

# **Монолит и его рефакторинг.**

**Проектирование монолита.  
Первые предпосылки  
перехода к микросервисам**

**Чернухин Максим**

Software Architect

АО Альфа-Банк

# В прошлом модуле

- Что такое нефункциональные требования (НФТ)?
- Как их выделить и разбить на категории?
- Атрибуты качества и как они помогают построить архитектурное видение
- Способы проверки реализации нефункциональных требований

# В этом модуле

- Узнаем, как построить монолит
- Поймём, как безболезненно перейти от монолита к микросервисам
- Научимся строить интеграции с монолитом

# На этом уроке мы узнаем

- Как уменьшить стоимость перехода от монолита к микросервисам?
- Что служит основными предпосылками для перехода на микросервисы?

# Что же лучше?

Монолит

Микросервисы

Любой тип  
распределённой  
архитектуры

# Когда лучше?

Монолит

Микросервисы

Любой тип  
распределённой  
архитектуры

# Всему есть цена



# Как посчитать цену монолита?

- Поддержка текущего решения
- Время на разработку нового функционала
- Добавочное время к тестированию, если монолит разрастается
- Сложность работы над ним нескольких команд
- Сложность масштабирования
- Сложность оптимизаций
- Сложность отказа от монолита\*

# Как посчитать цену микросервисов?

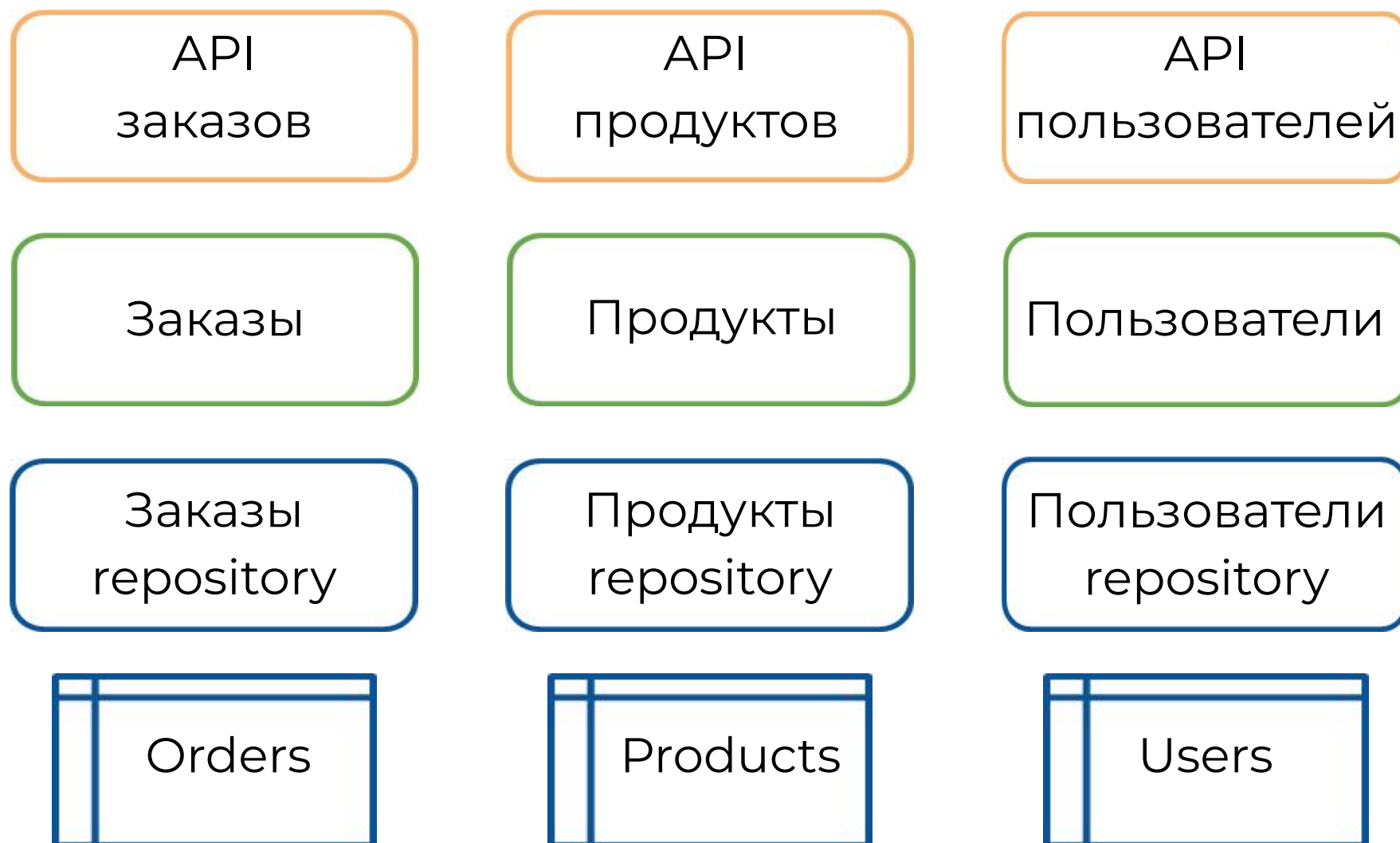
- Поддержка инфраструктуры
- Работа с распределёнными транзакциями
- Сложность логирования
- Сложность мониторинга
- Сложность версионирования и соблюдения контрактов
- Интеграции между микросервисами
- PipeLine по установке новых версий



