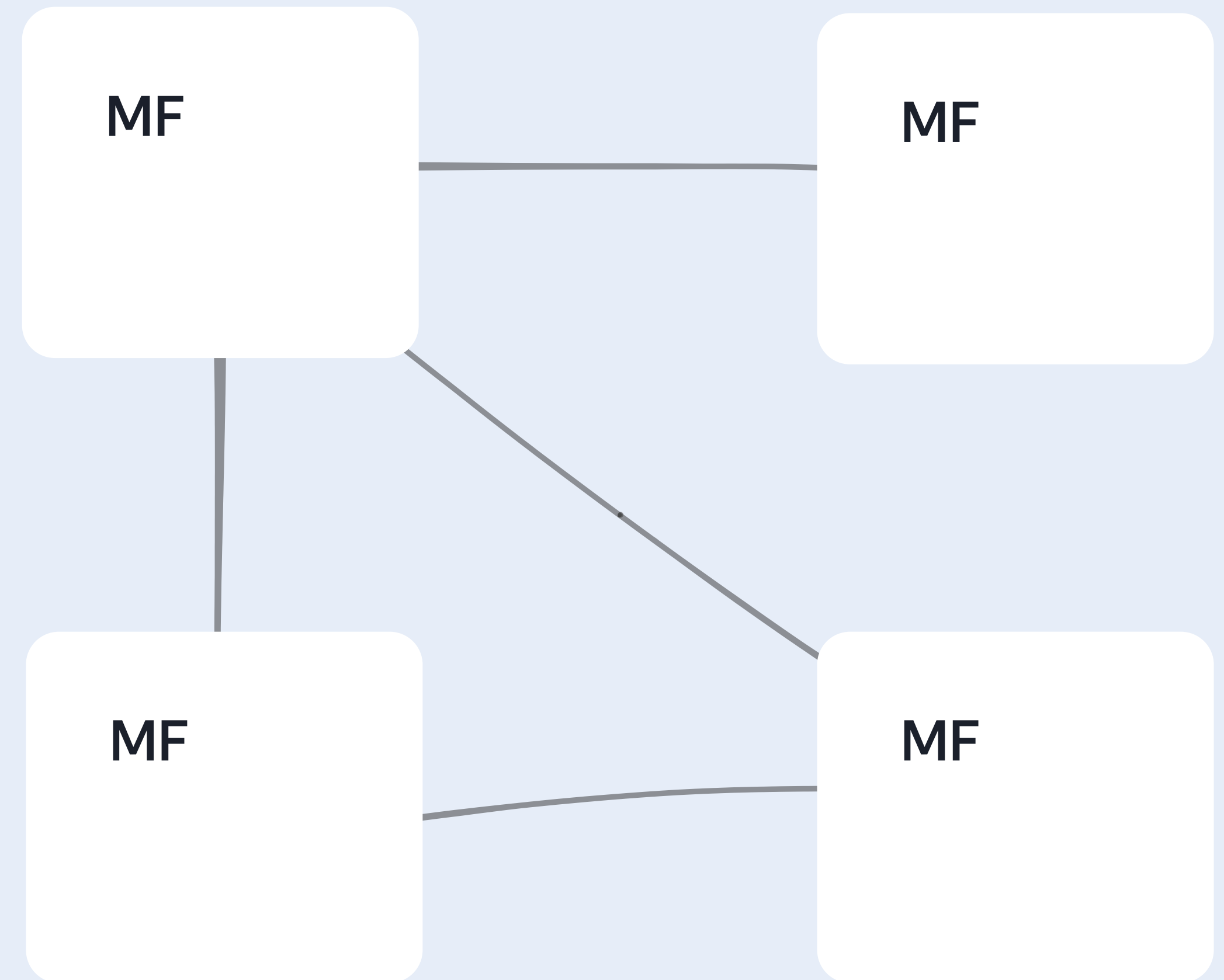
Последствия микрофронтенда

- ⊗ Сложнее инфраструктура
- ⊗ Сложнее переиспользовать код
- ⊗ Для SSR нужны микросервисы
- ⊗ Сложнее релизный процесс
- ⊗ Зоопарк технологий и решений
- ⊗ Решение одних и тех же задач разными командами

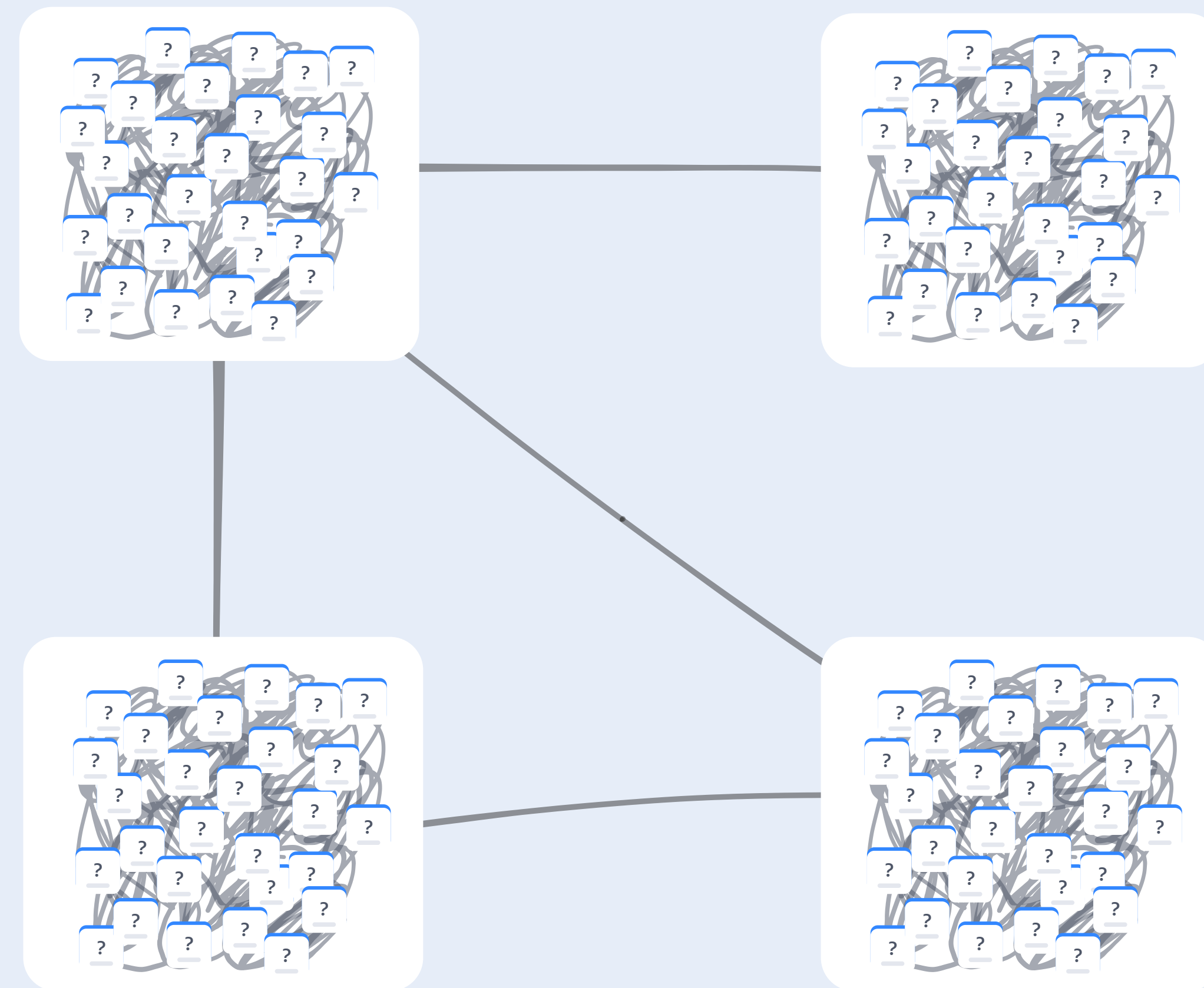
монорепозиторий
управление версиями
npm
публикация пакетов
эксплуатация
rollup
webpack
мониторинг
module federation
трейсинг
микросервисы
turbopack
vite
скорость приложения
прямая совместимость
балансировка ресурсов
обратная совместимость
размер бандлов
консистентность UI
переключение между проектами

