



Образование для инженеров
и технических лидеров

Проектирование микросервисов на примере разделения монолита

Спикер: Щербаков Пётр
Solution Architect

КУПЛЕНО НА
SKLADCHIK.COM



План урока

- 01** Что такое монолит
- 02** Слои приложения
- 03** Как выглядят монолиты в архитектуре
- 04** Основные проблемы монолитов
- 05** Когда полезен монолит
- 06** Анализ монолита



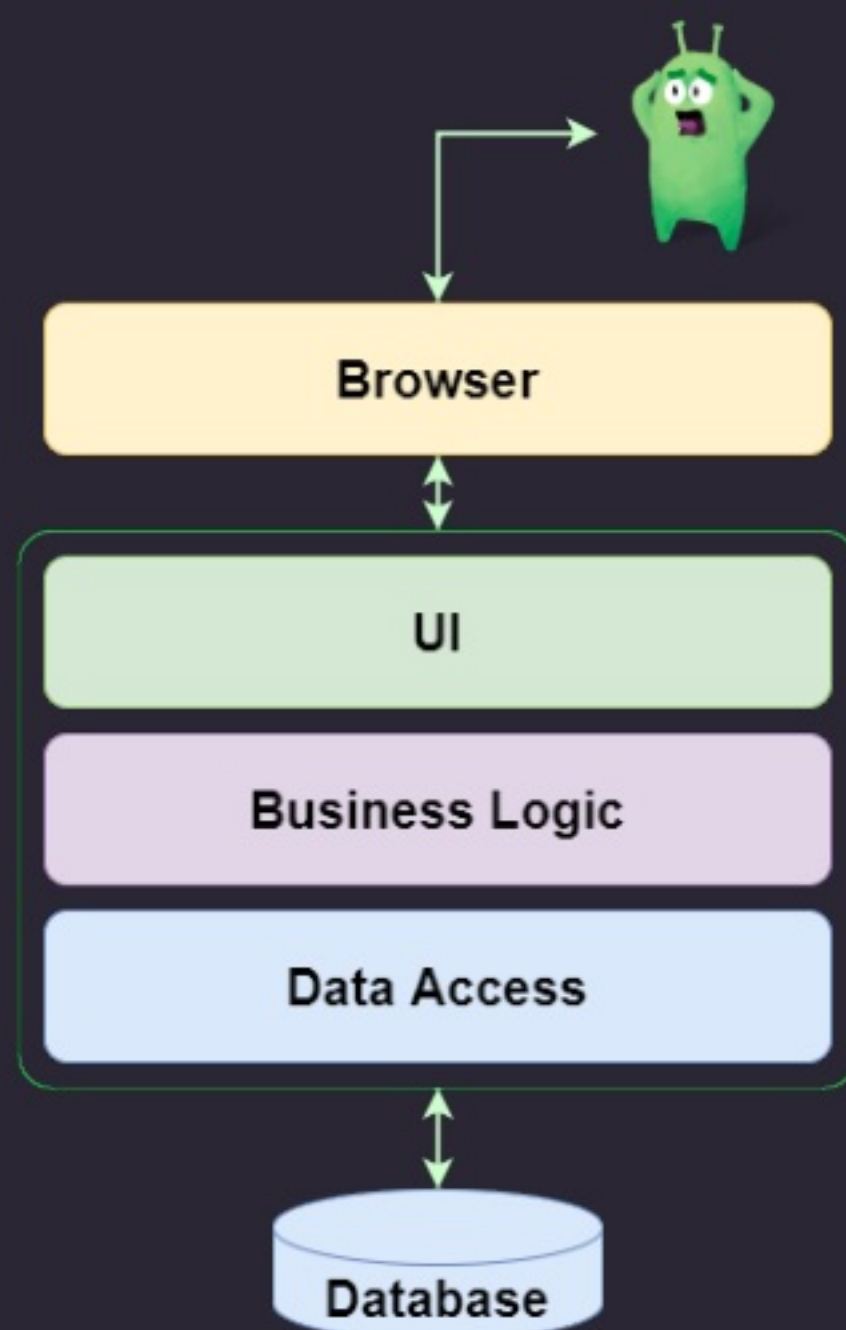
Понятие монолита

- Монолит – это одно приложение
- Монолит – это одно приложение созданное для уборщика, менеджера, поста охраны, администратора и гостя отеля
- Монолит – содержит в себе код пользовательского интерфейса и код доступа к данным и бизнес-логику
- Монолит – может быть многопоточным или однопоточным
- Монолит – может быть модульным, одним приложением или связанной группой приложений
- Монолит – можно засунуть в Docker-контейнер или VM, Bare metal
- Монолит – единая кодовая база для всего проекта, нельзя просто взять и выделить отдельный код и запустить параллельно от монолита
- Монолит – когда для обновления нужно сначала заглушить двигатель
- Монолит – это иногда одна БД для всех приложений.

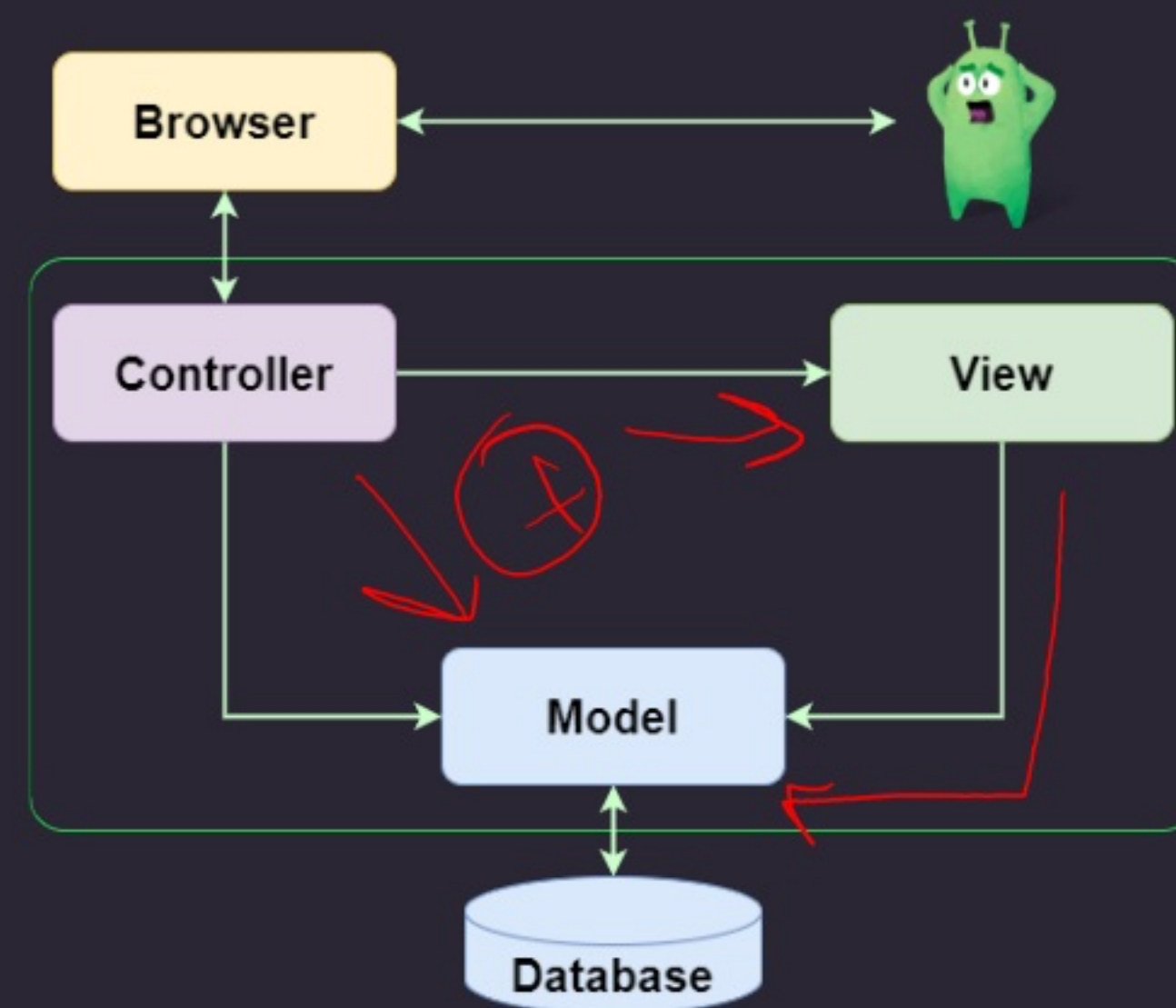


Слои приложения

- Пользовательский интерфейс (UI)
- Слои бизнес логики (Business Logic)
- Слои доступа к данным (Data Access)