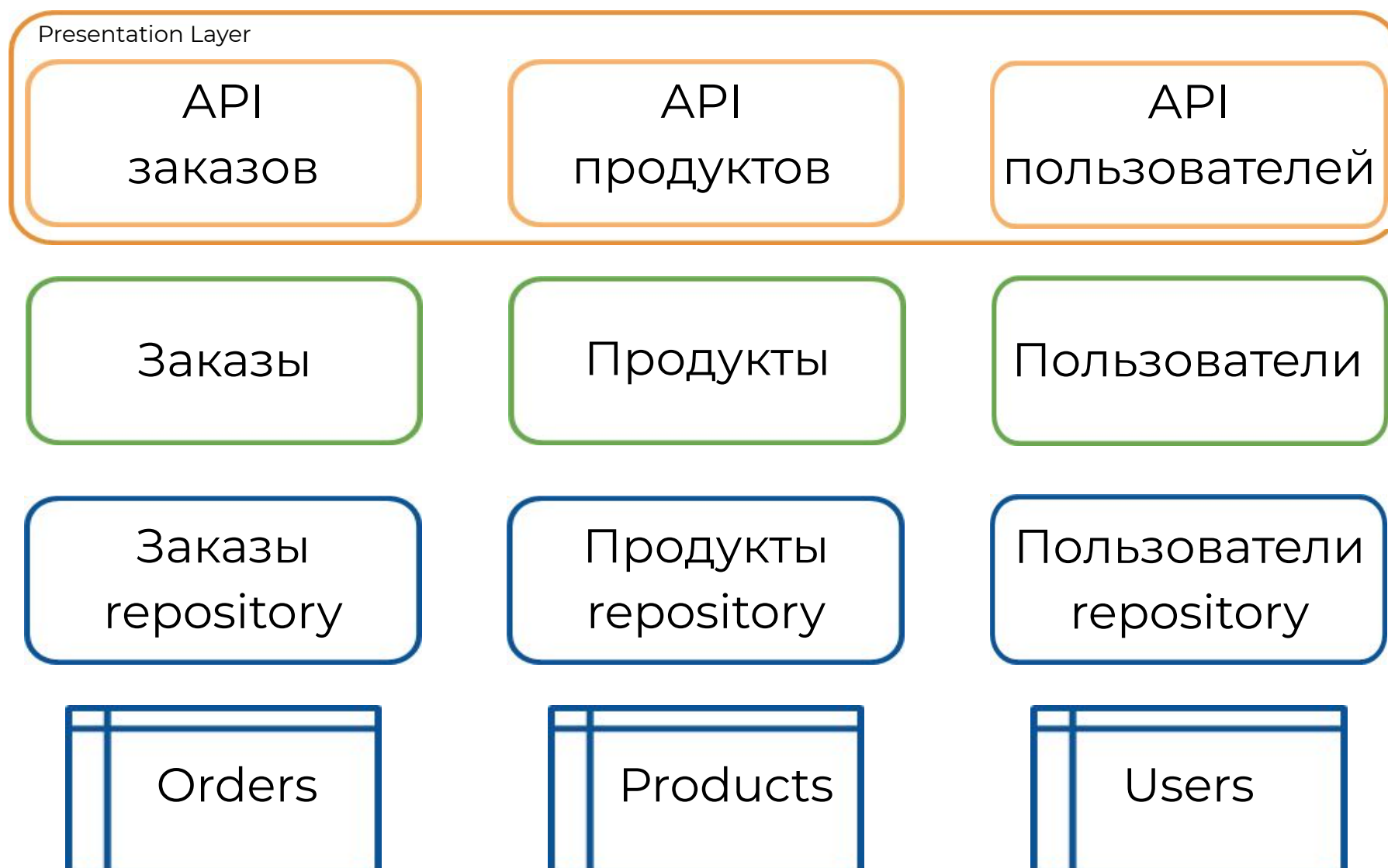
# Выбор



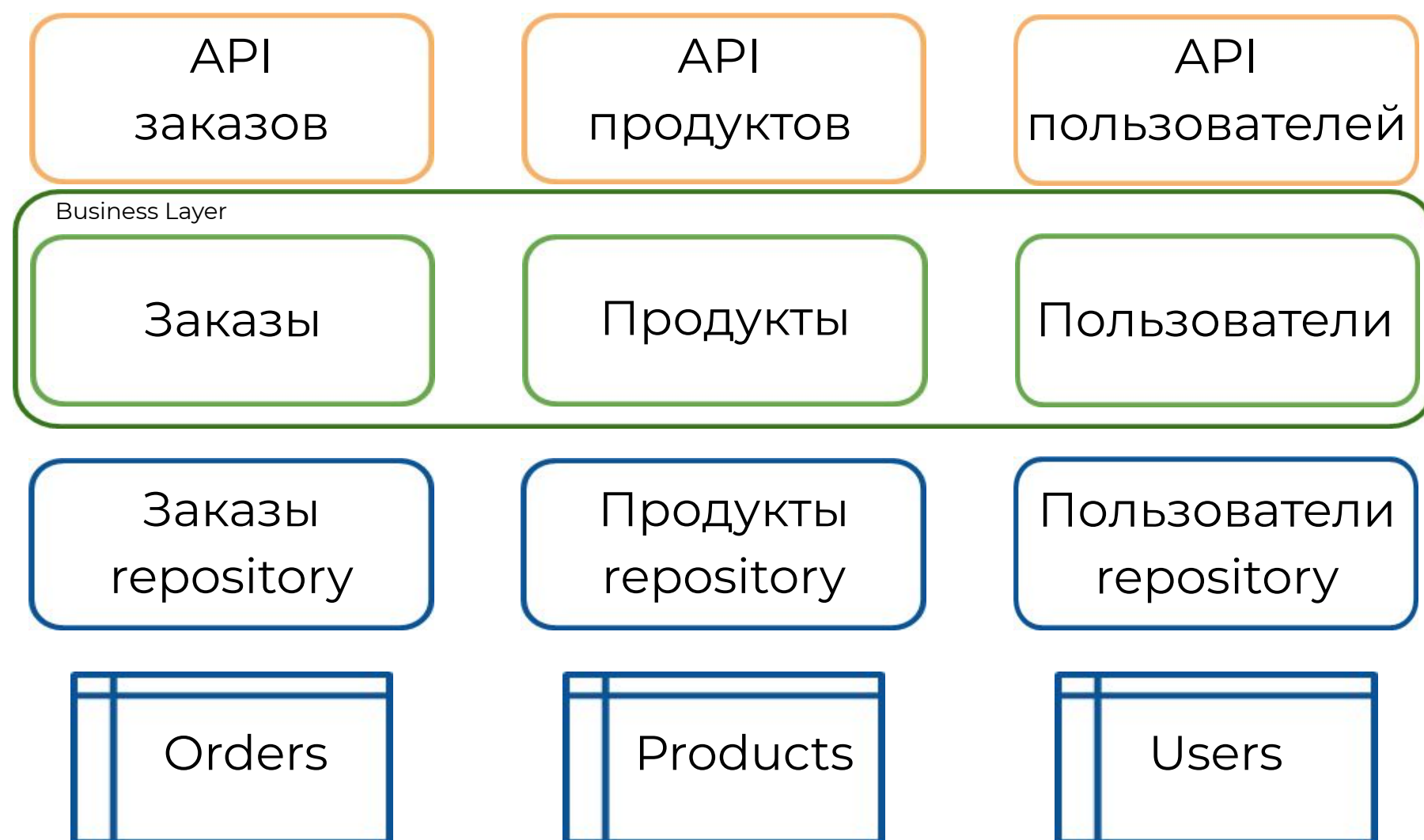
# Уменьшаем стоимость ухода от монолита



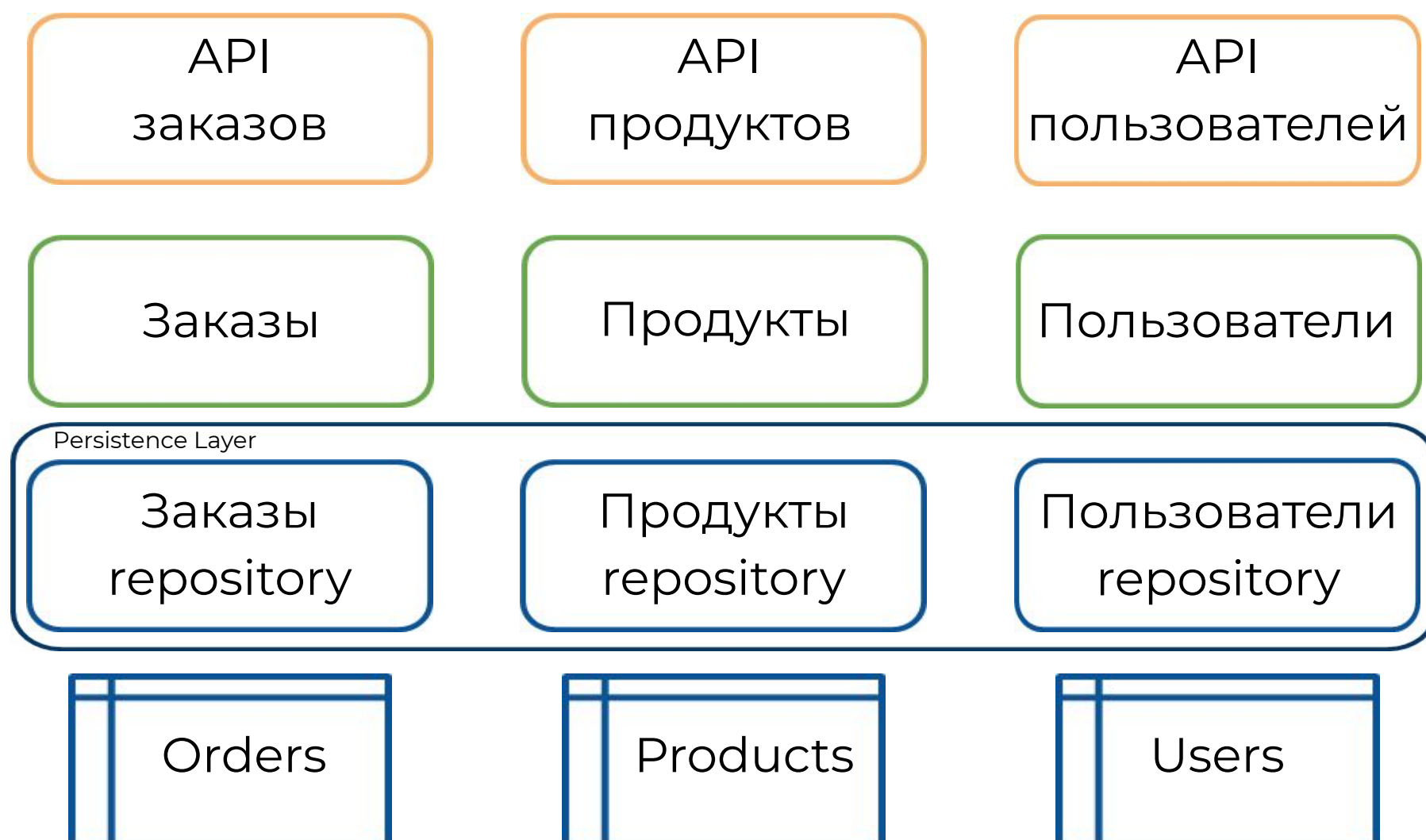
# Проектирование монолита



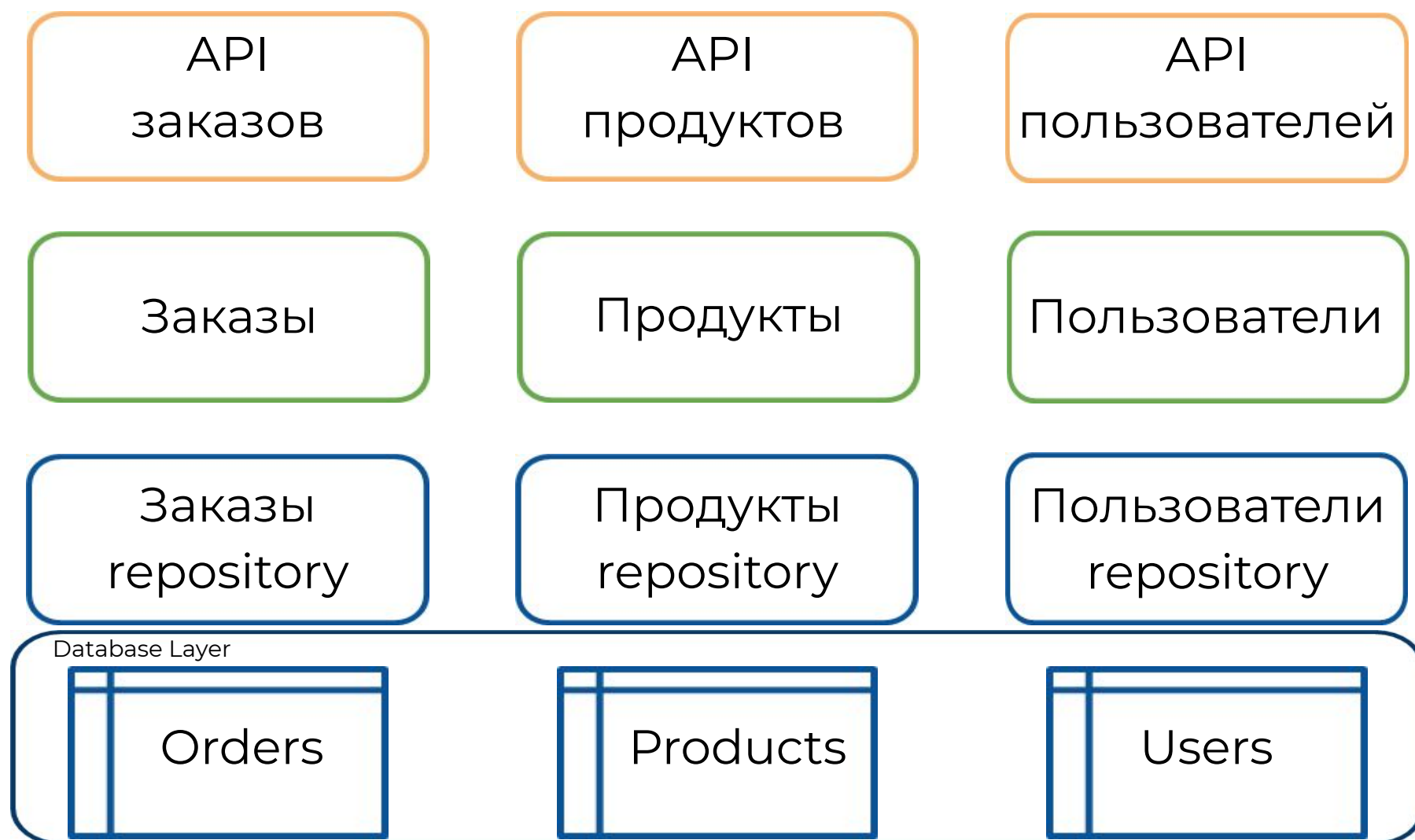
# Проектирование монолита



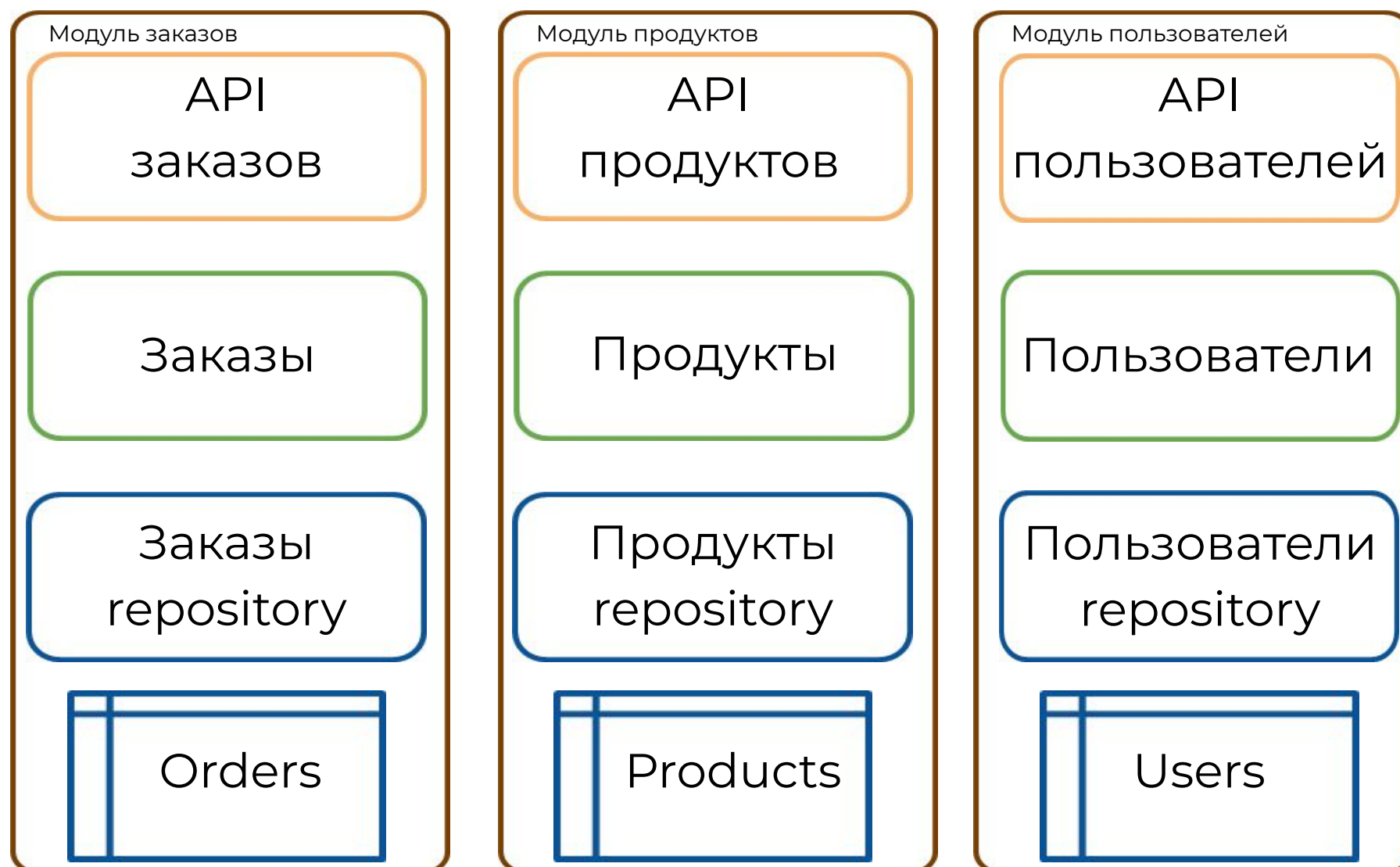