Микрофронтенд не всегда микро

Монолит



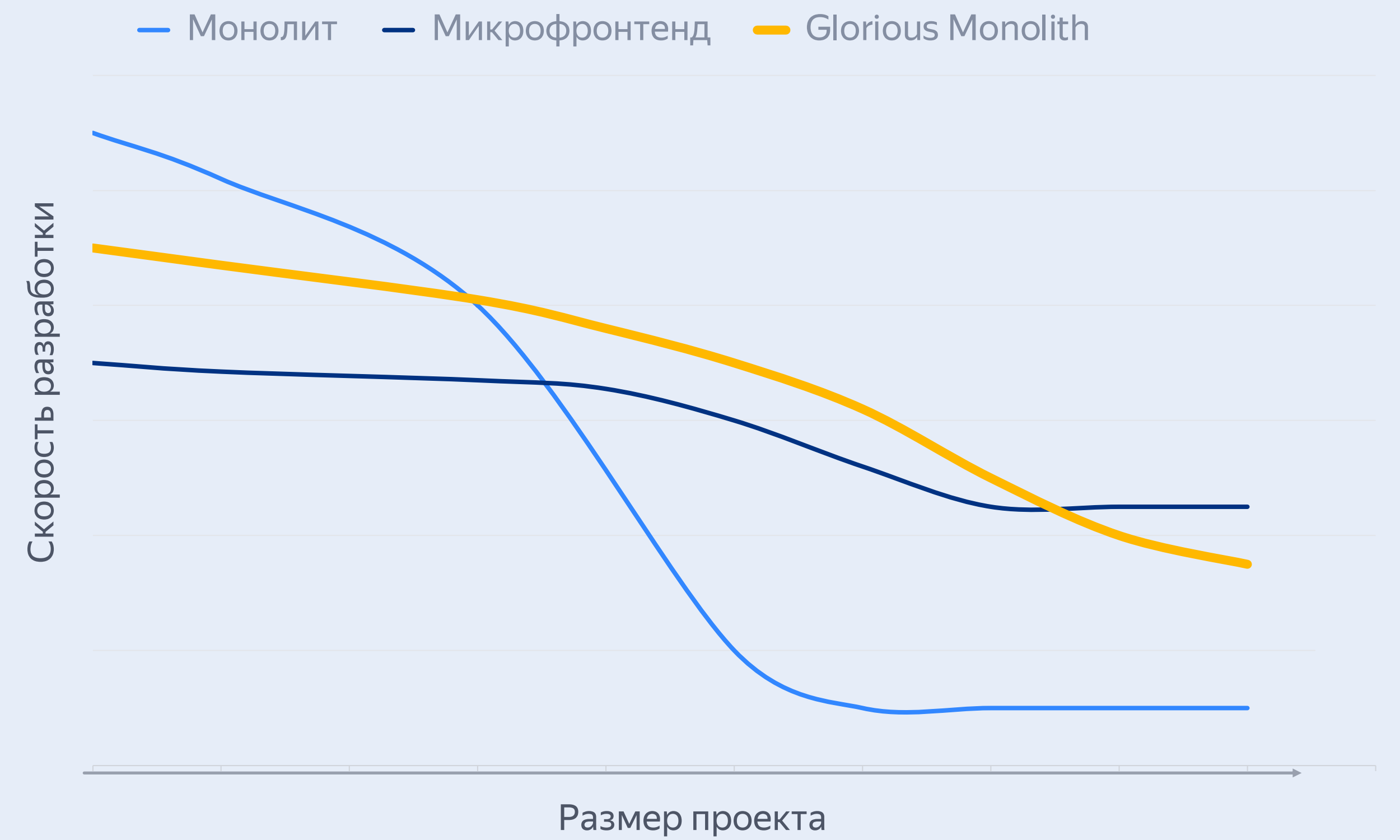
Микрофронтенд не всегда микро



Glorious Monolith

Glorious Monolith

- ✓ Относительно просто реализовать
- ✓ Хорошо переживает рост кодовой базы



Glorious Monolith

Физическая
архитектура

Монолит

Логическая
архитектура

Микрофронтенд

- **Что нужно для
Glorious Monolith?**
- **Правила
и ограничения**

Путь к Glorious Monolith

01

Модуль

- Структура
- Изоляция
- Public API



02

Runtime для модулей



03

Границы между модулями



Добавить

Атрибуция: Автоматическая



Настроить порядок виджетов

Max Zemskov

3 900 075,45 ₽

Пополнить

Сквозная аналитика из Яндекс Метрики

Данные за последние 30 дней

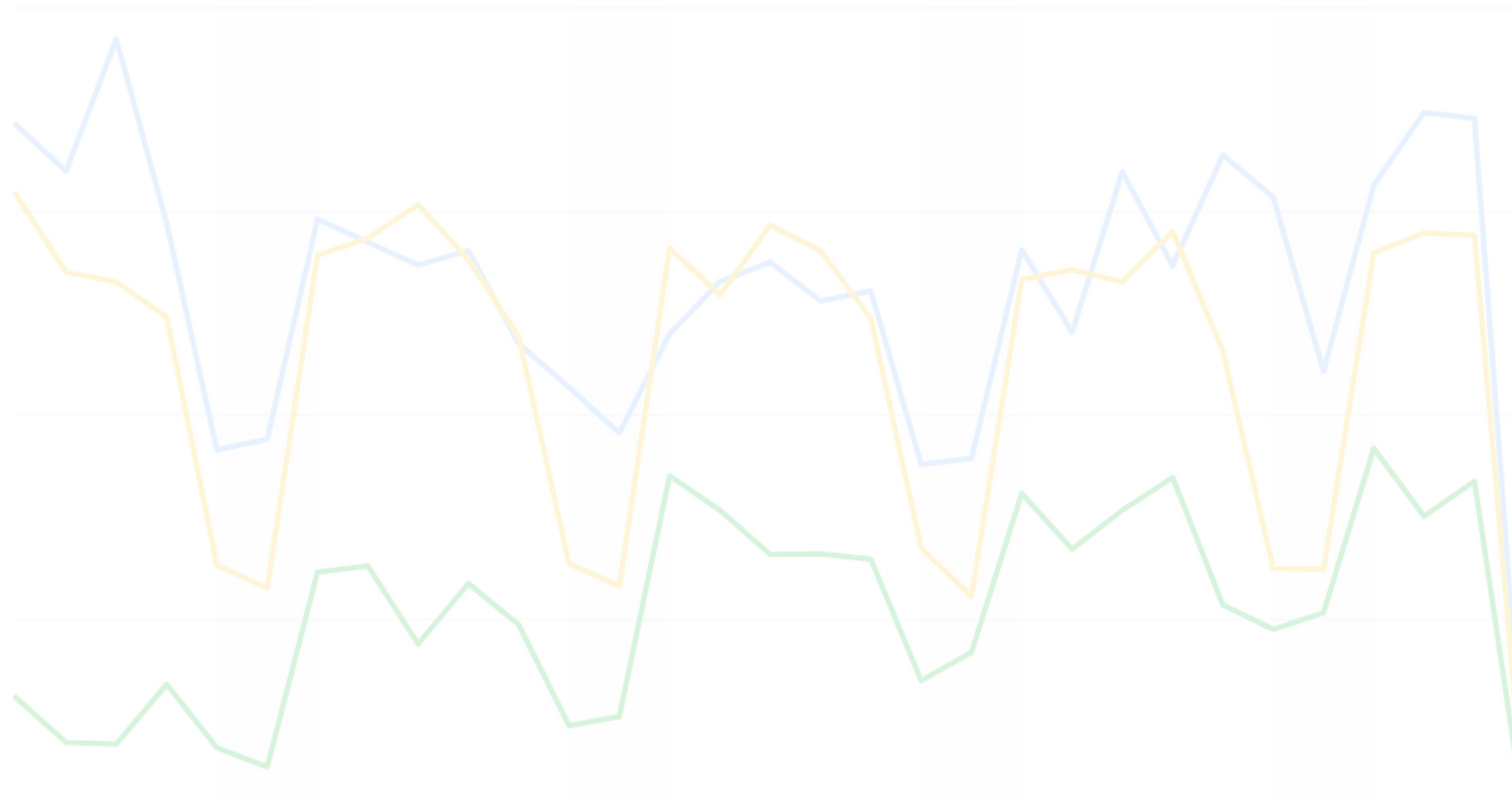
Клики

Конверсии

Доходы

nodge.ru • 34

Источник: Яндекс Метрика



<input checked="" type="checkbox"/>	Другие источники	21 663
<input checked="" type="checkbox"/>	Переходы из поисковых сис...	18 652
<input checked="" type="checkbox"/>	Яндекс Директ	8 555
<input checked="" type="checkbox"/>	ВКонтакте	308
<input checked="" type="checkbox"/>	Zen	82
<input checked="" type="checkbox"/>	Other ad: identified by tags	41
<input checked="" type="checkbox"/>	ya.ru	39
<input checked="" type="checkbox"/>	Telegram	27
<input checked="" type="checkbox"/>	Реклама ВКонтакте	25

Кампании

Рекомендации

Конверсии

Статистика

Библиотека

Инструменты

Информация

Поиск

Администрирование

Добавить

Атрибуция: Автоматическая



Настроить порядок виджетов

Max Zemskov

3 900 075,45 ₽

Пополнить

Сквозная аналитика из Яндекс Метрики

Данные за последние 30 дней

Обзор

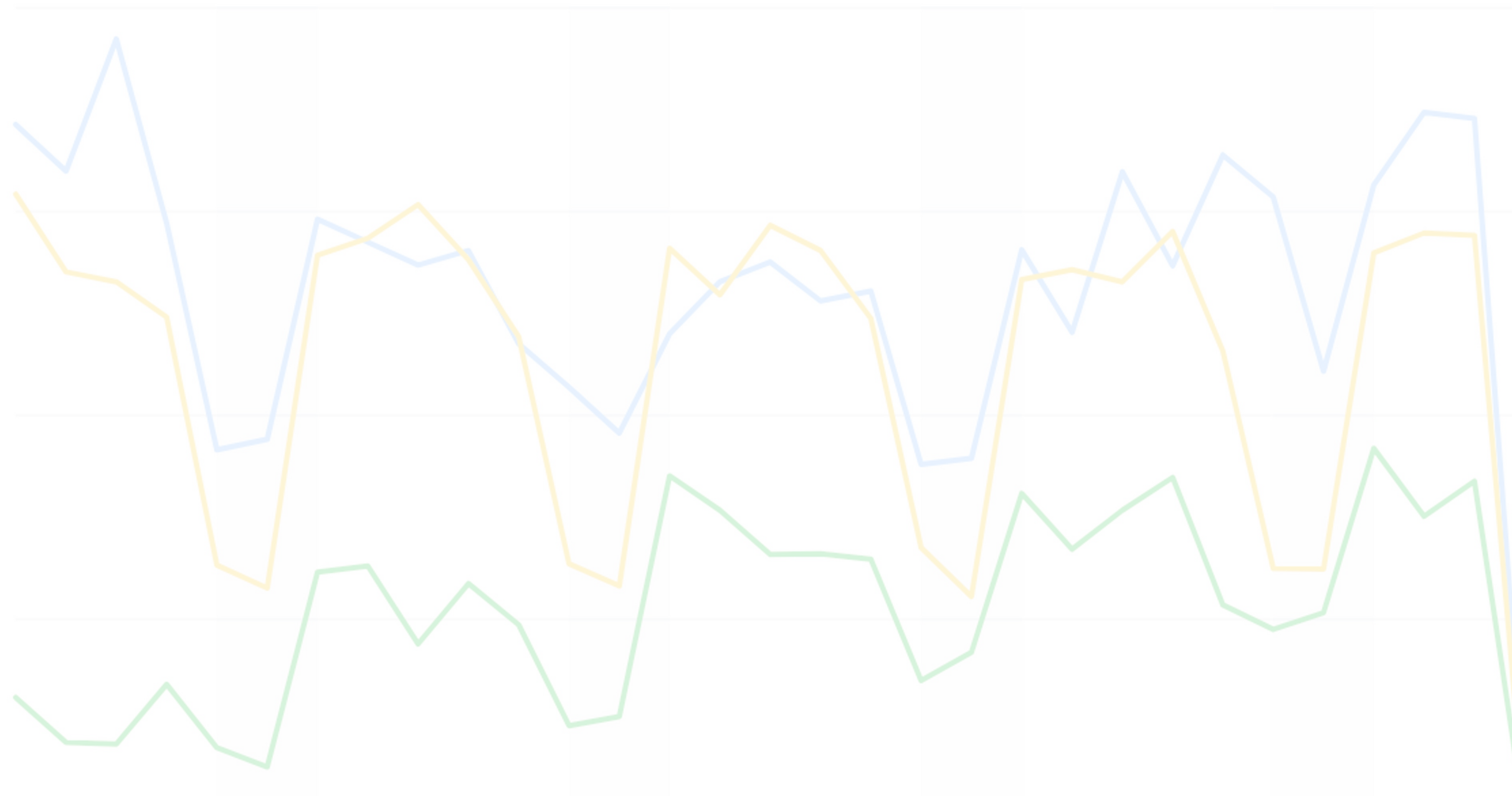
Клики

Конверсии

Доходы

nodge.ru • 34

Источник: Яндекс Метрика



<input checked="" type="checkbox"/>	Другие источники	21 663
<input checked="" type="checkbox"/>	Переходы из поисковых сис...	18 652
<input checked="" type="checkbox"/>	Яндекс Директ	8 555
<input checked="" type="checkbox"/>	ВКонтакте	308
<input checked="" type="checkbox"/>	Zen	82
<input checked="" type="checkbox"/>	Other ad: identified by tags	41
<input checked="" type="checkbox"/>	ya.ru	39
<input checked="" type="checkbox"/>	Telegram	27
<input checked="" type="checkbox"/>	Реклама ВКонтакте	25

Кампании

Рекомендации

Конверсии

Статистика

Библиотека

Инструменты

Информация

Поиск

Администрирование

Добавить

Атрибуция: Автоматическая



Настроить порядок виджетов

Max Zemskov

3 900 075,45 ₽

Пополнить

Обзор

Кампании

Рекомендации

Конверсии

Статистика

Библиотека

Инструменты

Информация

Поиск

Администрирование

Сквозная аналитика из Яндекс Метрики

Данные за последние 30 дней

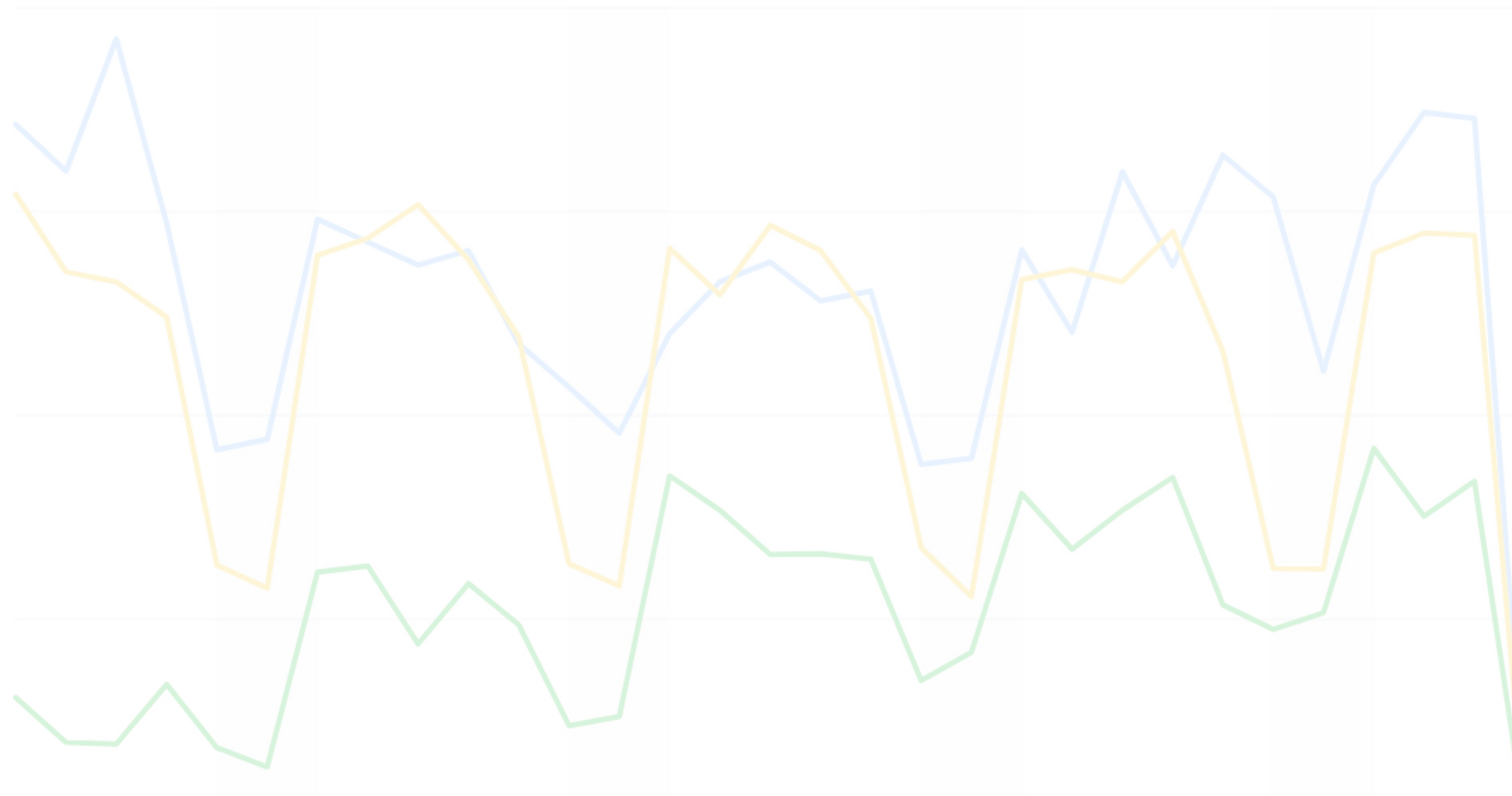
Клики

Конверсии

Доходы

nodge.ru • 34

Источник: Яндекс Метрика



<input checked="" type="checkbox"/>	Другие источники	21 663
<input checked="" type="checkbox"/>	Переходы из поисковых сис...	18 652
<input checked="" type="checkbox"/>	Яндекс Директ	8 555
<input checked="" type="checkbox"/>	ВКонтакте	308
<input checked="" type="checkbox"/>	Zen	82
<input checked="" type="checkbox"/>	Other ad: identified by tags	41
<input checked="" type="checkbox"/>	ya.ru	39
<input checked="" type="checkbox"/>	Telegram	27
<input checked="" type="checkbox"/>	Реклама ВКонтакте	25

Добавить

Max Zemskov

3 900 075,45 ₽

Пополнить

Обзор β

Кампании

Рекомендации

Конверсии

Статистика

Библиотека

Инструменты

Информация

Поиск

Администри-
рование

Обзор результатов продвижения в Директ

Атрибуция: Автоматическая ▼



Настроить порядок виджетов

Сквозная аналитика из Яндекс Метрики ?