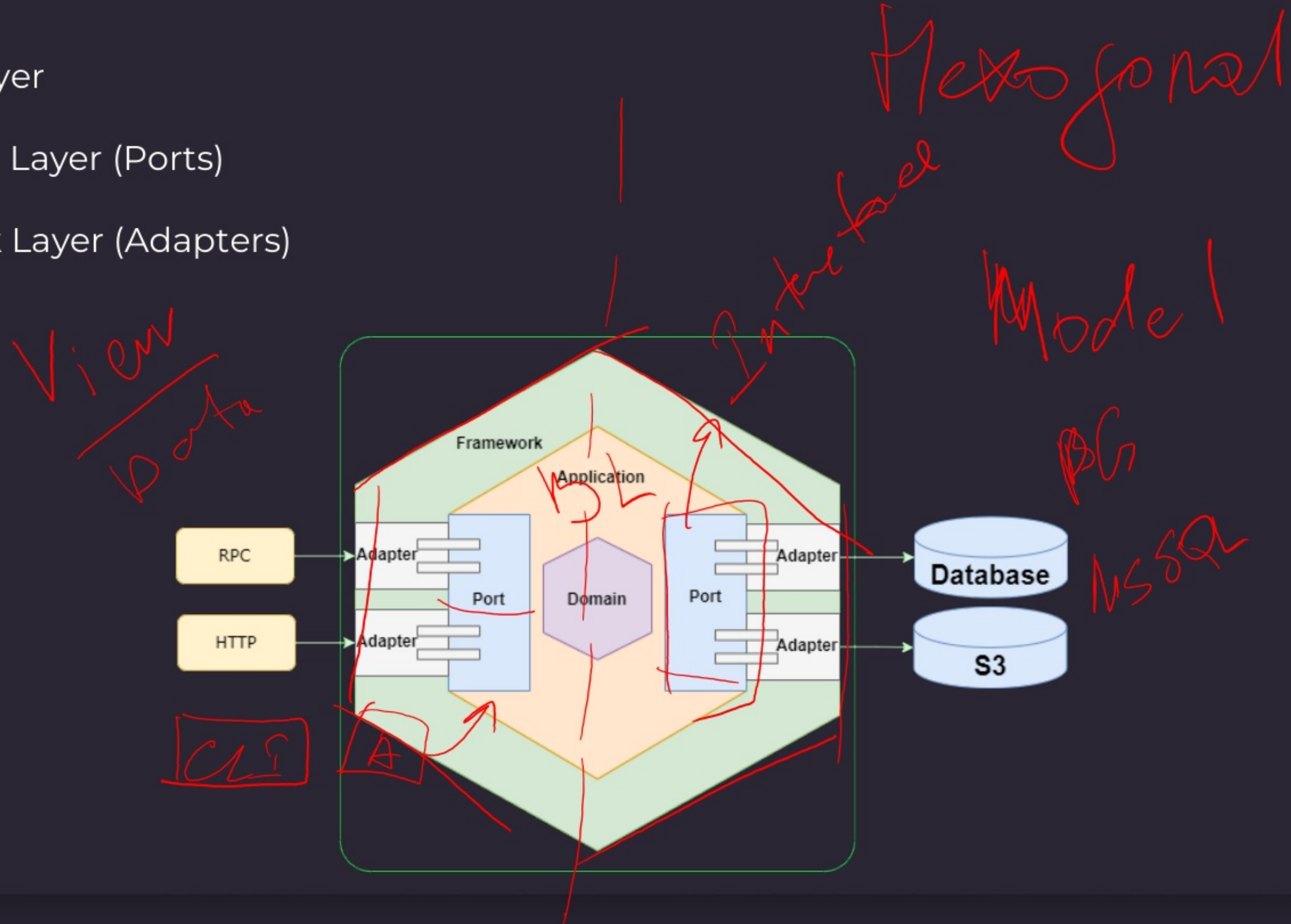


- View
- Controller
- Model



Слои приложения

- Domain Layer
- Application Layer (Ports)
- Framework Layer (Adapters)



Слои приложения

- Domain - Entities
- Use Cases – Use Cases
- Interface Adapters - Controller
- Framework & Drivers

Framework & Drivers Initialize

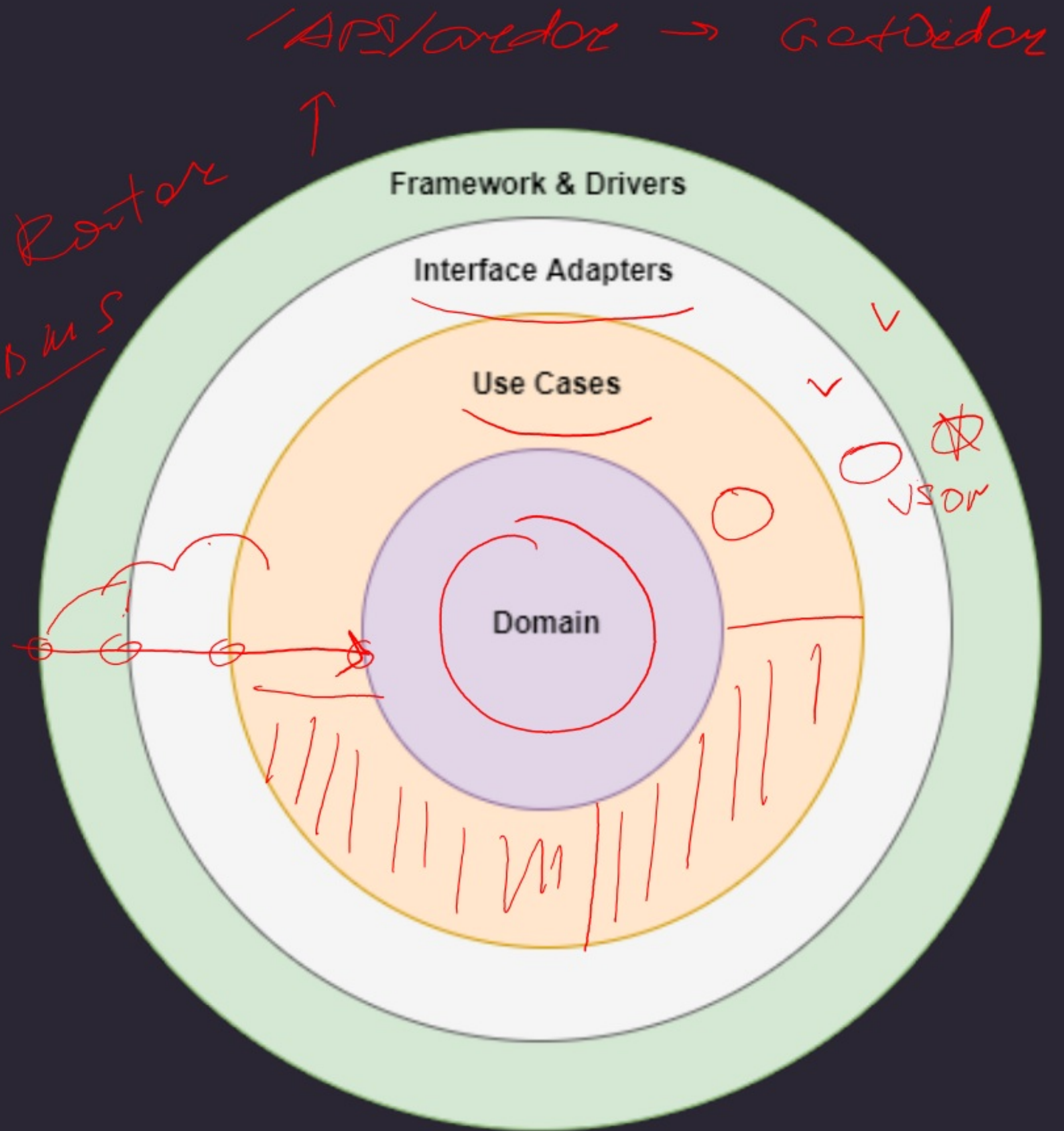
Infrastructure Adapter
Interface Adapter

Всё то что обеспечивает запуск приложения

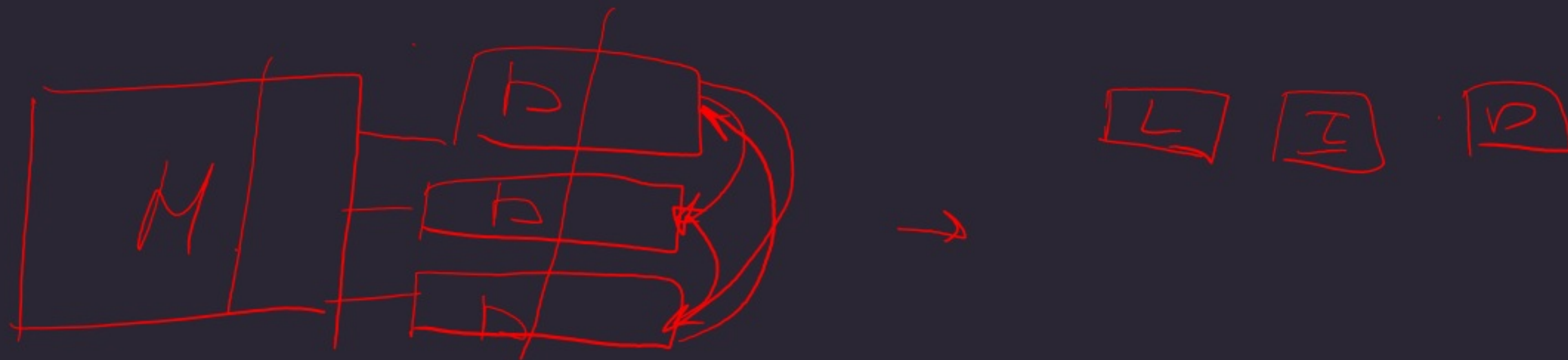
Interface Adapters Implements

Infrastructure Adapter
Interface Adapter

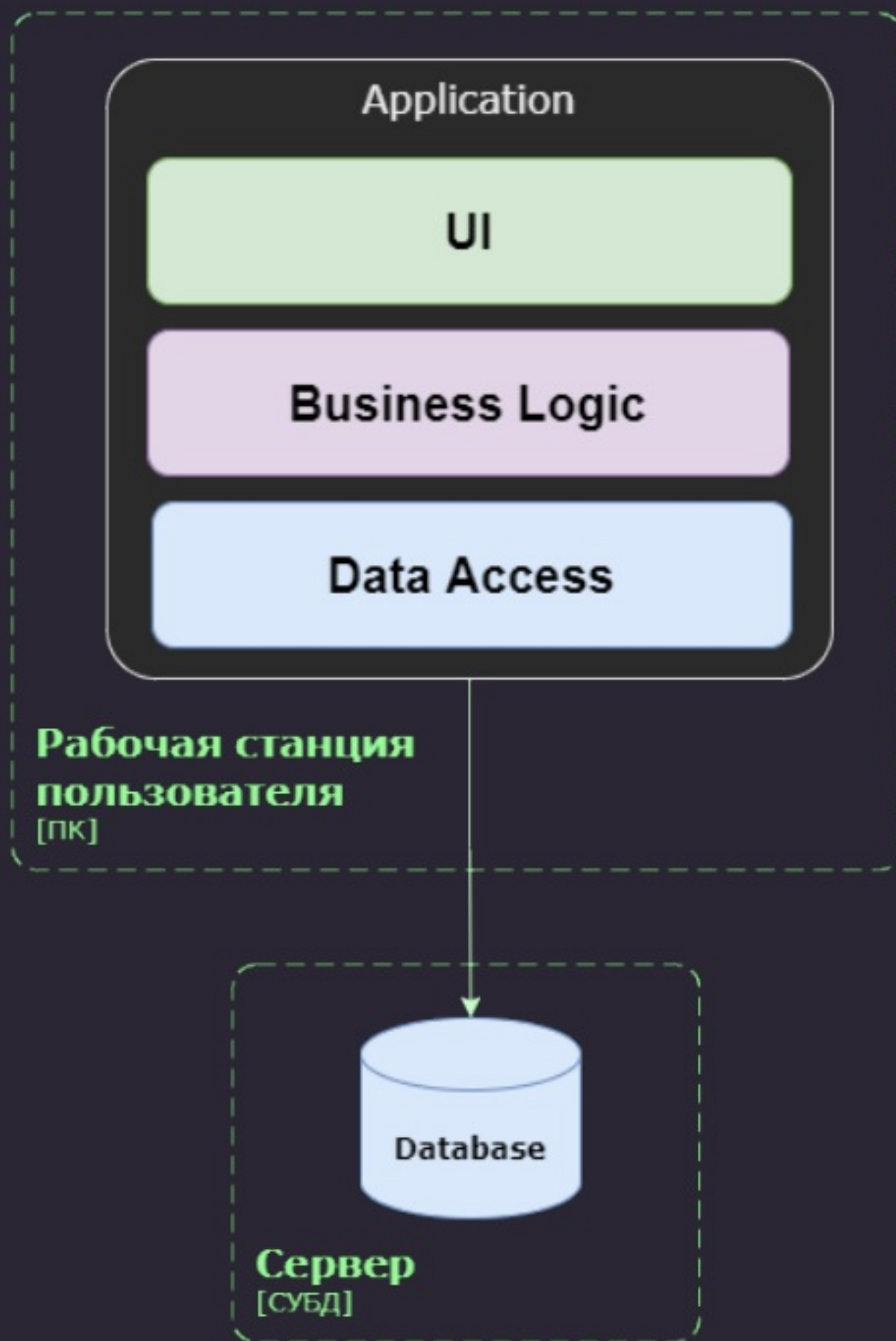
Всё то что обеспечивает выполнение функций приложения