# Проектирование монолита



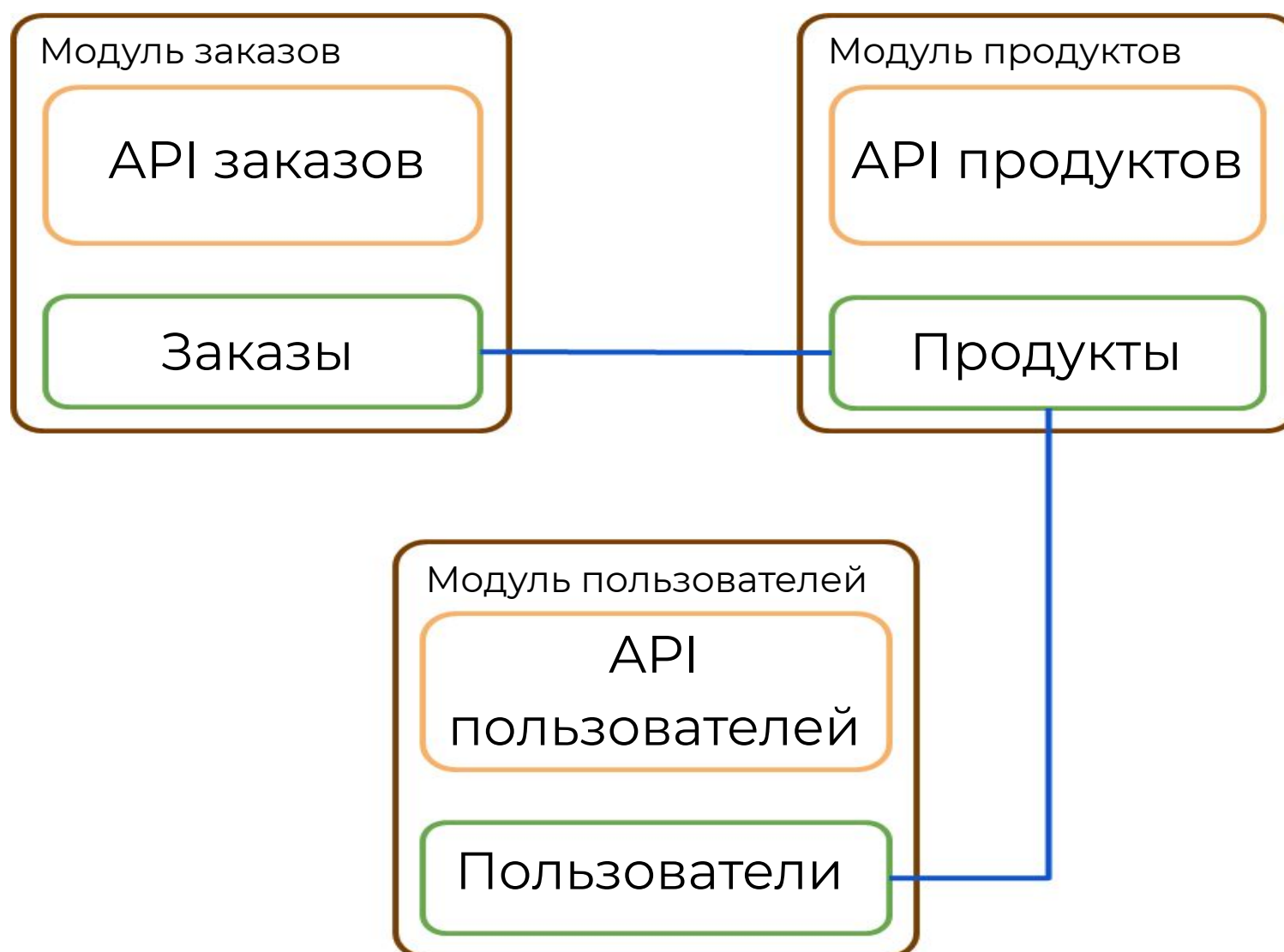
# Проектирование монолита



# Проектирование монолита

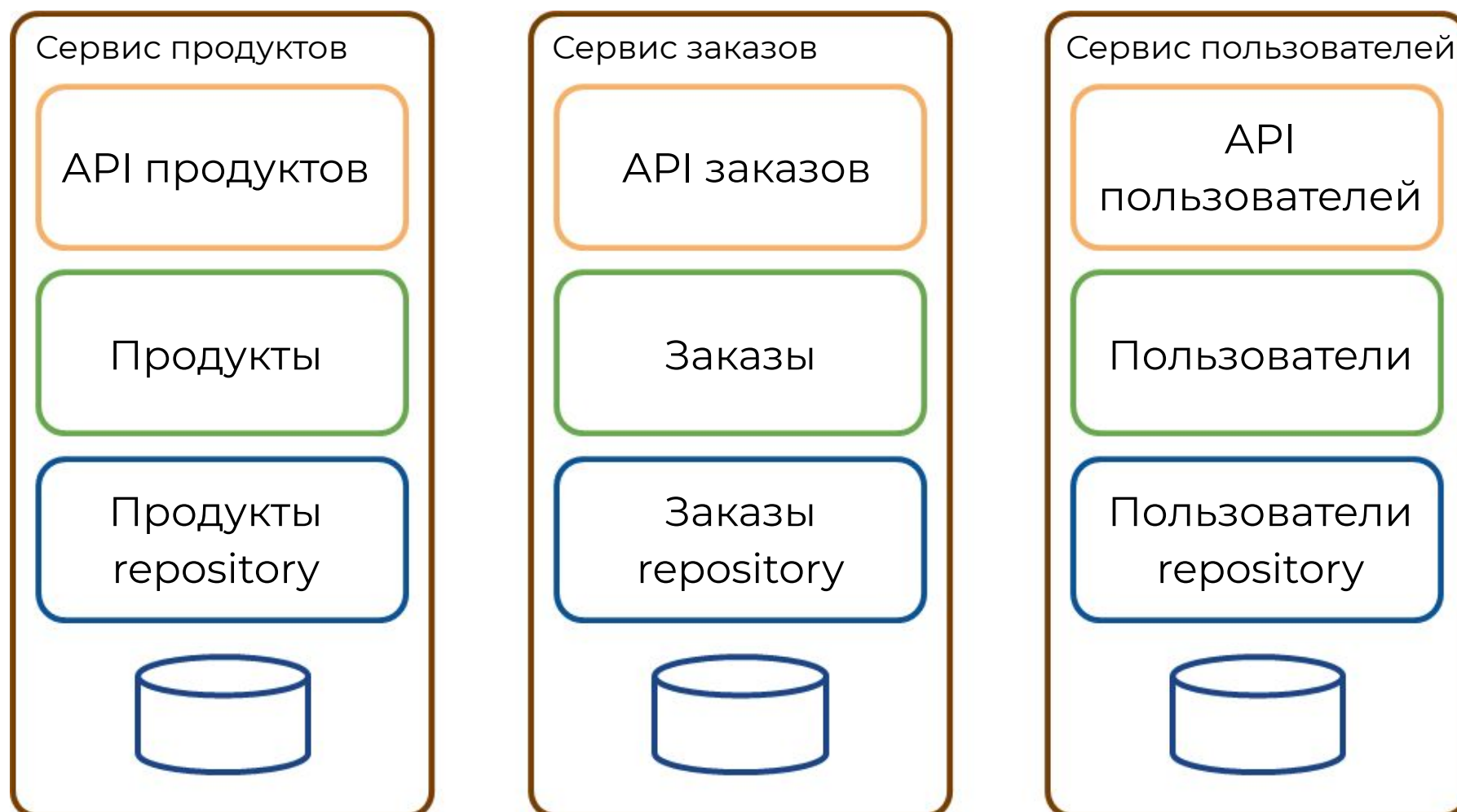


# СВЯЗИ В МОНОЛИТЕ

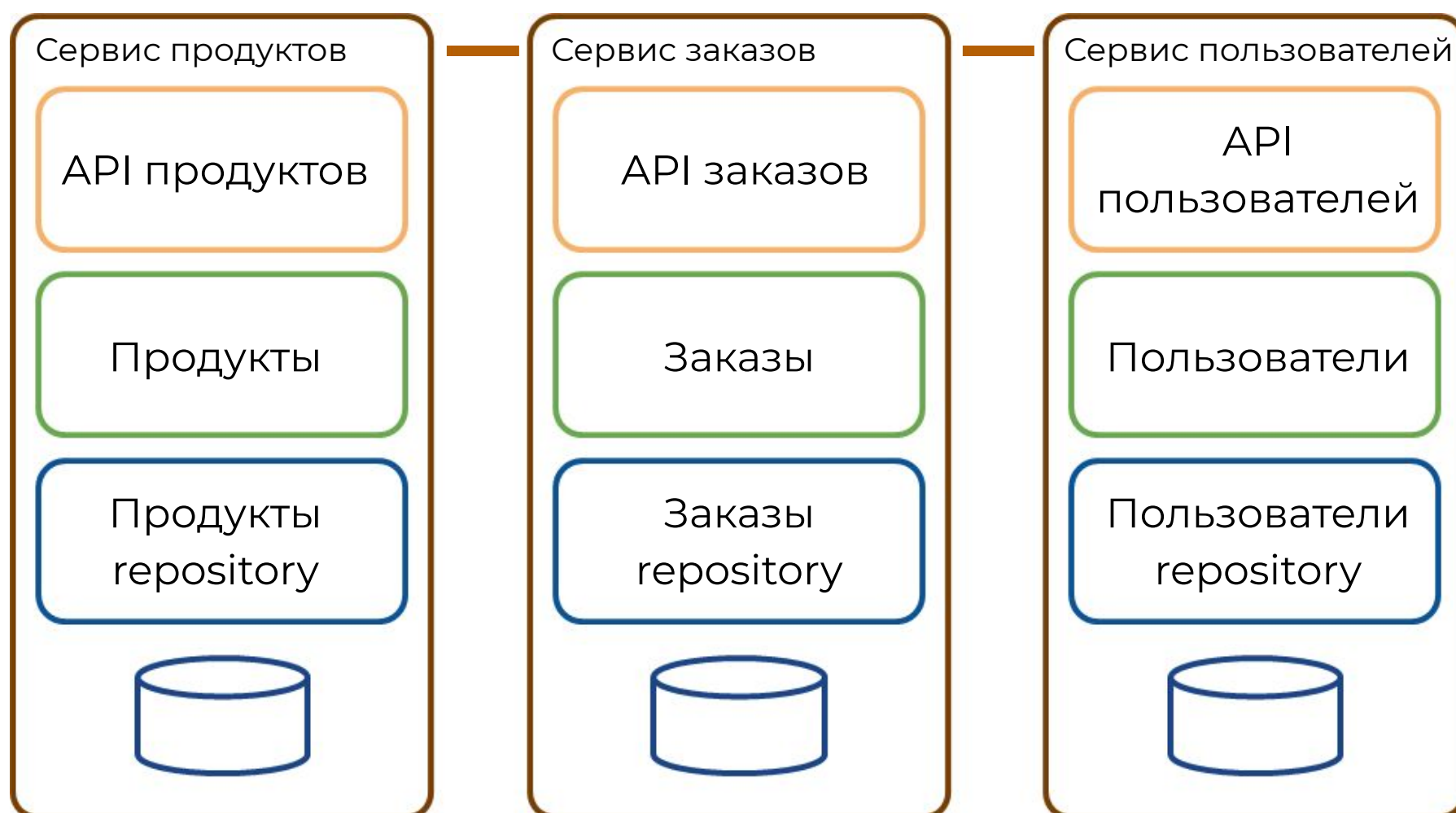




# Разделение на сервисы



# Разделение на сервисы



# Как же понять, когда делить монолит?