Данные за последние 30 дней

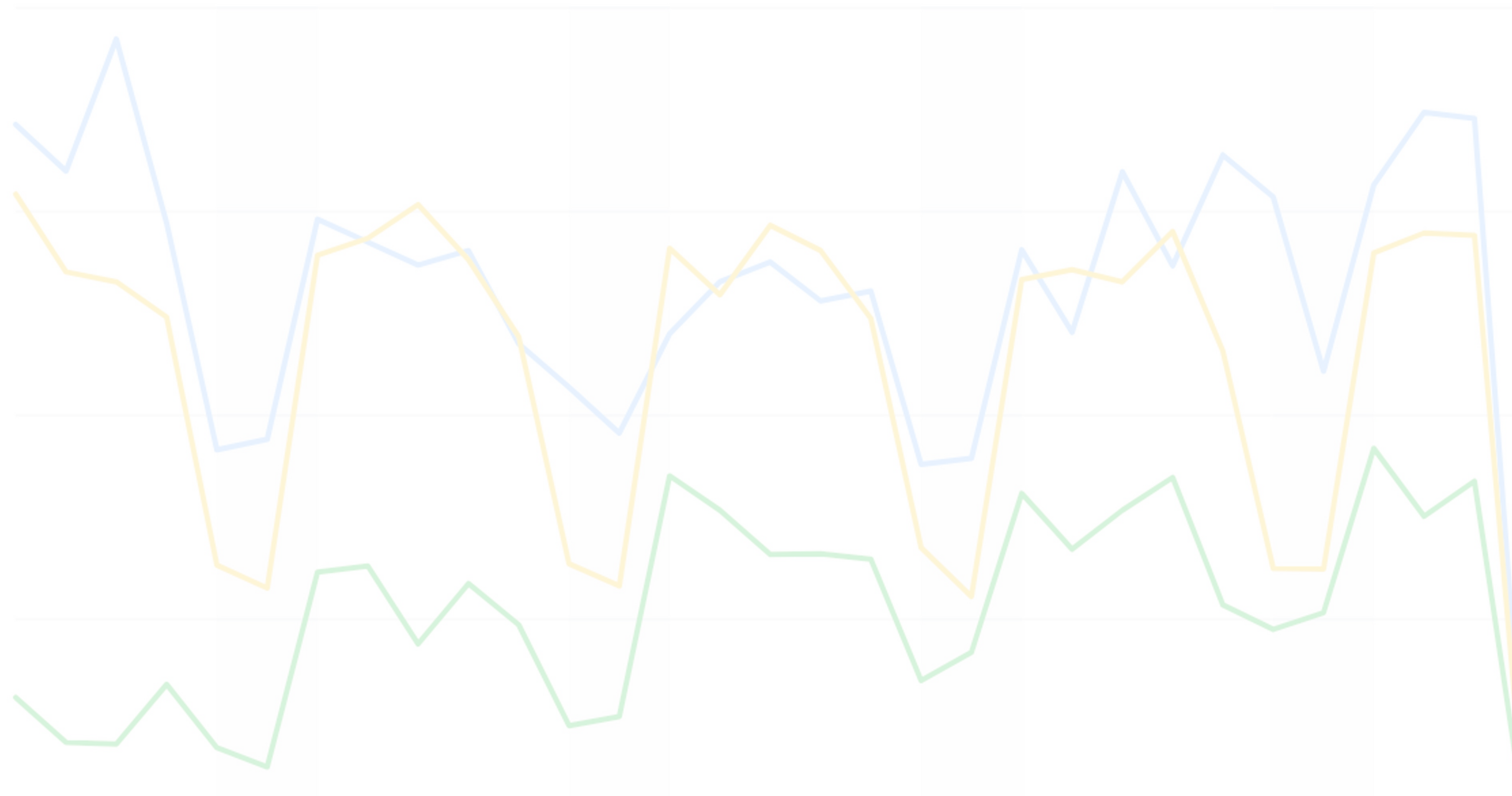
Клики

Конверсии

Доходы

nodge.ru • 34 ▼

Источник: Яндекс Метрика



<input checked="" type="checkbox"/>	Другие источники	21 663
<input checked="" type="checkbox"/>	Переходы из поисковых сис...	18 652
<input checked="" type="checkbox"/>	Яндекс Директ	8 555
<input checked="" type="checkbox"/>	ВКонтакте	308
<input checked="" type="checkbox"/>	Zen	82
<input checked="" type="checkbox"/>	Other ad: identified by tags	41
<input checked="" type="checkbox"/>	ya.ru	39
<input checked="" type="checkbox"/>	Telegram	27
<input checked="" type="checkbox"/>	Реклама ВКонтакте	25

Добавить

Max Zemskov

3 900 075,45 ₽

Пополнить

Обзор β

Кампании

Рекомендации

Конверсии

Статистика

Библиотека

Инструменты

Информация

Поиск

Администри-
рование

Обзор результатов продвижения в Директ

Атрибуция: Автоматическая ▼



Настроить порядок виджетов

Сквозная аналитика из Яндекс Метрики ?

Данные за последние 30 дней

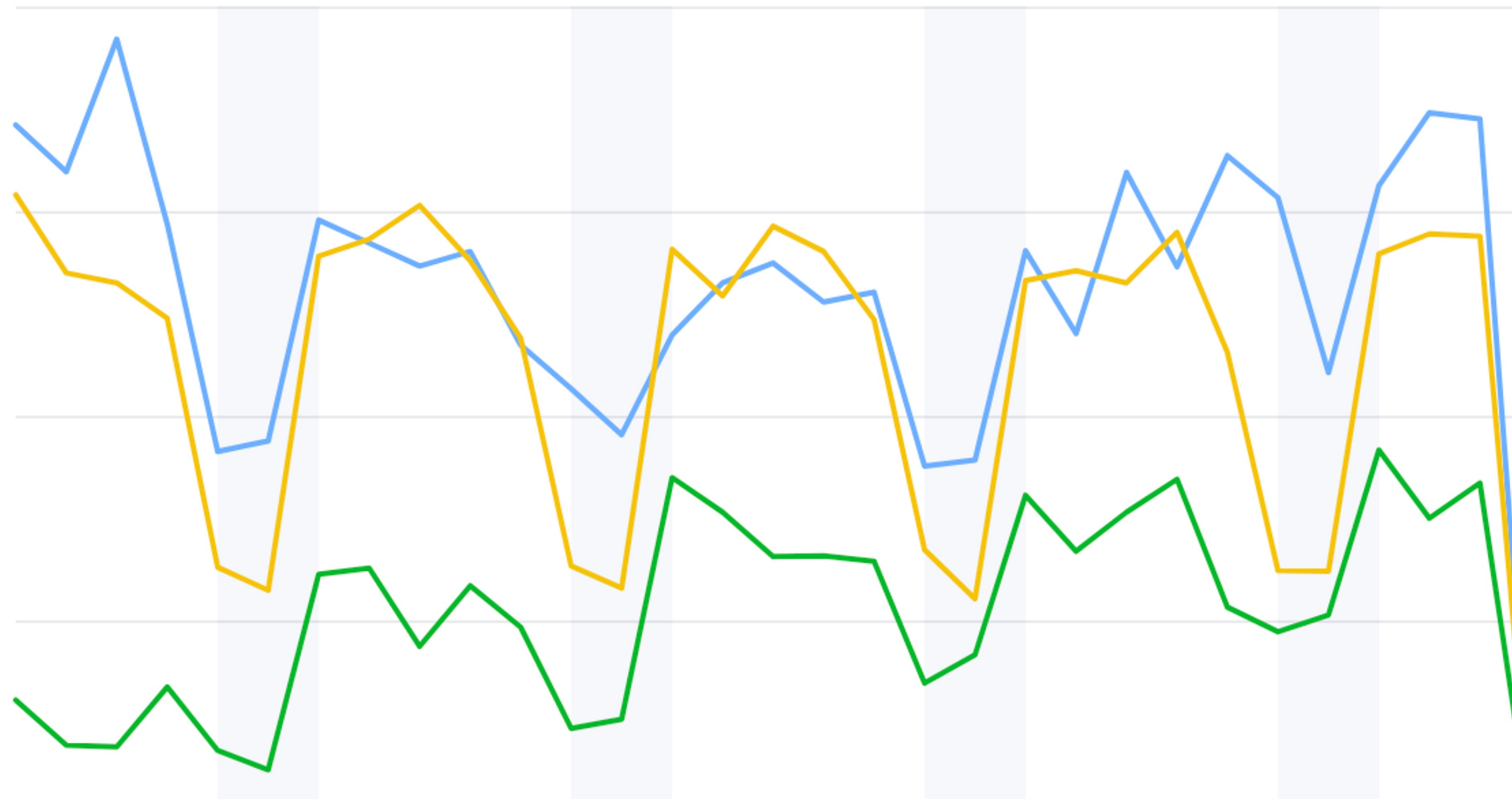
Клики

Конверсии

Доходы

nodge.ru • 34 ▼

Источник: [Яндекс Метрика](#)



<input checked="" type="checkbox"/>	Другие источники	21 663
<input checked="" type="checkbox"/>	Переходы из поисковых сис...	18 652
<input checked="" type="checkbox"/>	Яндекс Директ	8 555
<input checked="" type="checkbox"/>	ВКонтакте	308
<input checked="" type="checkbox"/>	Zen	82
<input checked="" type="checkbox"/>	Other ad: identified by tags	41
<input checked="" type="checkbox"/>	ya.ru	39
<input checked="" type="checkbox"/>	Telegram	27
<input checked="" type="checkbox"/>	Реклама ВКонтакте	25