Для рисования



Как выглядят монолиты в архитектуре



1. Находится на машине пользователя
2. Работает с удаленной базой
3. Бизнес логика покрывает множество доменов

0. Называется – **толстый клиент**



Как выглядят монолиты в архитектуре



АРМ – Автоматизированное рабочее место

АИС – автоматизированная информационная система



Как выглядят монолиты в архитектуре



3-х уровневая архитектура – архитектурная модель ПО, состоящая из трёх типов компонентов (уровней): клиентское приложение (с которыми работают пользователи), сервер приложений (с которым работает клиентское приложение) и сервер базы данных (с которым работает сервер приложений)

Тонкий клиент, CSR – client side rendering



Как выглядят монолиты в архитектуре



Выглядит как 3-х уровневая архитектура

Работает как Тонкий клиент

В результате SSR – server side rendering

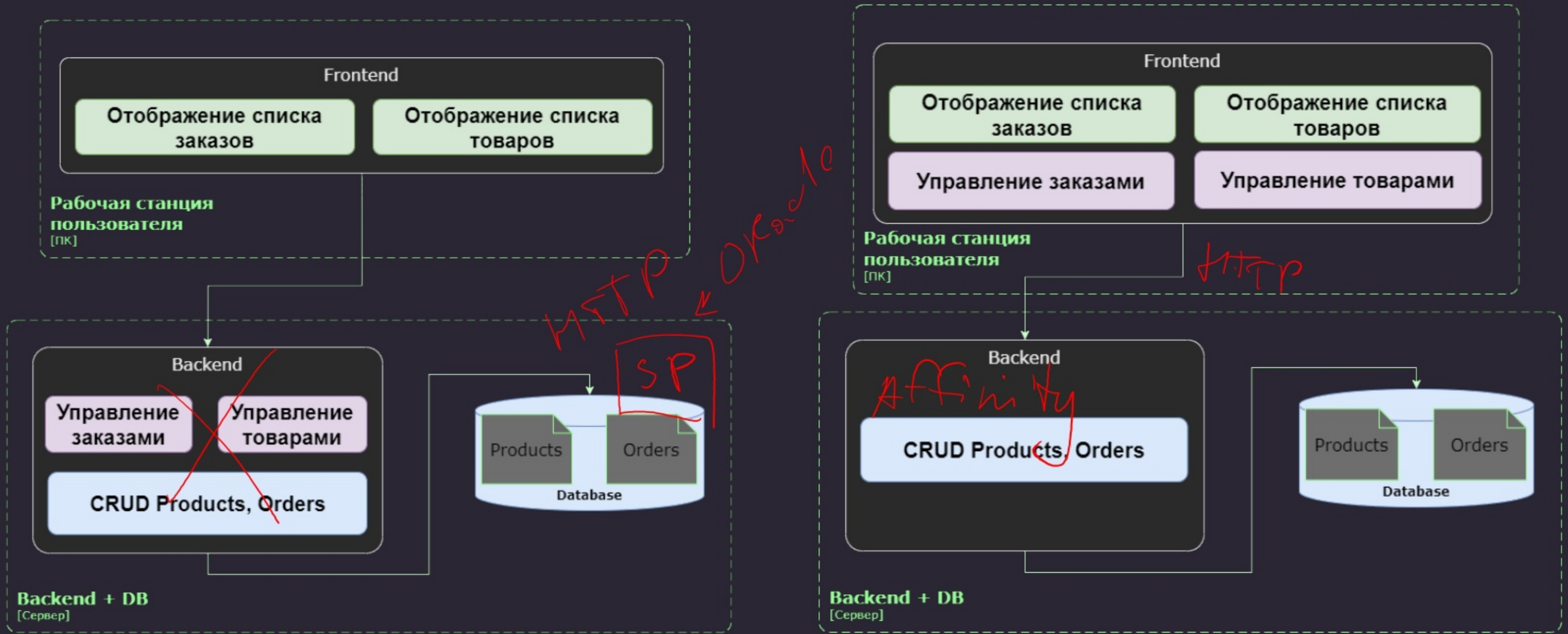
Engine

HTML + CSS

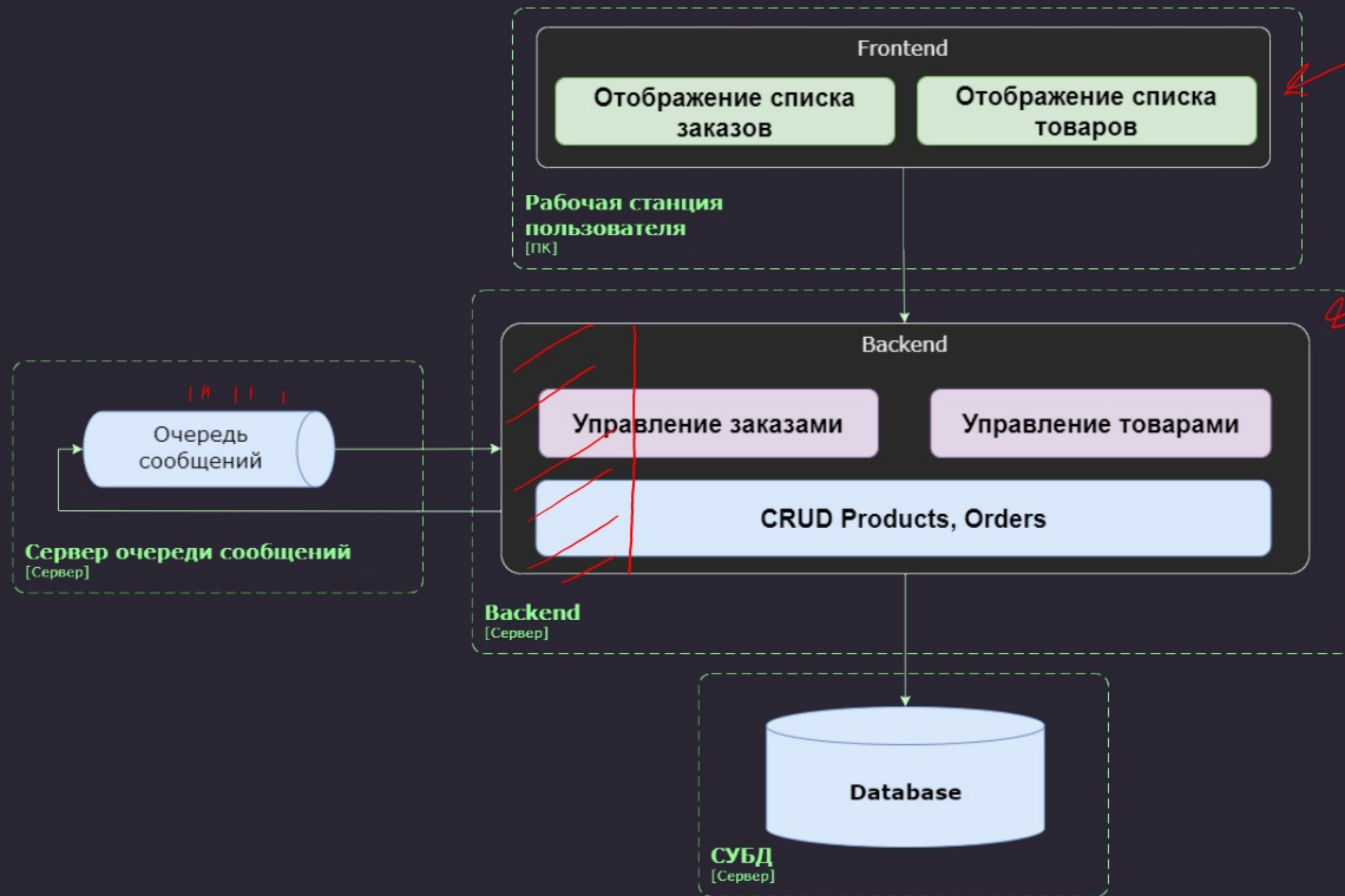
DLL – DevKit



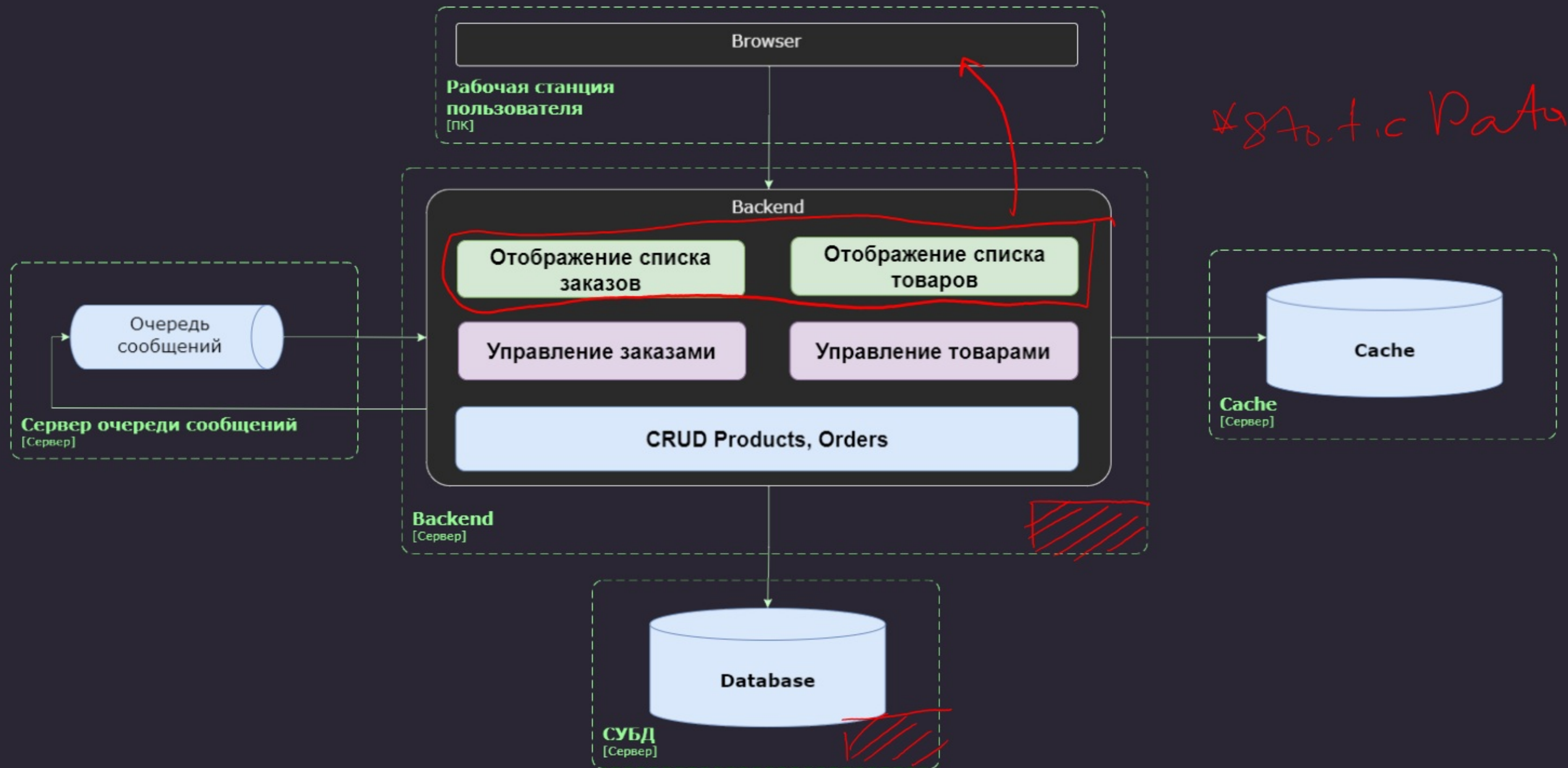
Как выглядят монолиты в архитектуре



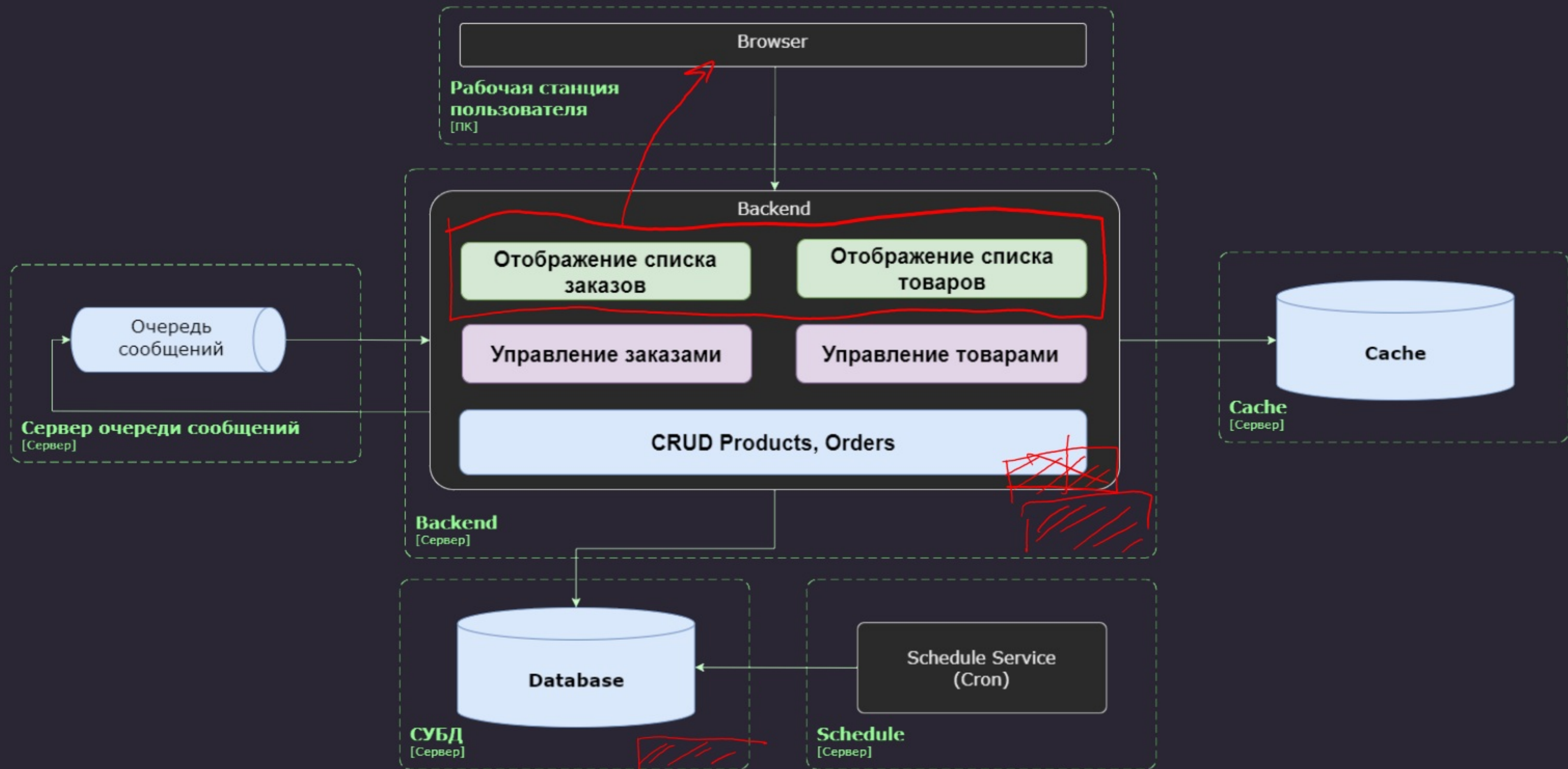
Как выглядят монолиты в архитектуре



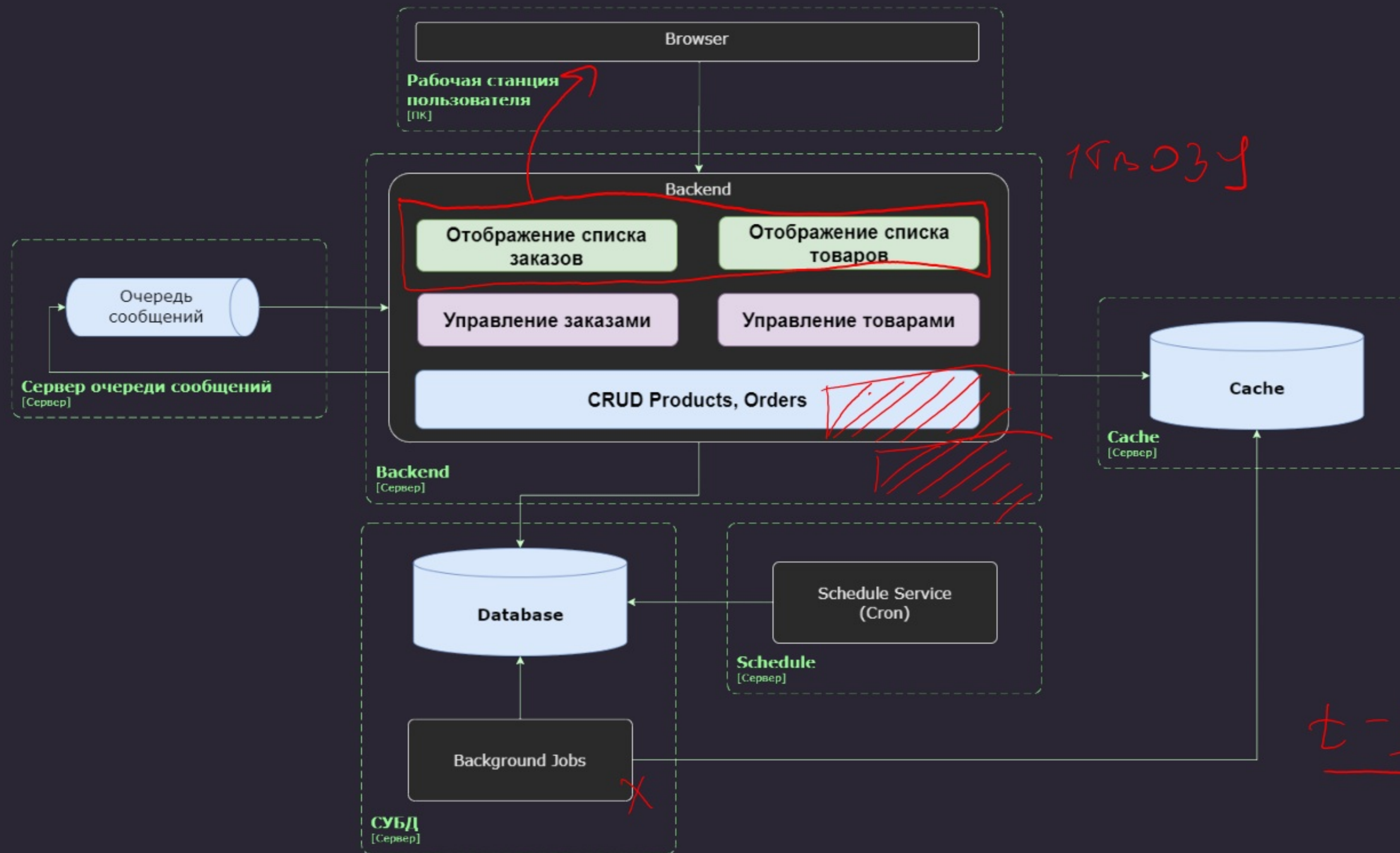
Как выглядят монолиты в архитектуре



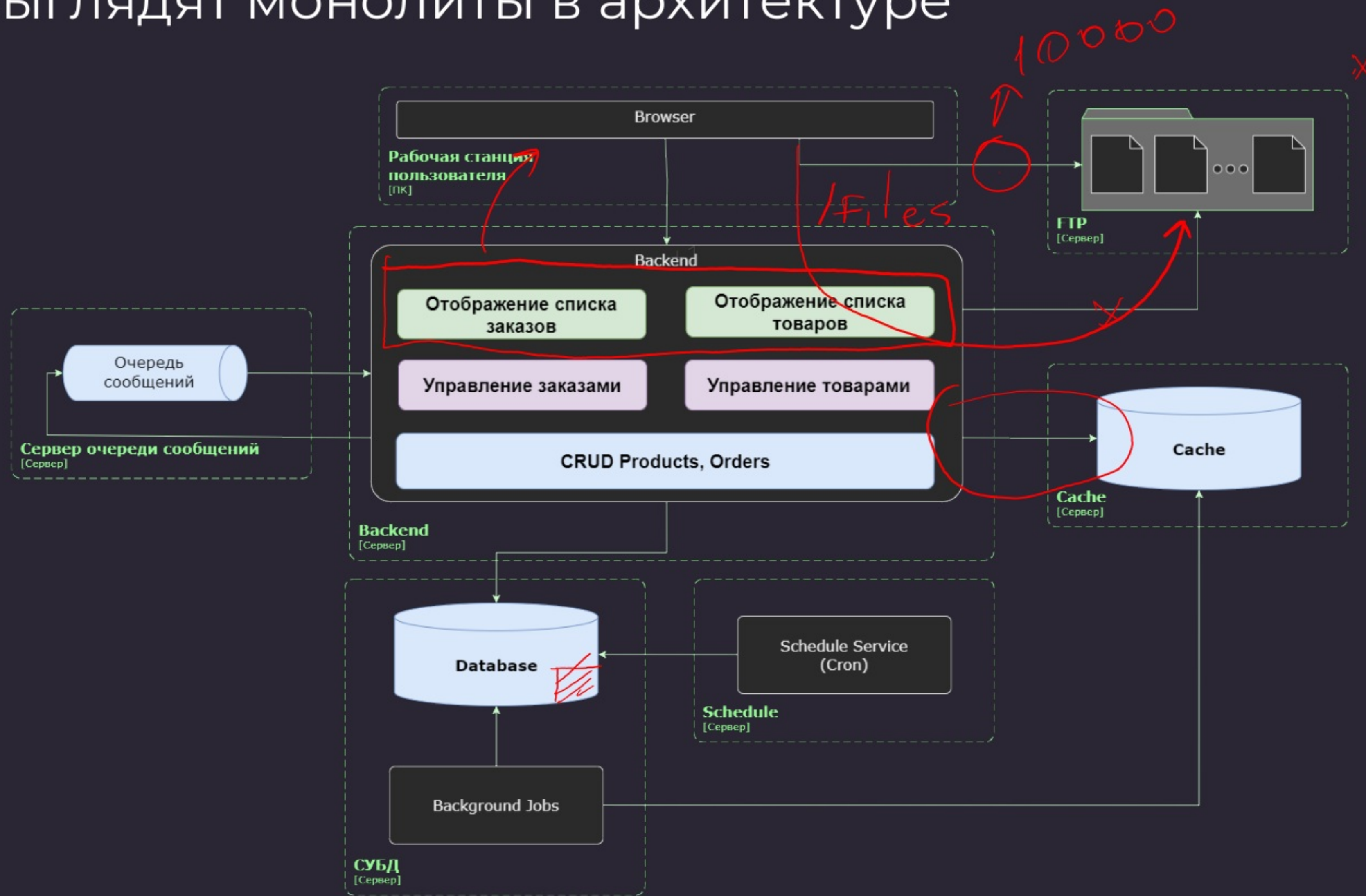
Как выглядят монолиты в архитектуре



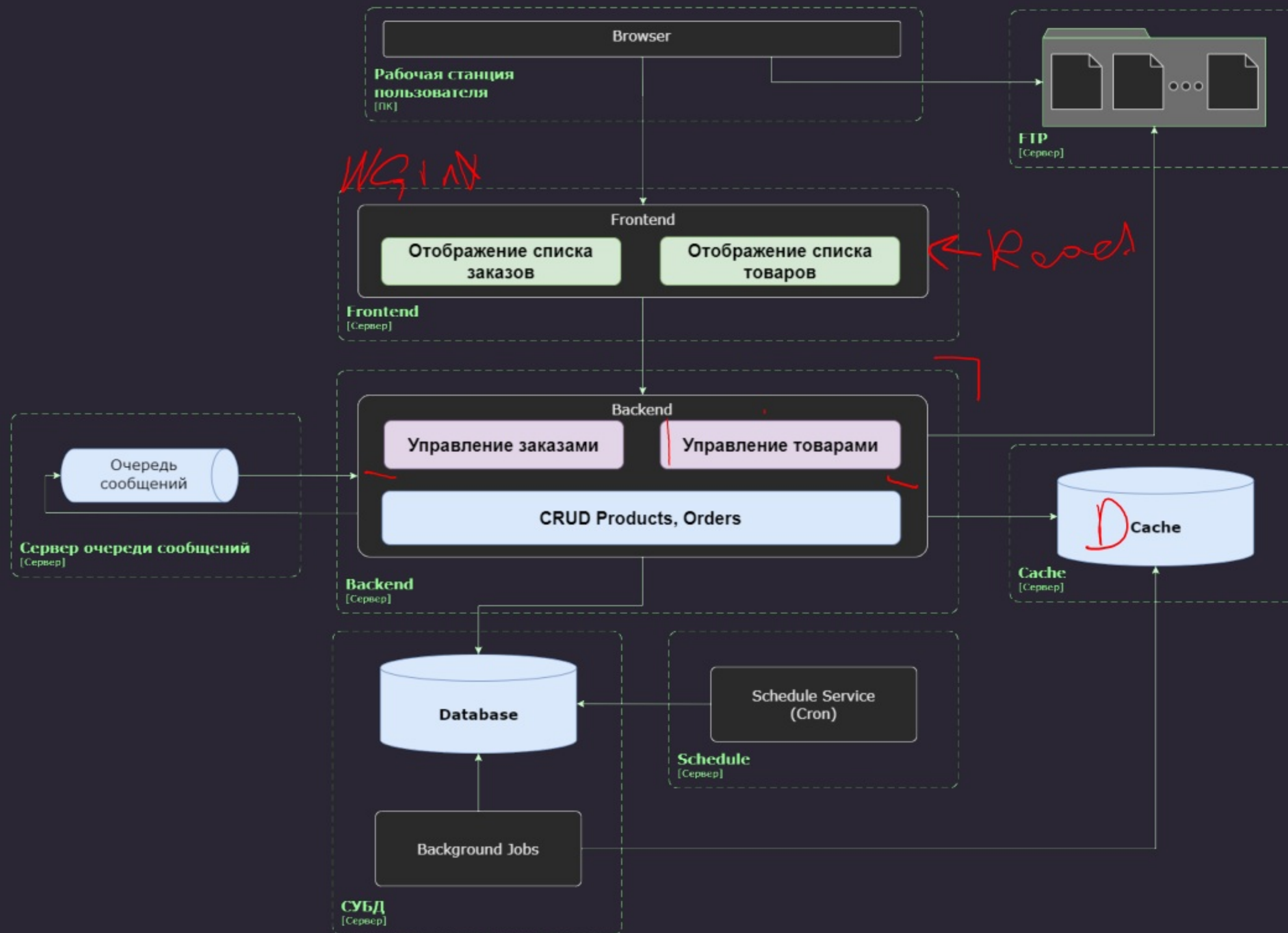
Как выглядят монолиты в архитектуре



Как выглядят монолиты в архитектуре



Как выглядят монолиты в архитектуре



Основные проблемы монолитов