- Необходимо увеличить скорость разработки
- Увеличилось число команд
- Сложная и долгая процедура тестирования
- Появляются отдельные высоконагруженные модули
- Дорого покупать более мощное железо
- Вся логика приложения не помещается в голове одного разработчика\*

# Процесс разработки

- Делится на 4 этапа
- Тестирование выделено как один общий этап



# Проектирование

- Увеличилось количество документации
- Сложные взаимосвязи
- Много связей с одним модулем

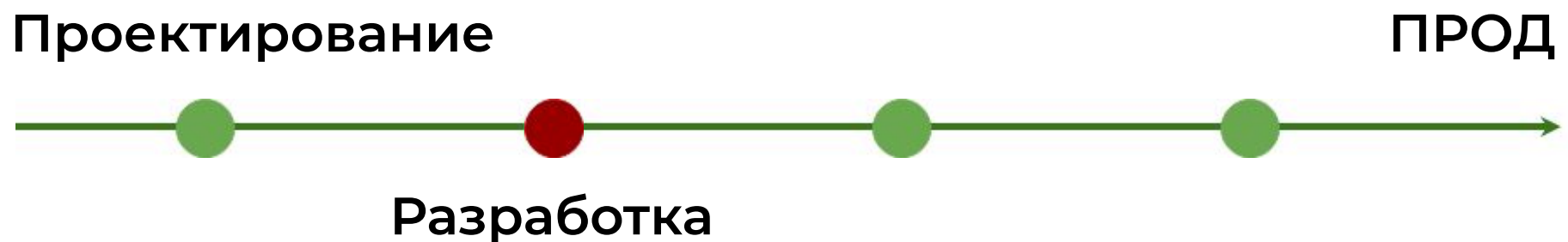
Проектирование

ПРОД



# Разработка

- Начал появляться дублирующий функционал
- Больше число конфликтов при работе с версионированием



# Тестирование

- Увеличилось время тестирования
- Увеличилось время на покрытие тестами
- Усложнились тесты
- Надо проверять действия на соседние модули



# Внедрение в ПРОД

- Время релиза увеличилось
- Увеличилась сложность проверок после релиза
- Вас больше не устраивает время простоя во время релиза





# Сопровождение

- Нагрузка на один модуль отражается на других
- Более мощных серверов или не существует или они очень дорогие
- Необходимость быстро и гибко масштабироваться

Сопровождение



# Выбор



# Итоги

- Начинать почти всегда лучше с монолита
- Хорошо спроектированный монолит — лёгкость перехода на микросервисы
- Разобрали этапы разработки и предпосылки для перехода на микросервисы

**Микросервисы**

**Монолит**



# Что дальше?

- Узнаем, как выделять модули в микросервисы
- Какие подходы выделения бывают?
- На что стоит обращать внимание?

**Спасибо  
за внимание!**