Правила для модулей

Структура
модуля

01

Изоляция
модуля

02

Коммуникация
между модулями

03

01

Структура модуля

Что внутри модуля

Весь код проекта раскладывается по модулям

Продукт

UI Компоненты

Бизнес-логика

Запросы к API

CSS

Шрифты, картинки

Серверный код

Инфраструктура

Любые тесты

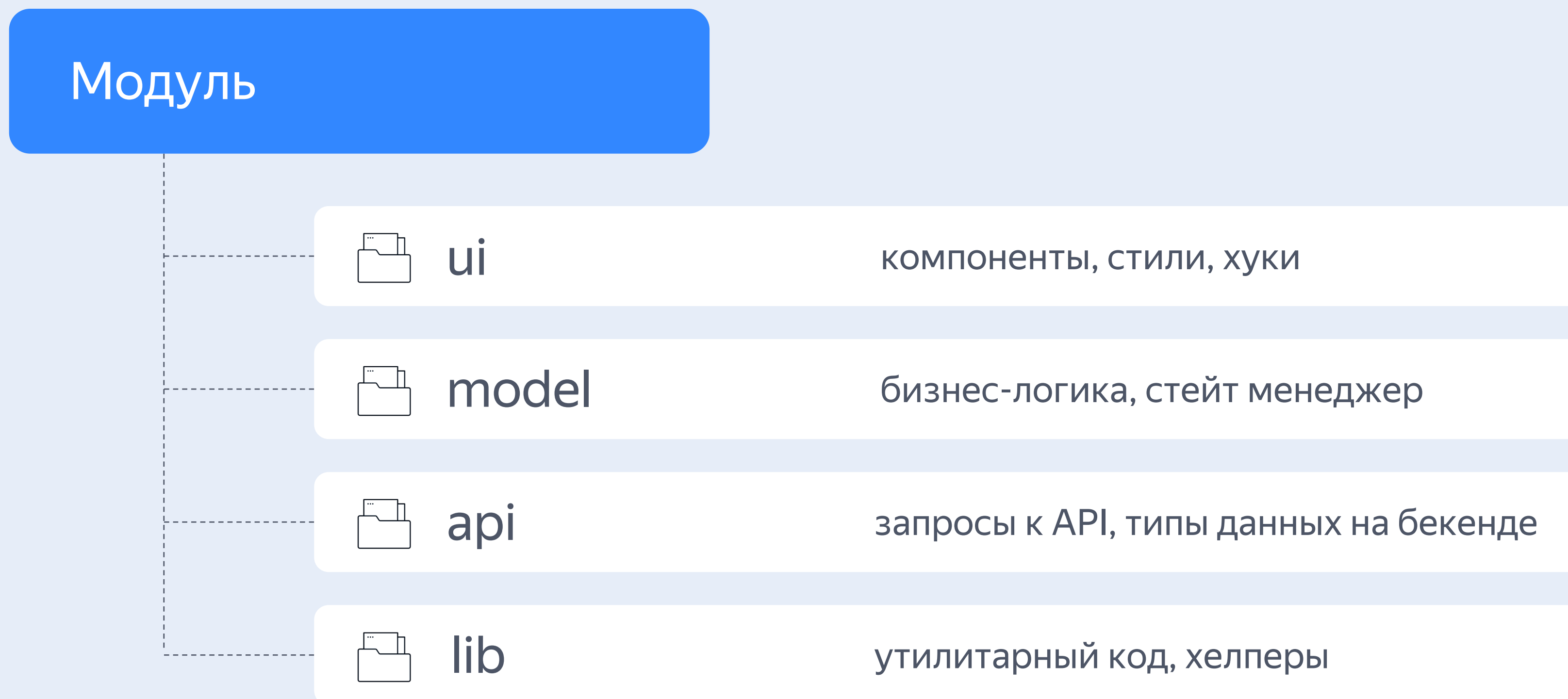
Документация

Утилитарный код

Фреймворк

Библиотеки

Структура модуля



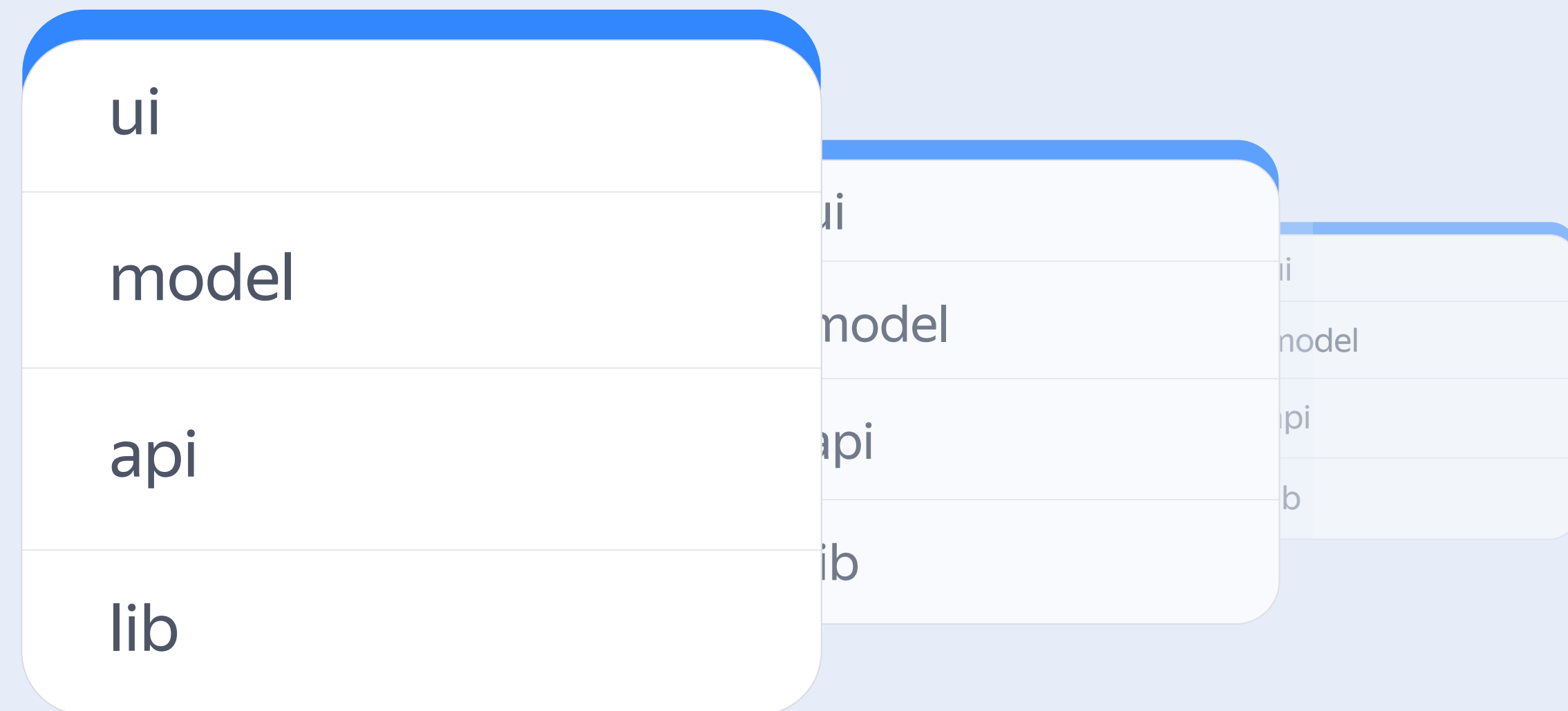
Однонаправленные импорты



Сегмент	Может использовать	Может быть использован
UI	ui, model, api, lib	ui
MODEL	model, api, lib	ui, model
API	api, lib	ui, model, api
LIB	lib	ui, model, api, lib

Что дают сегменты

- ✓ Унификация кодовой базы
- ✓ Легко решить куда поселить новый код
- ✓ Быстрое переключение команды между модулями
- ✓ Колокация связанного кода
- ✓ Защита от смешивания UI и бизнес-логики



02

Изоляция кода

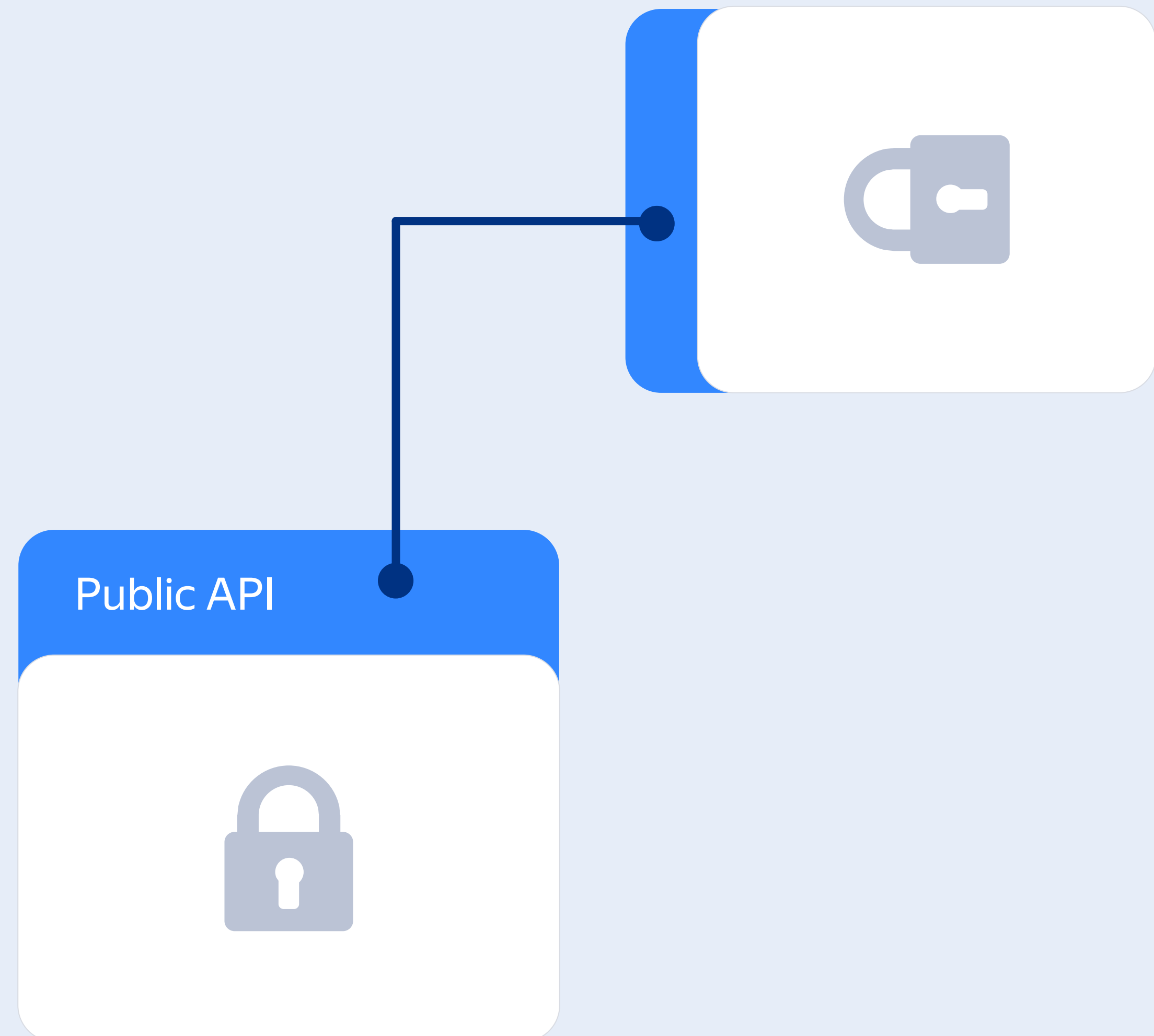
Изоляция модуля

Зачем изолируем?

- ✓ Low Coupling
- ✓ Безопасное внесение изменений

Что изолируем?

- Код
- Стили
- Данные



Изоляция кода

Отсутствие сайд-эффектов

- ✘ Доступ к глобальным объектам
- ✘ Загрузка полифилов
- ✘ Мутация DOM

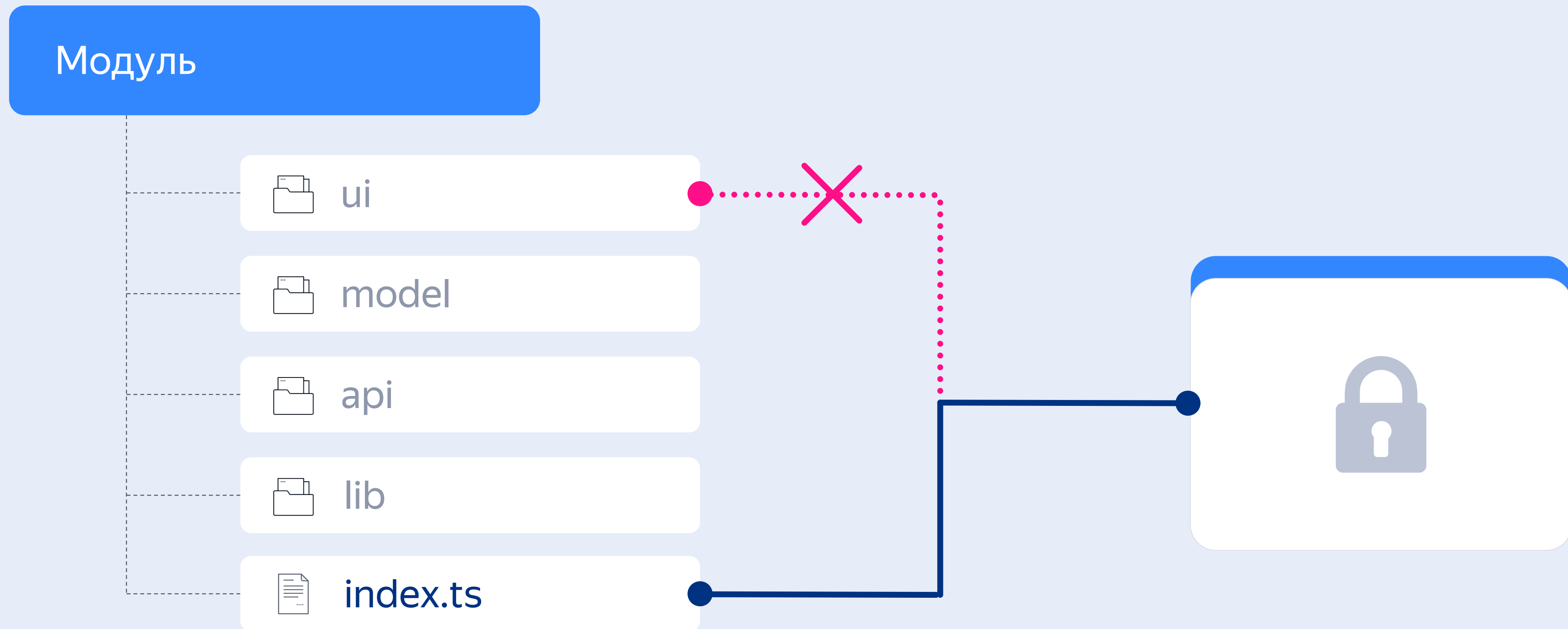
01

Контроль доступа к функциональности модуля

- ✘ Импорт файлов из модуля (код, стили, ассеты)
- ✘ Прямой доступ к хранилищу данных (idb, local storage и тд)