- **Не** Гибкость
- **Сложность** Масштабирования
- **Трудность** в Тестировании *+ 30 минут*
- **Тяжёлая** Отладка
- **Долгая** Доставка (Публикация), **и с остановкой**
- **Не** Устойчивость к падениям
- **Плохая** Доступность *—*
- **Плохая** Отказоустойчивость



Когда монолит полезен

Когда вы пишете десктопное приложение и оно должно работать в offline

offline + online

Когда мы не до конца понимаем каким должен быть продукт на выходе и используем подход MonolithFirst, со следующих точек зрения:

- Бизнес. Функциональное наполнение —
- Поддержка. Условия эксплуатации —
- Скорости вывода продукта на рынок (StartUp) *



MonolithFirst

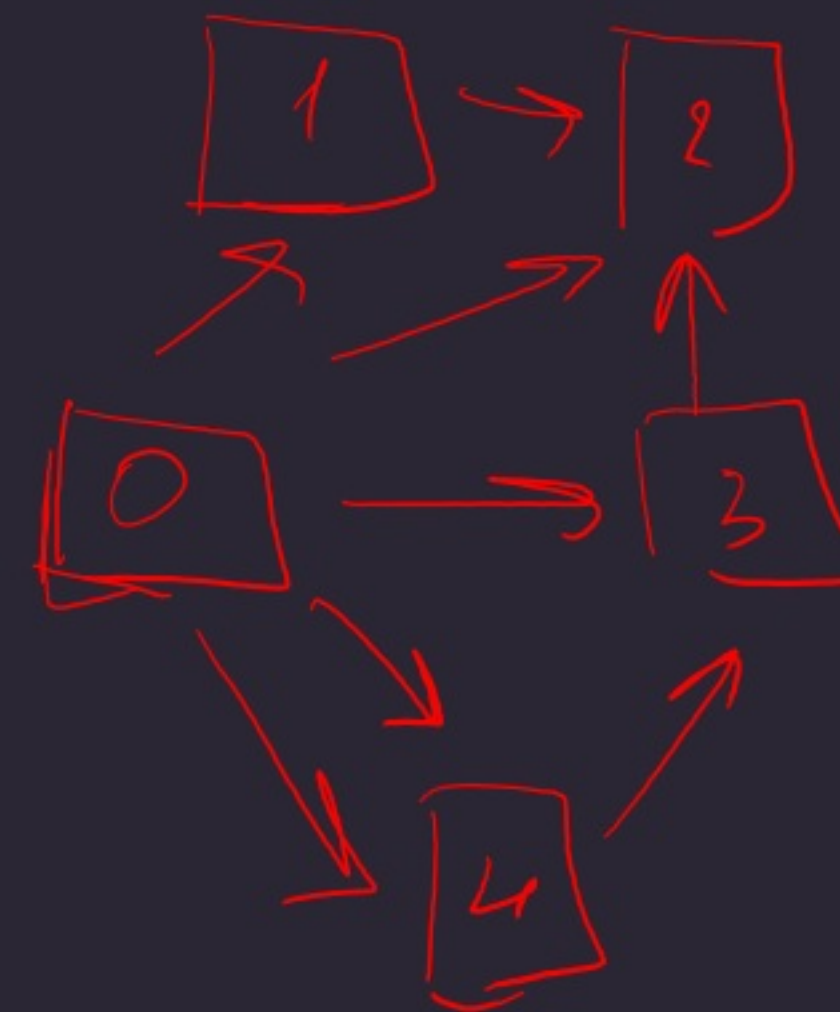
Позволяет понять границы компонентов и их наполнение