02

Entrypoint



Public API

ES модули

- ✓ Самый простой вариант
- ✓ Отлично дружит с инструментарием
- ✗ Сильнее связи между модулями

Dependency Injection

- ✓ Менее жесткие связи между модулями
- ✗ Усложняет инфраструктуру

Событийная модель

- ✓ Проще переход к микрофронтенду
- ✗ Нужно вручную контролировать загрузку модулей

Пример Public API



index.ts

```
// React компоненты
export { AddToCart } from './ui/AddToCart';
export { LazyCartView } from './ui/CartView.lazy';

// Методы для мутации данных в модели
export { clearCart } from './model/actions';

// Селекторы данных (хуки, геттеры)
export { useProductCount, getProductCount } from './model/selectors';

// Event Emitter
export { cartChangedSignal } from './model/cartChangedSignal';
```


Размер Public API

Количество экспортов

Больше экспортов →
больше связей в системе

Количество параметров в экспортируемых методах

Больше параметров →
сложнее использовать модуль

Сложность экспортируемых структур данных

Больше данных доступно →
сложнее рефакторинг модуля

Низкая сложность

Высокая сложность

Размер Public API



03

Изоляция CSS

Изоляция CSS



Module A

```
.button {  
  margin: 16px 8px 0;  
  --button-color: rgb(163, 172, 190);  
}
```



Module B

```
.button {  
  margin: 0;  
  --button-color: rgb(67, 94, 202);  
}
```

Варианты изоляции CSS

Shadow DOM

01

CSS Modules

02

CSS-in-JS

03

Tailwind CSS *

04

* когда является частью рантайма

Пример правил для CSS Modules

- ✓ Только селекторы классов, псевдо-классов и псевдо-элементов
- ✓ Миксуем стили через `className` в компонентах Public API
- ✓ Стремимся не переопределять стили для других модулей
- ✓ Stylelint для проверки договоренностей
- ✗ Не сбрасываем браузерные стили
- ✗ Не используем CSS переменные
- ✗ Не импортируем стили между модулями
- ✗ Не используем стили 3rd party библиотек

04

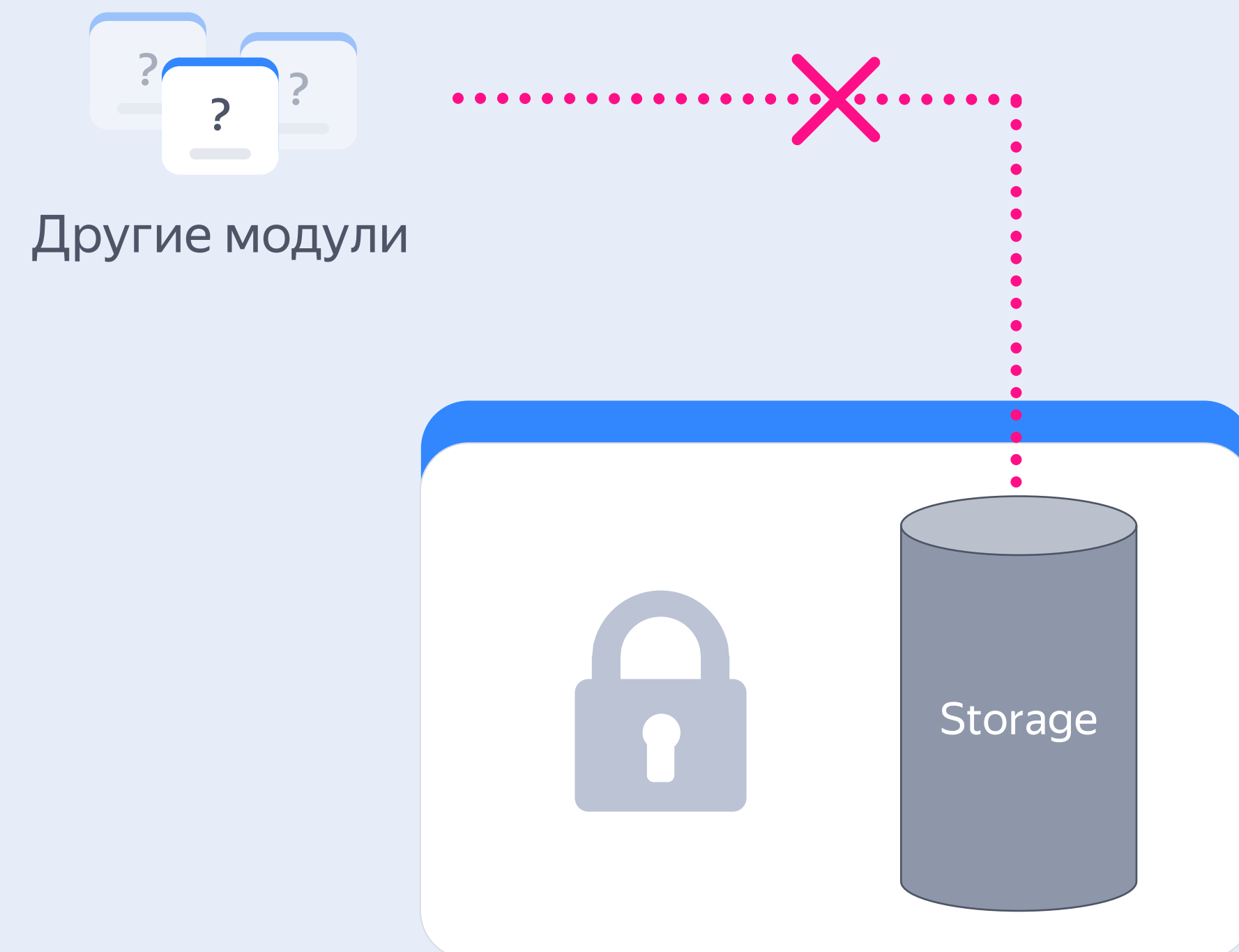
Изоляция данных

Изоляция данных

Глобальный стор с данными



Прямой доступ к данным модуля

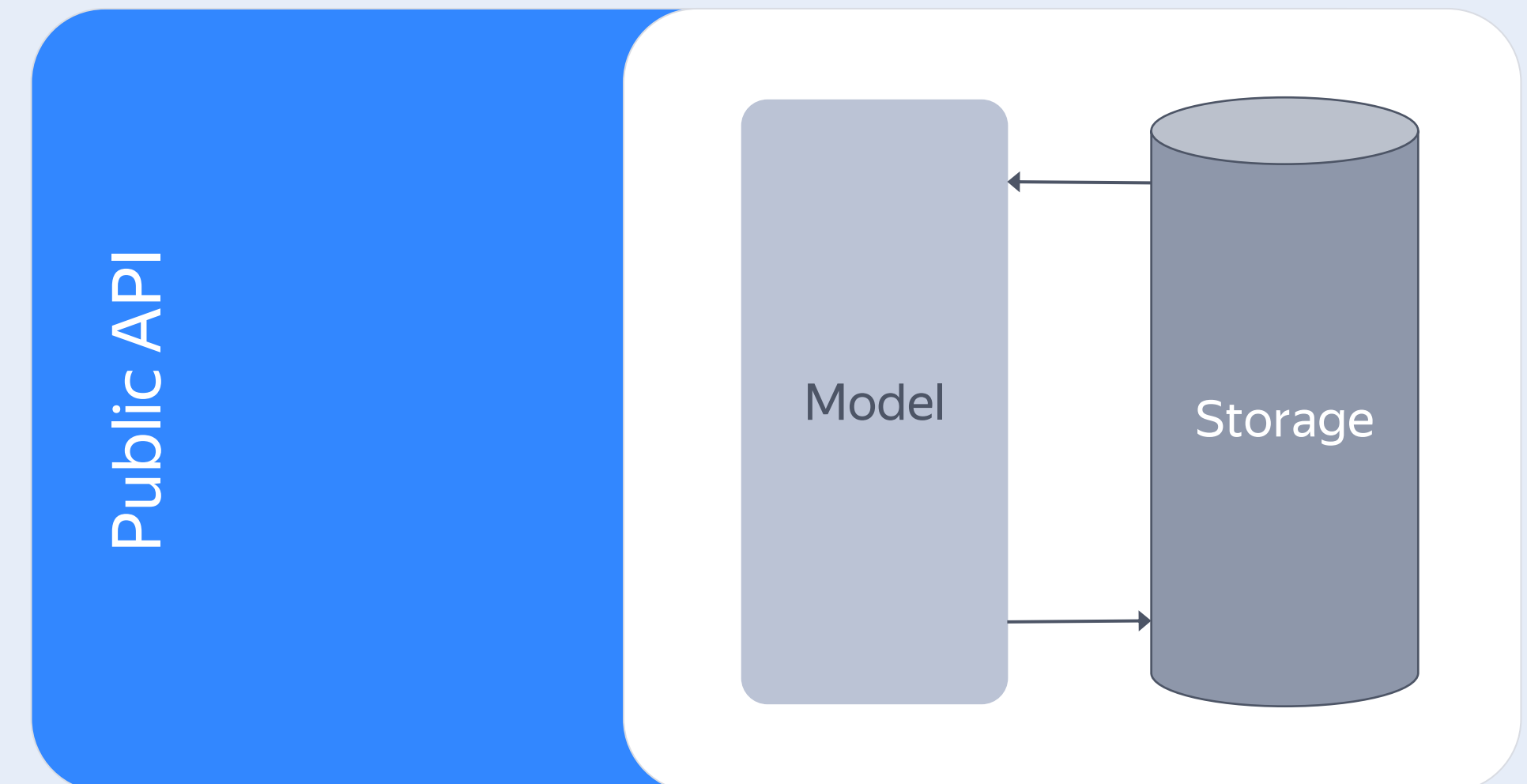


Как изолировать данные

- ✓ Хранилище данных и стейта внутри модуля

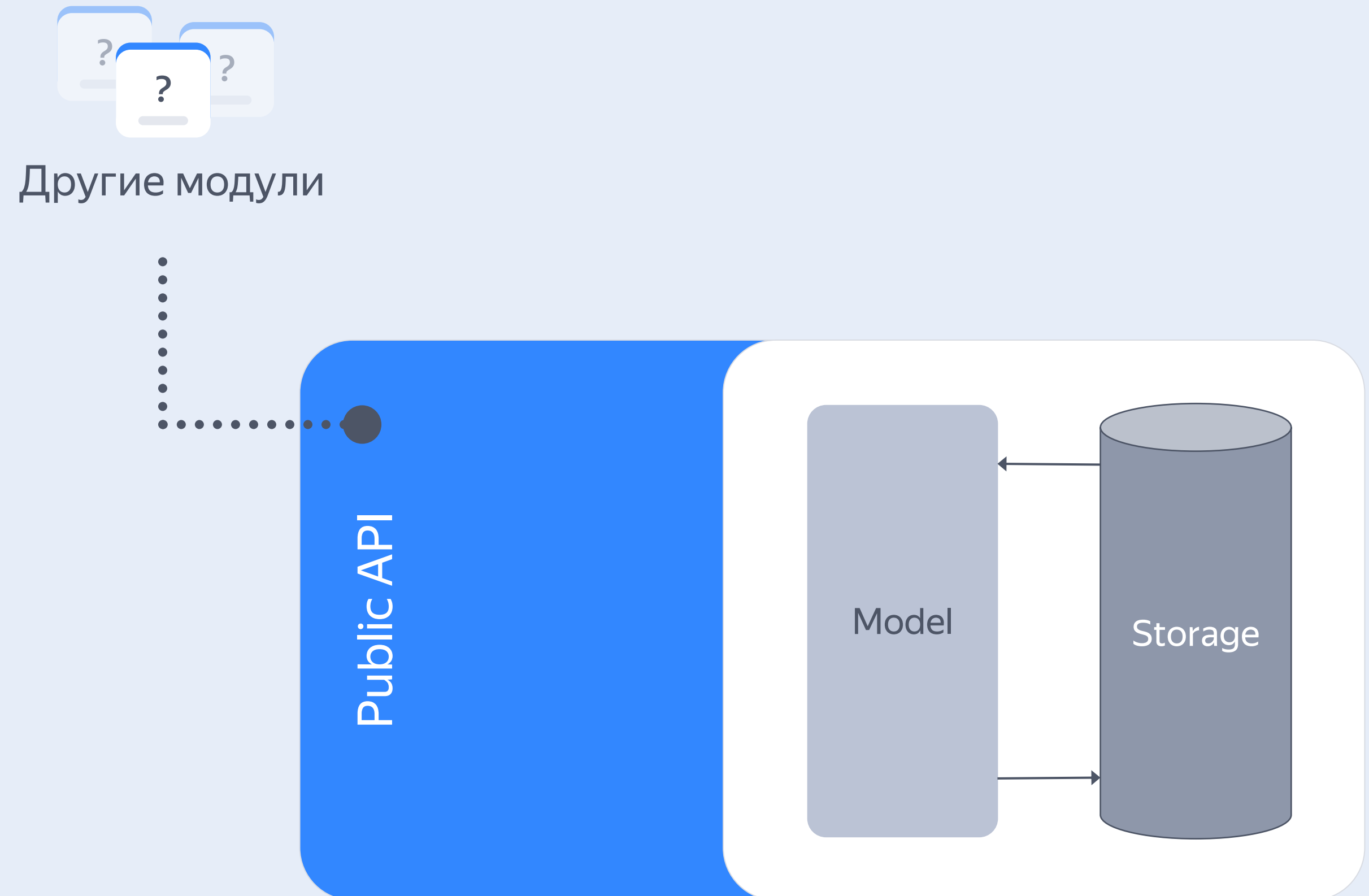


Другие модули



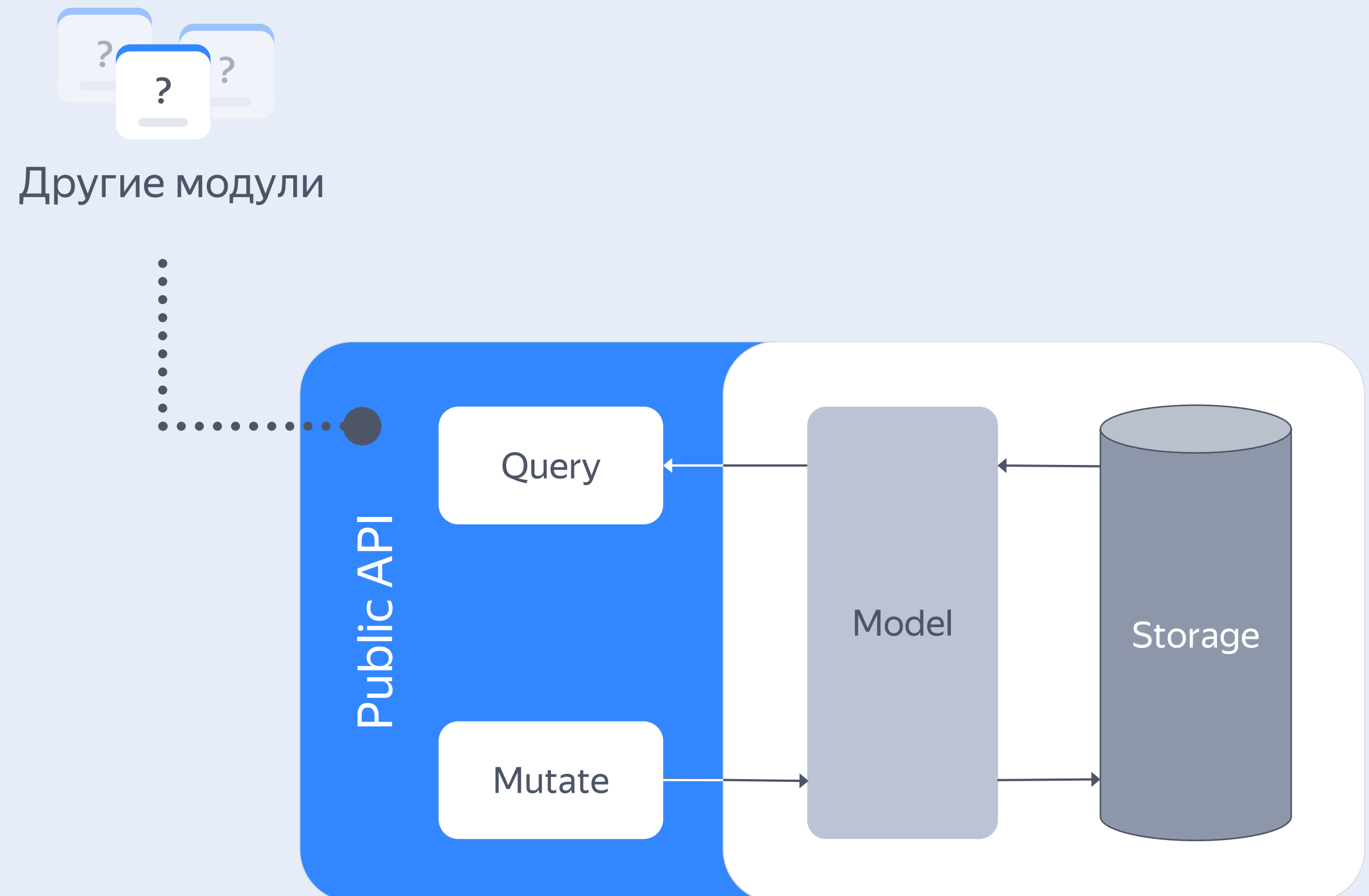
Как изолировать данные

- ✓ Хранилище данных и стейта внутри модуля
- ✓ Доступ только через Public API



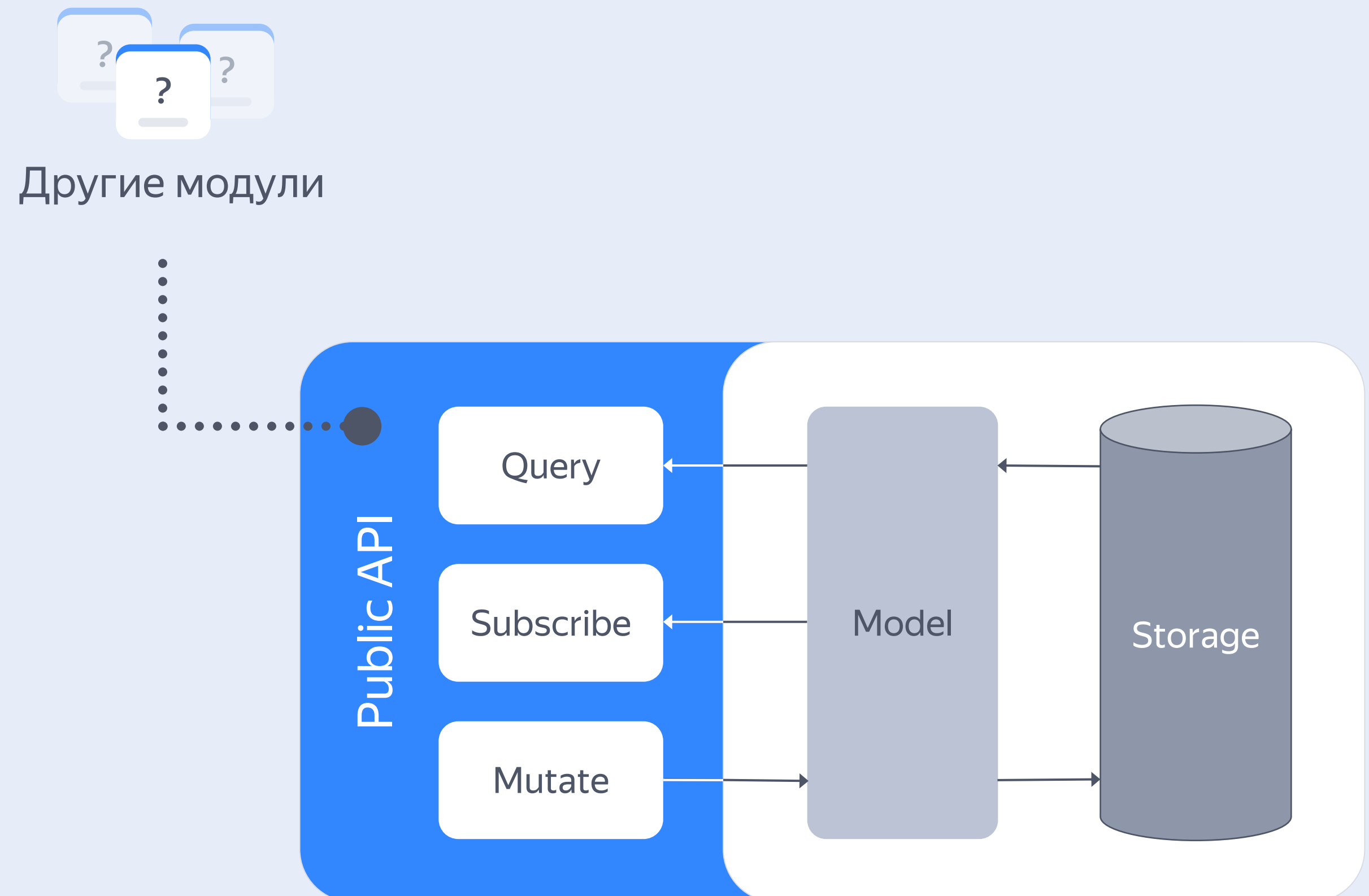
Как изолировать данные

- ✓ Хранилище данных и стейта внутри модуля
- ✓ Доступ только через Public API
- ✓ Команды для получения и мутации данных (CQRS)



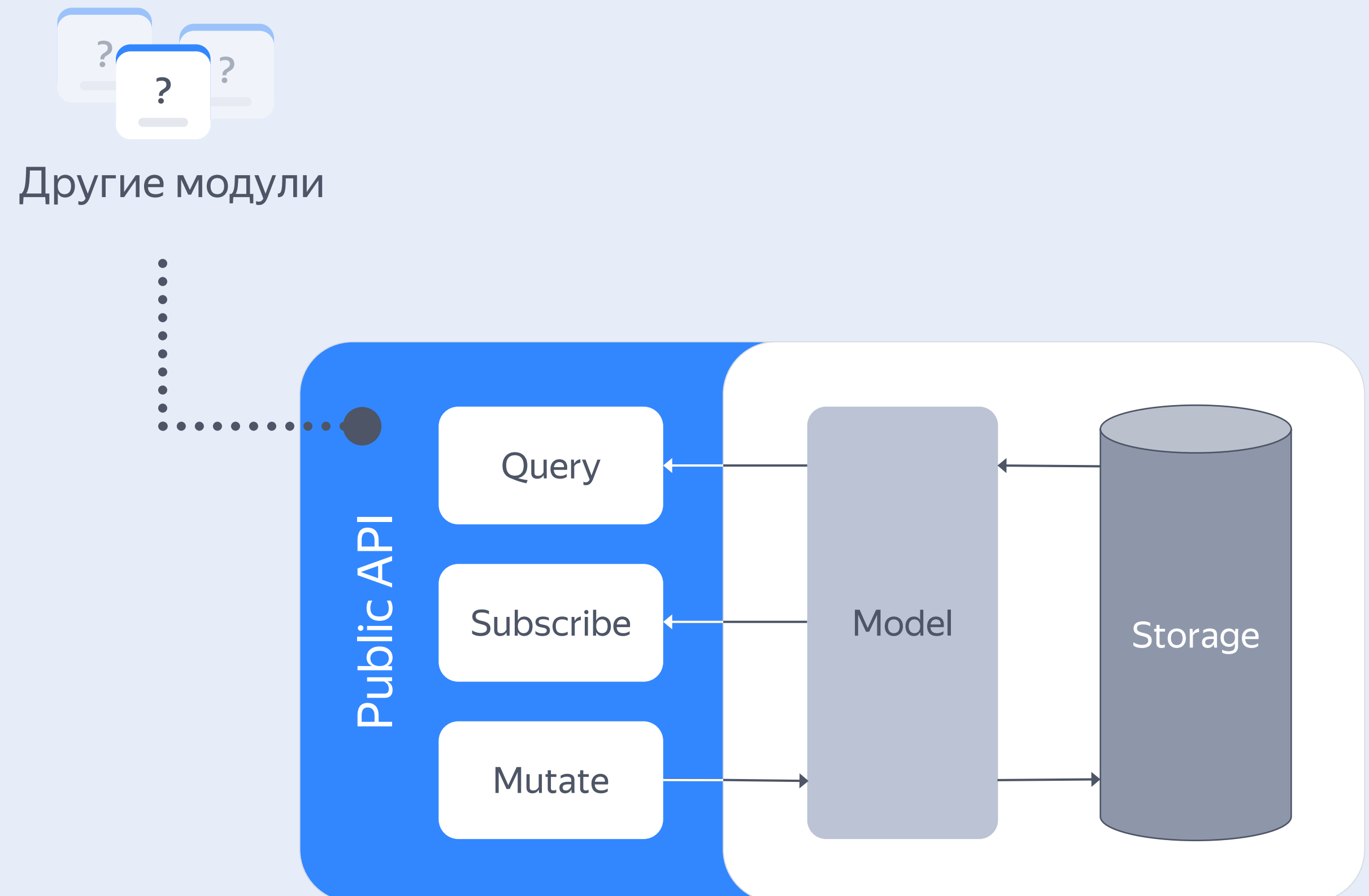
Как изолировать данные

- ✓ Хранилище данных и стейта внутри модуля
- ✓ Доступ только через Public API
- ✓ Команды для получения и мутации данных (CQRS)
- ✓ Подписка на изменение данных