Показывает узкие места приложения

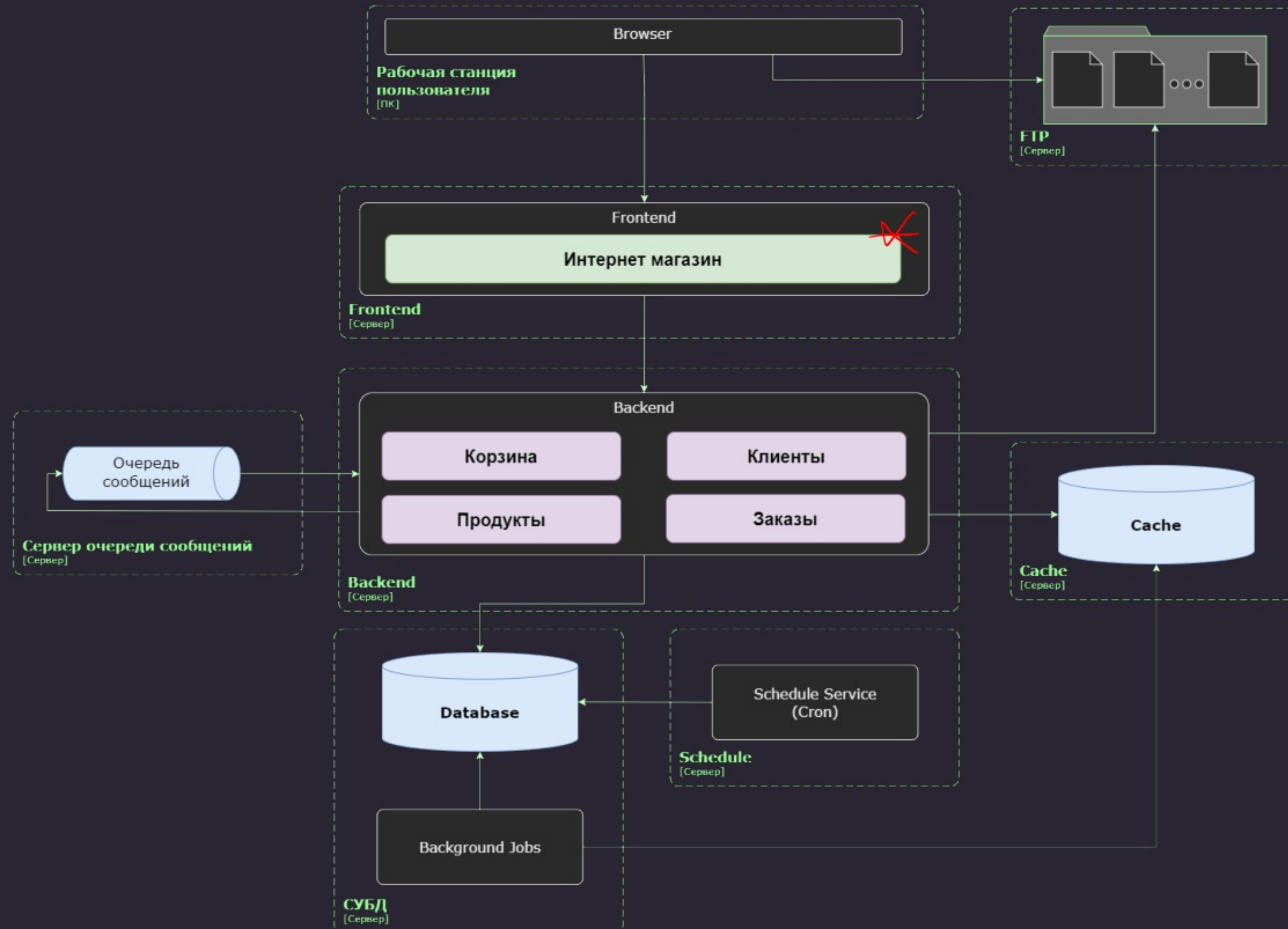
Помогает понять общие микросервисы или Core *



В конечном итоге разбивается на микросервисы

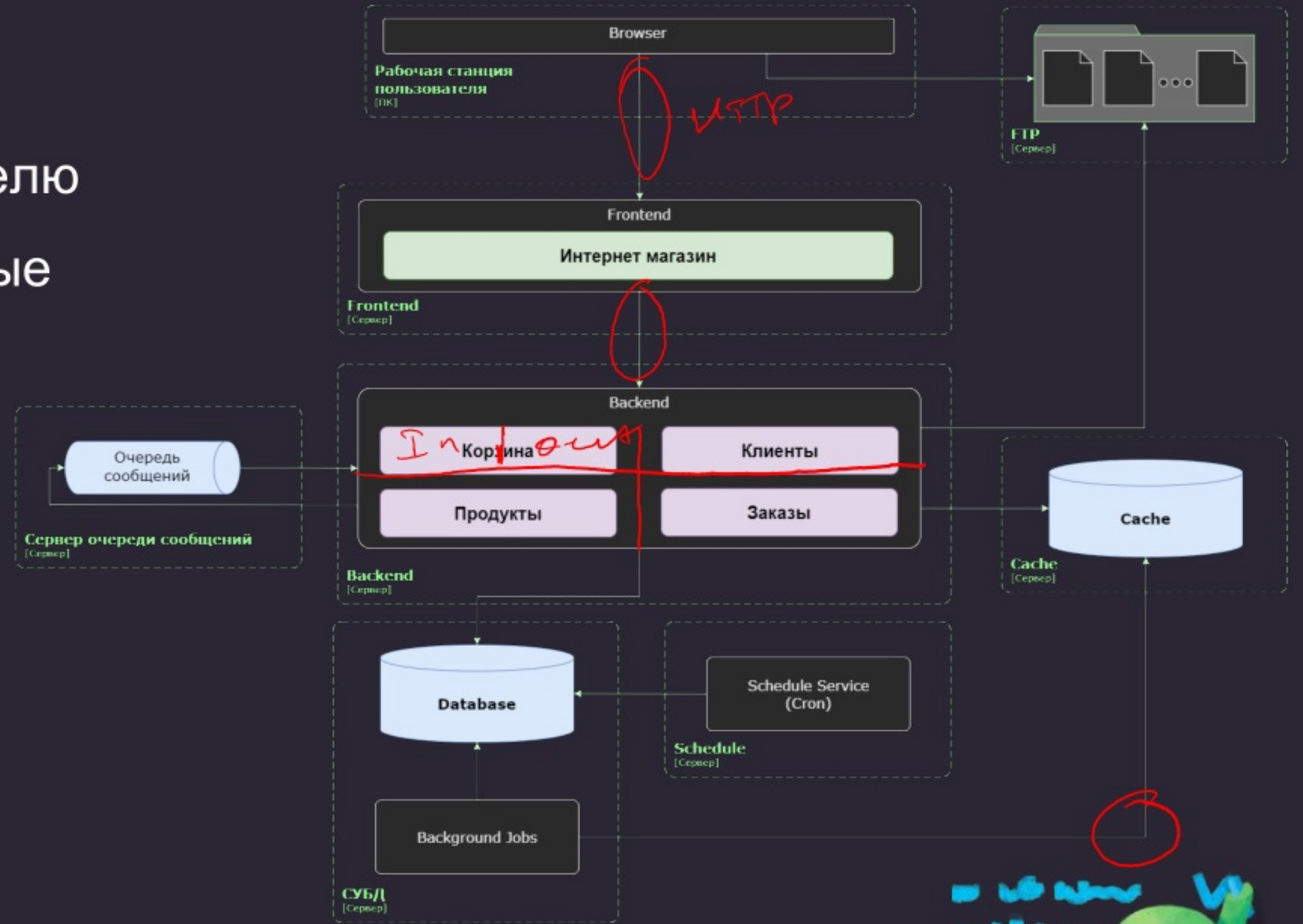


Анализ монолита



Анализ монолита

- Users** – Кто пользуется приложением
- Functions** – То что предоставляется пользователю
- Domains** – Сервисы, сущности, или какие данные
- Location** – Где и как хранятся данные
- Inventory** – Из чего состоит монолит
- Processing** – Что и где обрабатывается
- Timing** – Когда и с какой задержкой
- Contract** – Модели, структура данных в API
- Technology** – стек, протоколы
- Integration** – Внешние сервисы
- Security** – Безопасность и авторизация
- Logs'n'Metrics** – Мониторинг и логирование



Анализ монолита

Users – Кто пользуется приложением

Domains – Сервисы, сущности, или какие данные

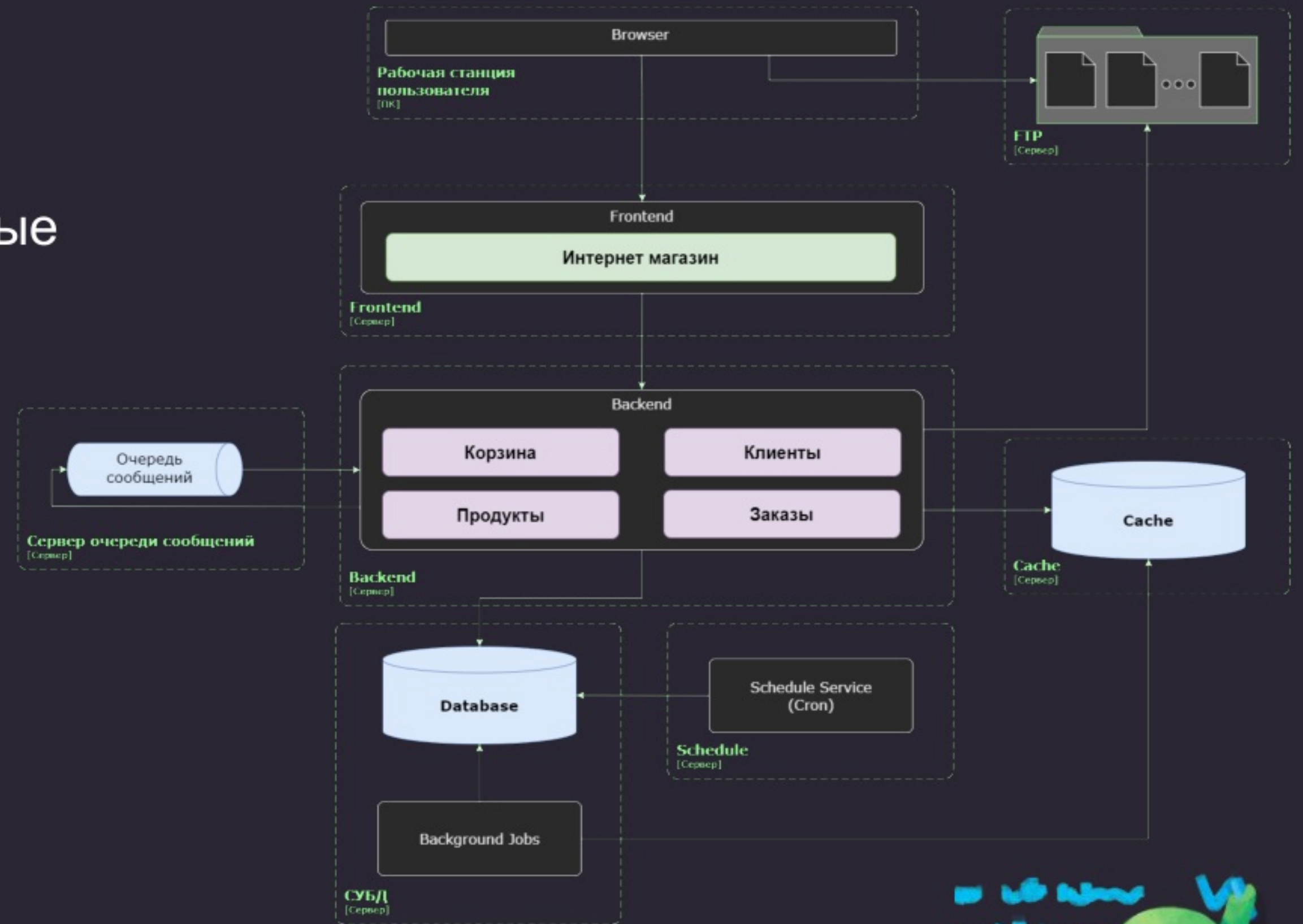
Location – Где и как хранятся данные

Inventory – Из чего состоит монолит

Processing – Что и где обрабатывается

Timing – Когда и с какой задержкой

Technology – стек, протоколы



Анализ монолита

Users – Кто пользуется приложением

Покупатели

← M

Администратор

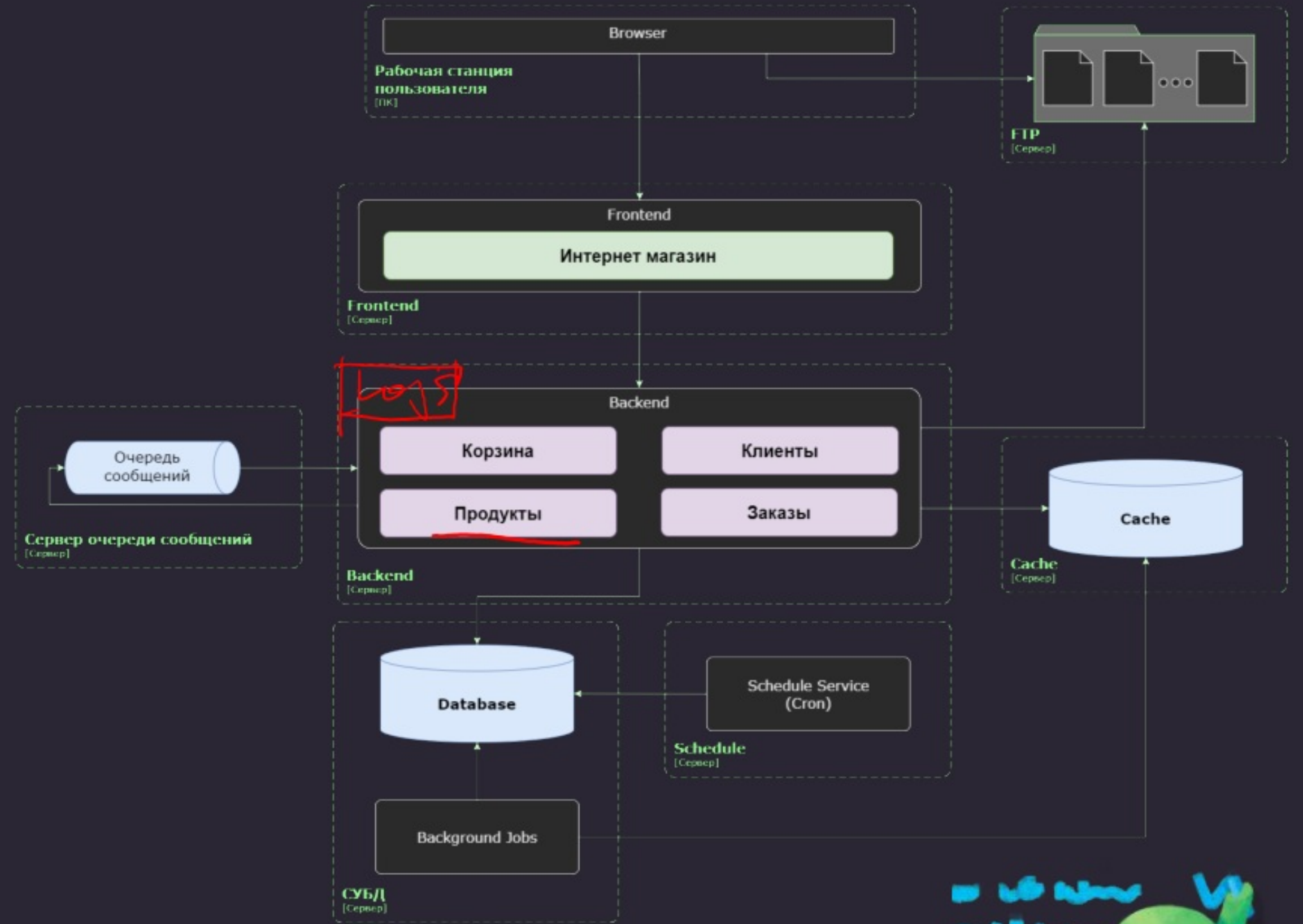
- 2 - 3

Модератор

- 7

Менеджер

- 10



Анализ монолита

Users – Кто пользуется приложением

Domains – Сервисы, сущности, или какие данные

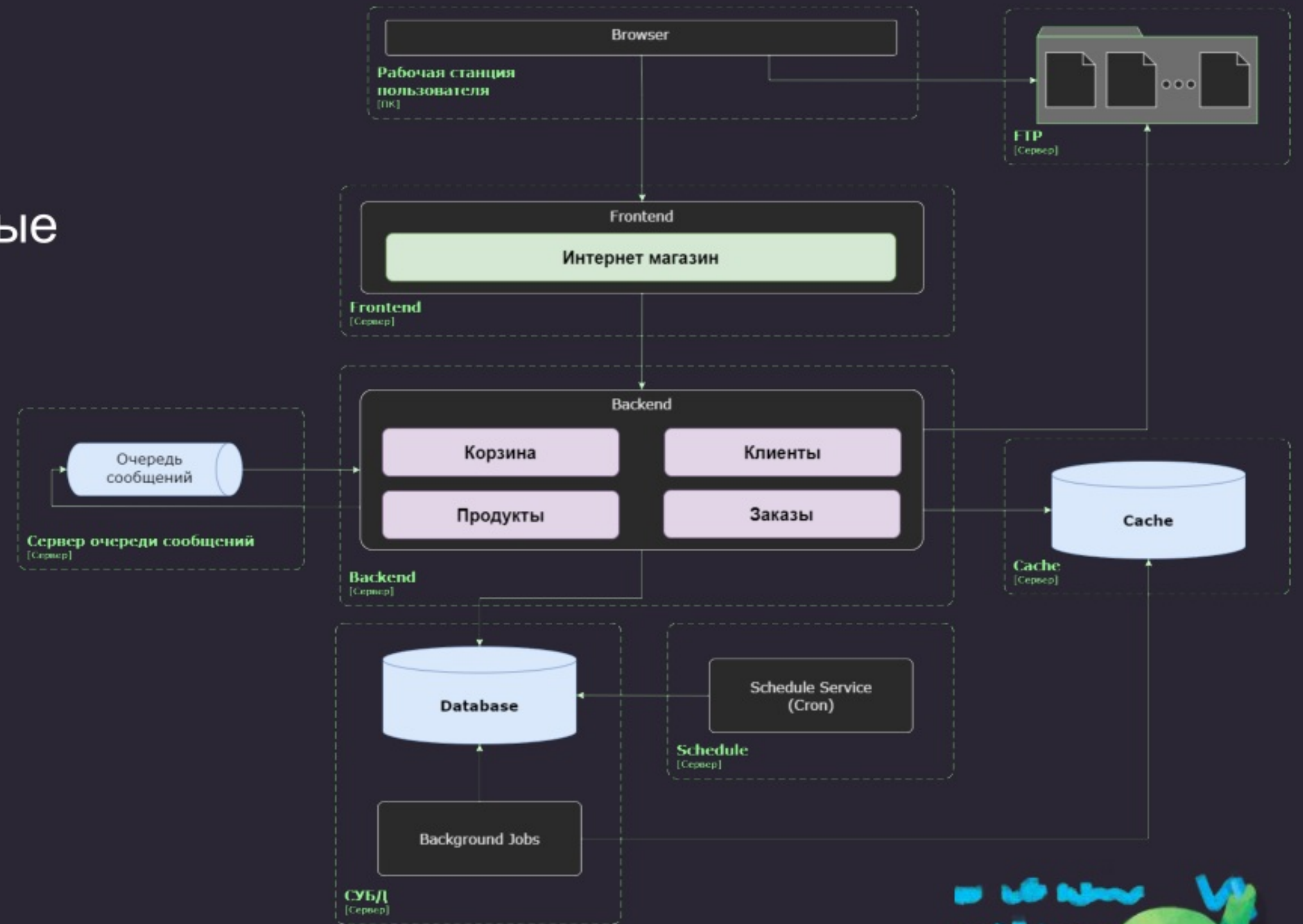
Location – Где и как хранятся данные

Inventory – Из чего состоит монолит

Processing – Что и где обрабатывается