Как изолировать данные

- ✓ Хранилище данных и стейта внутри модуля
- ✓ Доступ только через Public API
- ✓ Команды для получения и мутации данных (CQRS)
- ✓ Подписка на изменение данных
- ✓ Защита от мутаций: TS Readonly, Object.freeze



Путь к Glorious Monolith

01

Модуль

- Структура
- Изоляция
- Public API



02

Runtime для модулей

- Окружение запуска
- Доступные зависимости
- Готовые решения



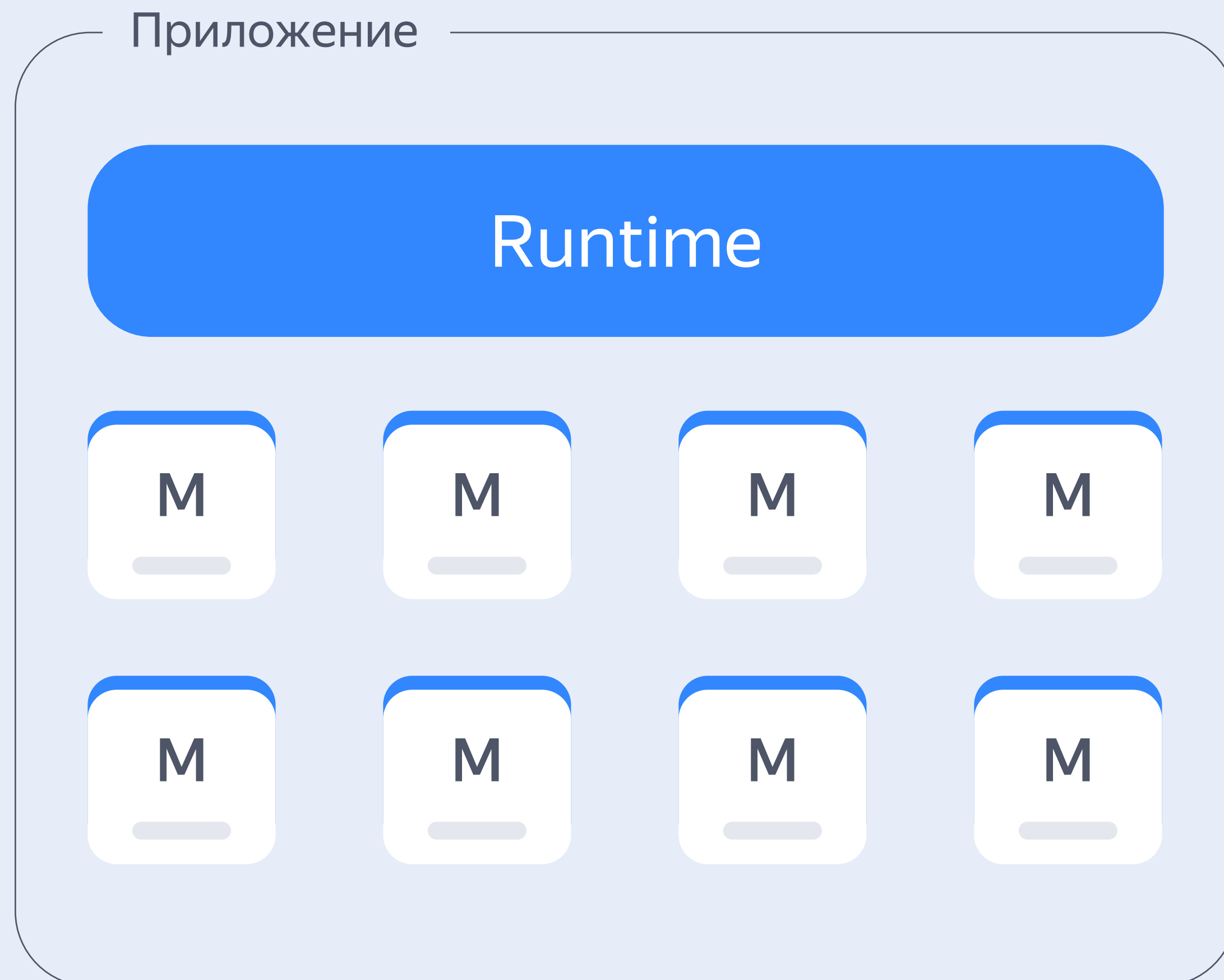
03

Границы между модулями



Runtime для модулей

- ✓ Один рантайм на все модули
- ✓ Гарантии по возможностям сборки
- ✓ Общие зависимости
- ✓ Способы межмодульной коммуникации
- ✓ ...



Пример Runtime

TypeScript

React

Стейт менеджер

HTTP клиент

Роутер

Feature Flags

Библиотека UI компонентов

CSS переменные дизайн-системы

Версия node.js для SSR

Версии целевых браузеров

Сборка и переменные окружения

Утилиты для обработки ошибок

Аналог web-vitals

Зависимости из package.json

Размер Runtime

Скромный runtime

Богатый runtime

✓ Больше ограничений

✗ Сложнее разработка модулей

✓ Унификация стека

✗ Сложнее поддержка

Путь к Glorious Monolith

01

Модуль

- Структура
- Изоляция
- Public API



02

Runtime для модулей

- Окружение запуска
- Доступные зависимости
- Готовые решения



03

Границы между модулями

- Зоны ответственности
- Ограничение зависимостей
- High cohesion



Как делить код на модули

Single Responsibility

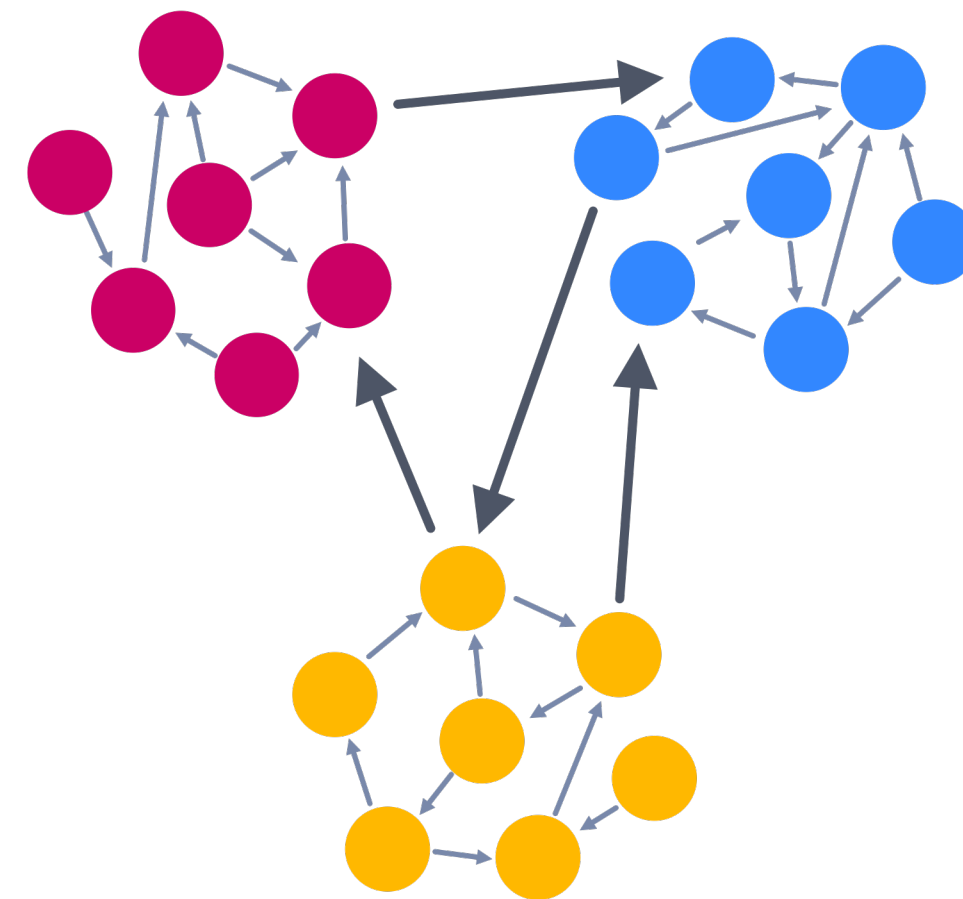
- Одна понятная зона ответственности
- Привязка к доменной области
- Или к конкретной технической задаче

Как делить код на модули

Single Responsibility

- Одна понятная зона ответственности
- Привязка к доменной области
- Или к конкретной технической задаче

High Cohesion & Low Coupling

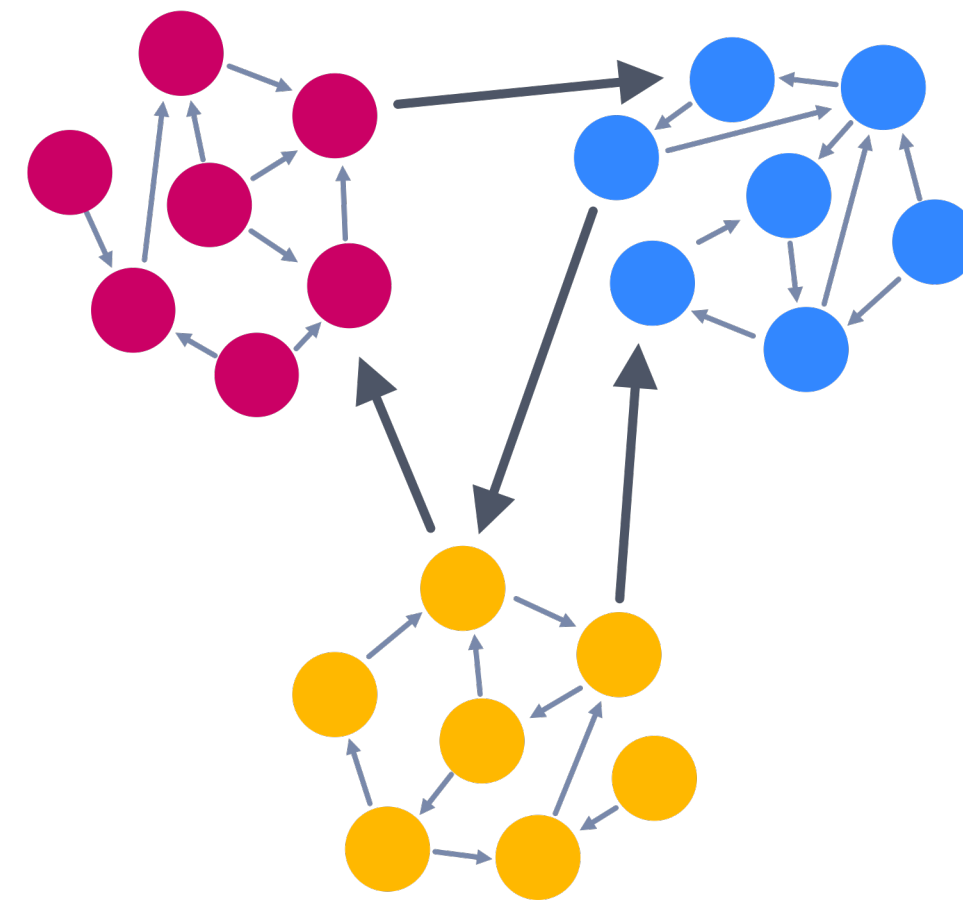


Как делить код на модули

Single Responsibility

- Одна понятная зона ответственности
- Привязка к доменной области
- Или к конкретной технической задаче

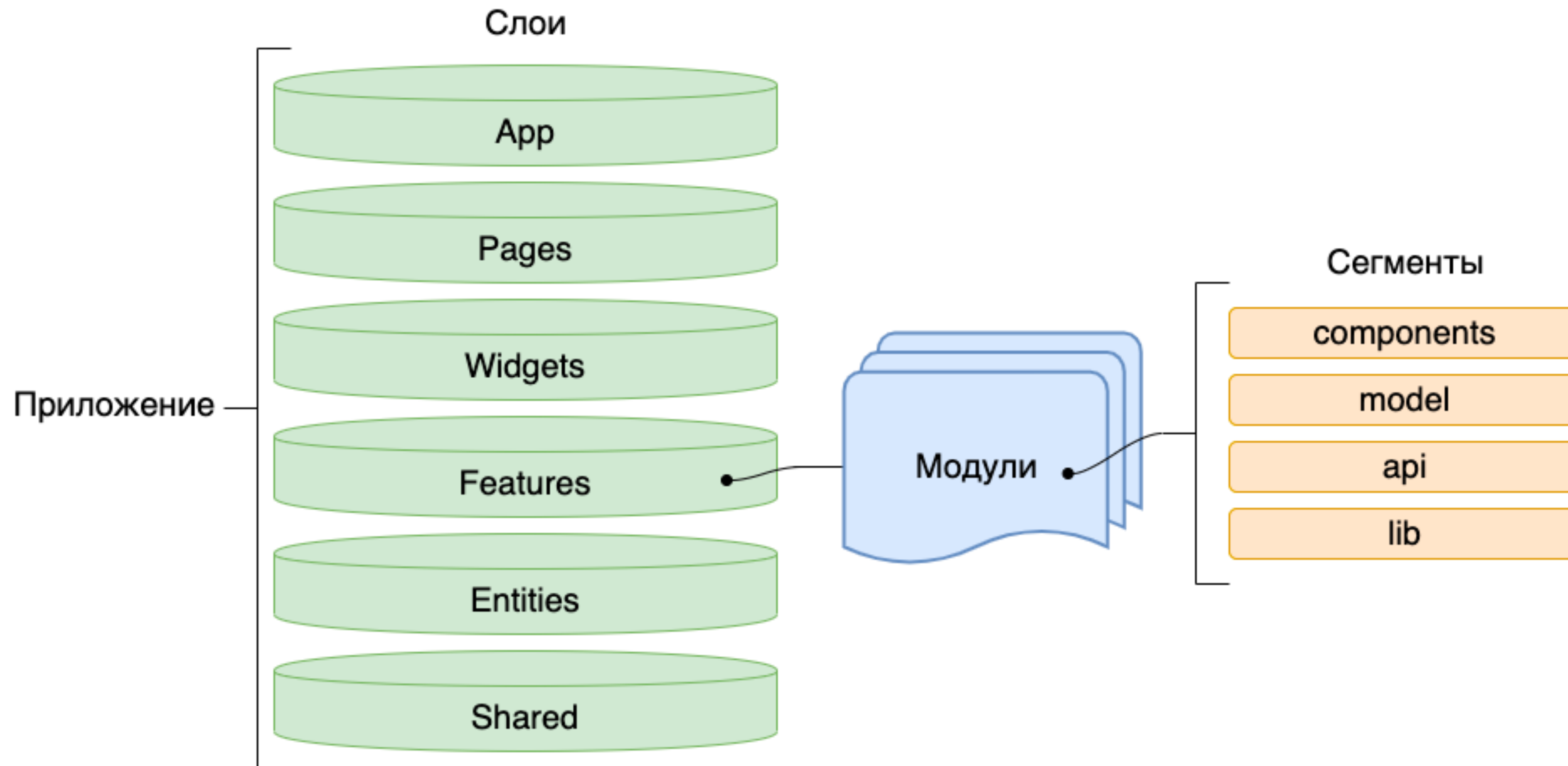
High Cohesion & Low Coupling



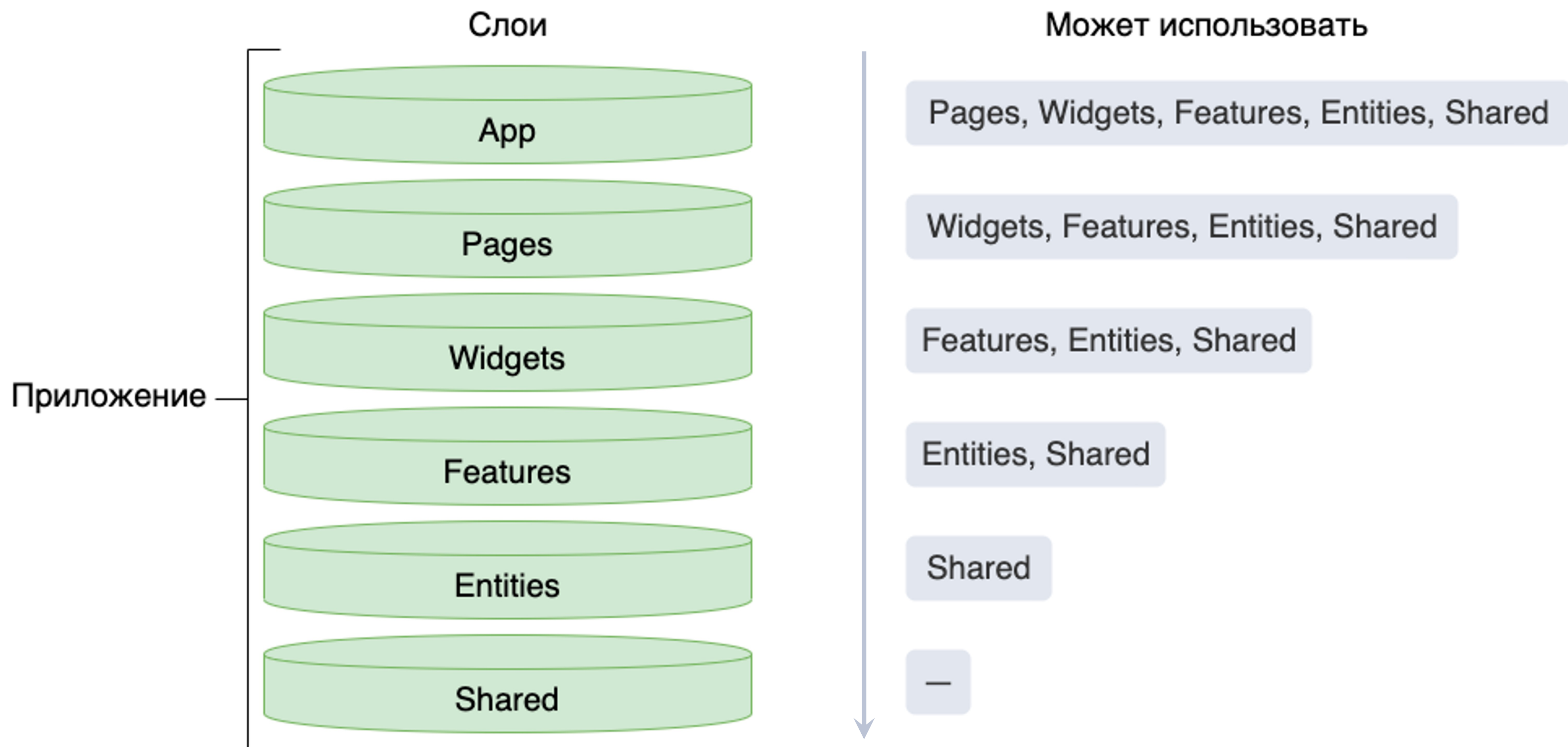
Ограничение зависимостей

- Деление модулей на смысловые группы
- Правила импортов между модулями из разных групп

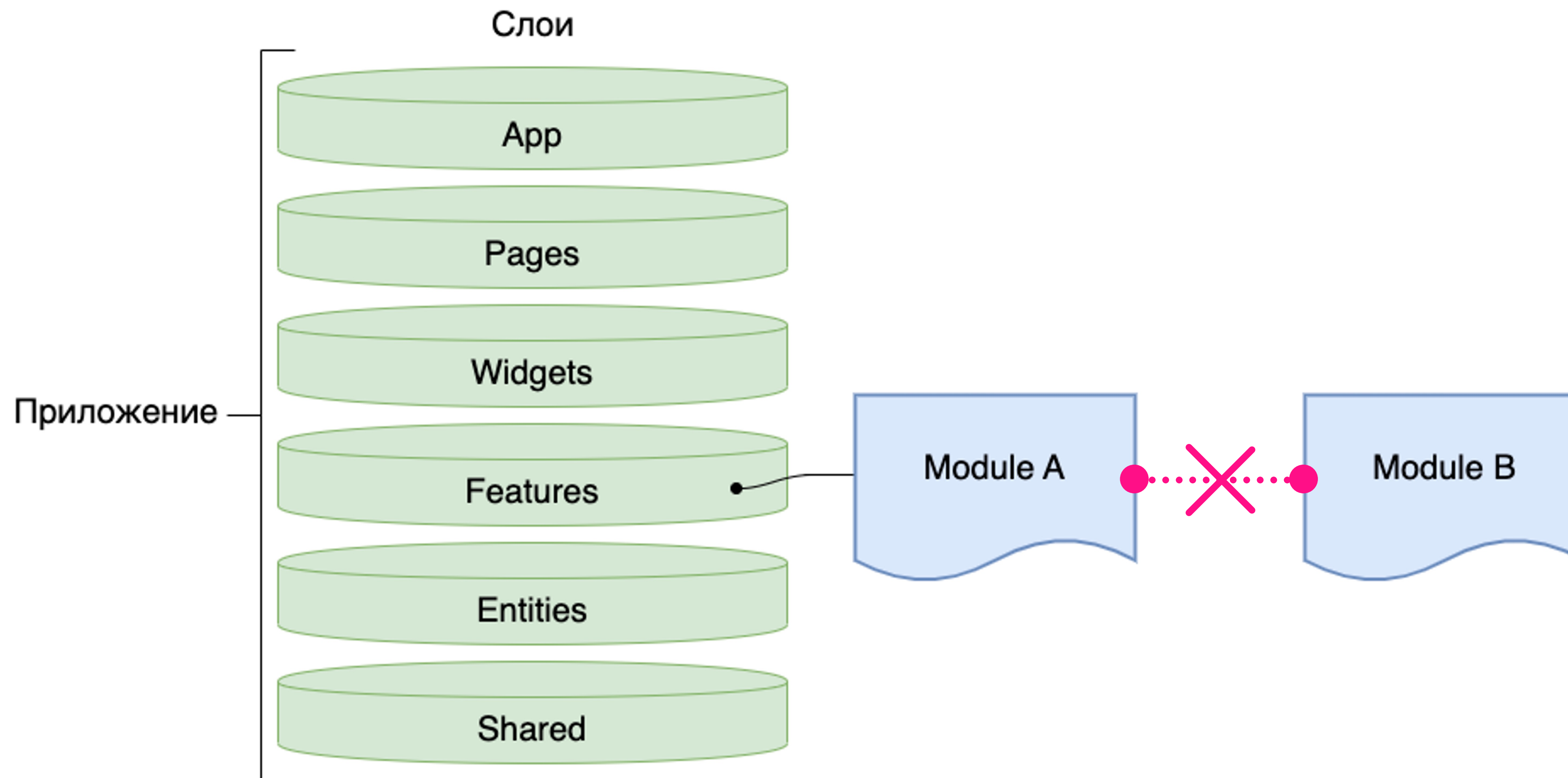
Feature Sliced Design



Направление импортов



Запрет импортов внутри слоя



Подводим итоги

Собираем все вместе

01

Модуль

- Структура
- Изоляция
- Public API
- Линтеры



02

Runtime для модулей

- Окружение запуска
- Доступные зависимости
- Готовые решения
- Линтеры



03

Границы между модулями

- Зоны ответственности
- Ограничение зависимостей
- High cohesion
- Линтеры



В результате

Плюсы Glorious Monolith

- ✓ Контроль сложности кодовой базы
- ✓ Хороший Developer Experience
- ✓ Относительно простая реализация
- ✓ Подходит для проектов любого размера
- ✓ Подходит для старта нового проекта

Основные статьи расходов

- Порог входа
- Настройка линтеров
- Скорость инструментов разработки *

* для очень больших проектов

- Нужно ли внедрять микрофронтенд?
- Скорее всего, нет

Что дальше

Теория

- Modular Monolith
- Domain Driven Design
- Clean Architecture
- Feature Sliced Design
- CQRS

Инструменты

- TypeScript
- eslint + custom rules
- eslint-plugin-boundaries
- eslint-plugin-import
- stylelint + custom rules
- dependency-cruiser

Ⓜ Реклама

Спасибо за внимание!

Максим Земсков
Ведущий разработчик интерфейсов

nodge@yandex-team.ru

t.me/nodge74

linkedin.com/in/maxim-zemskov