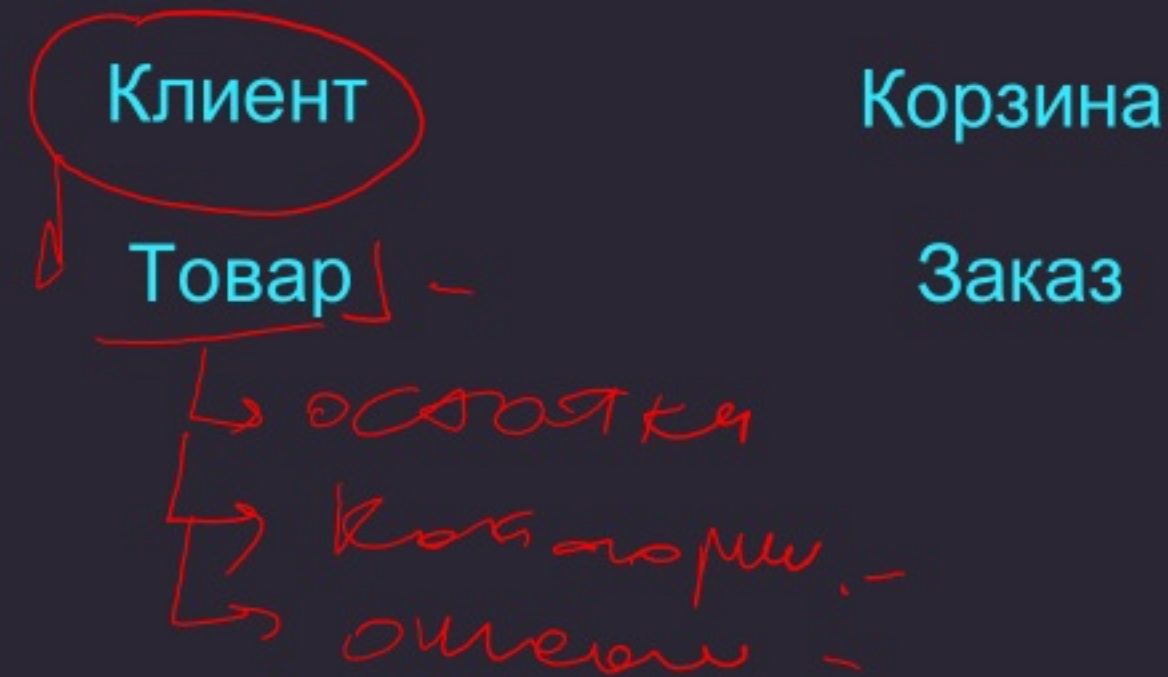
Timing – Когда и с какой задержкой

Technology – стек, протоколы

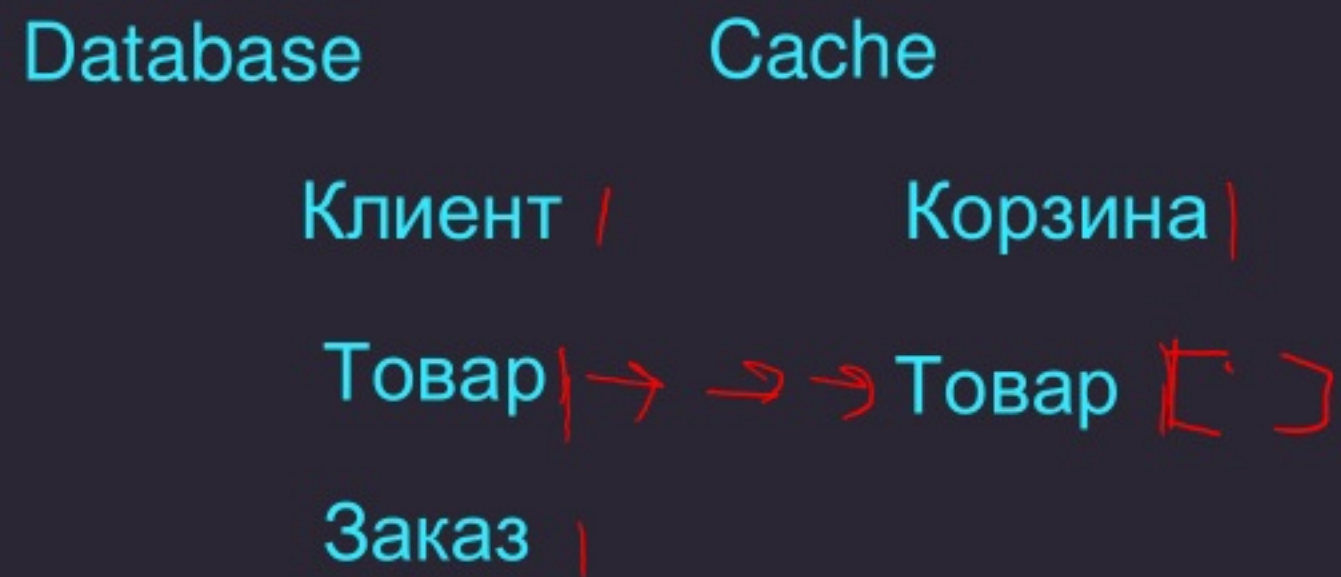


Анализ монолита

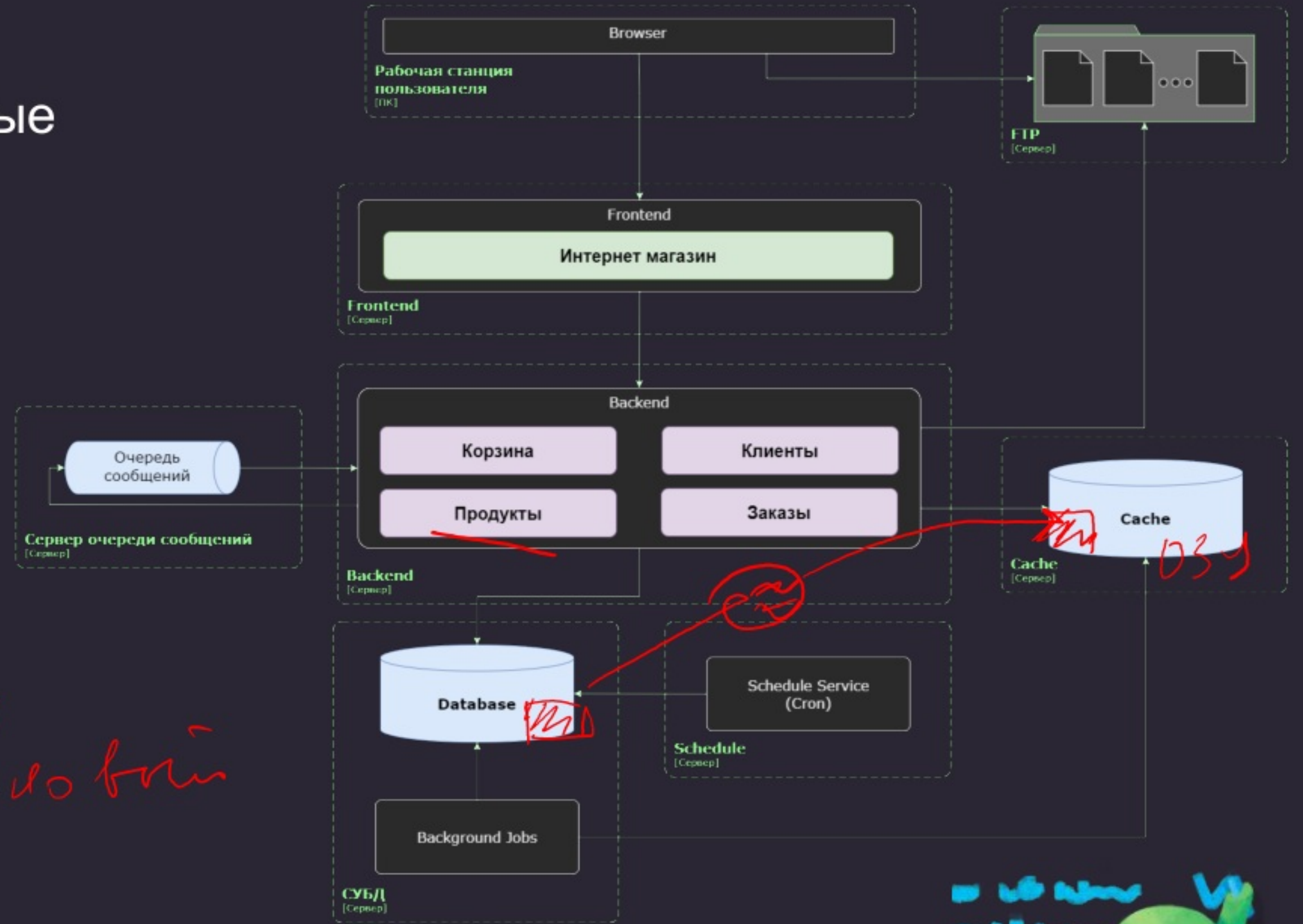
Domains – Сервисы, сущности, или какие данные



Location – Где и как хранятся данные



→ +1
→ Fido bin



Handwritten notes in blue and red, partially obscured by a green bear character.



Анализ монолита

Users — Кто пользуется приложением

Domains — Сервисы, сущности, или какие данные

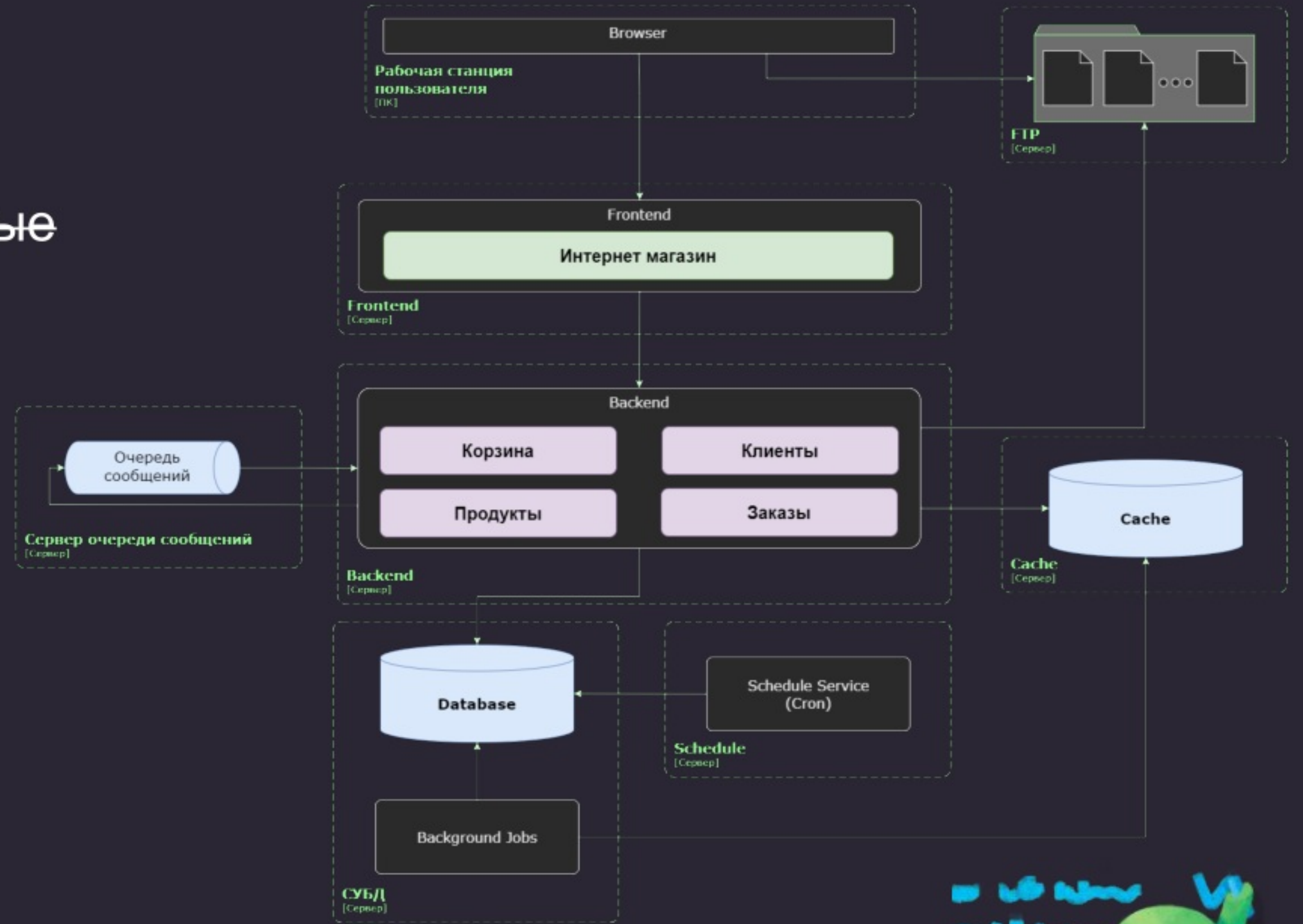
Location — Где и как хранятся данные

Inventory — Из чего состоит монолит

Processing — Что и где обрабатывается

Timing — Когда и с какой задержкой

Technology — стек, протоколы

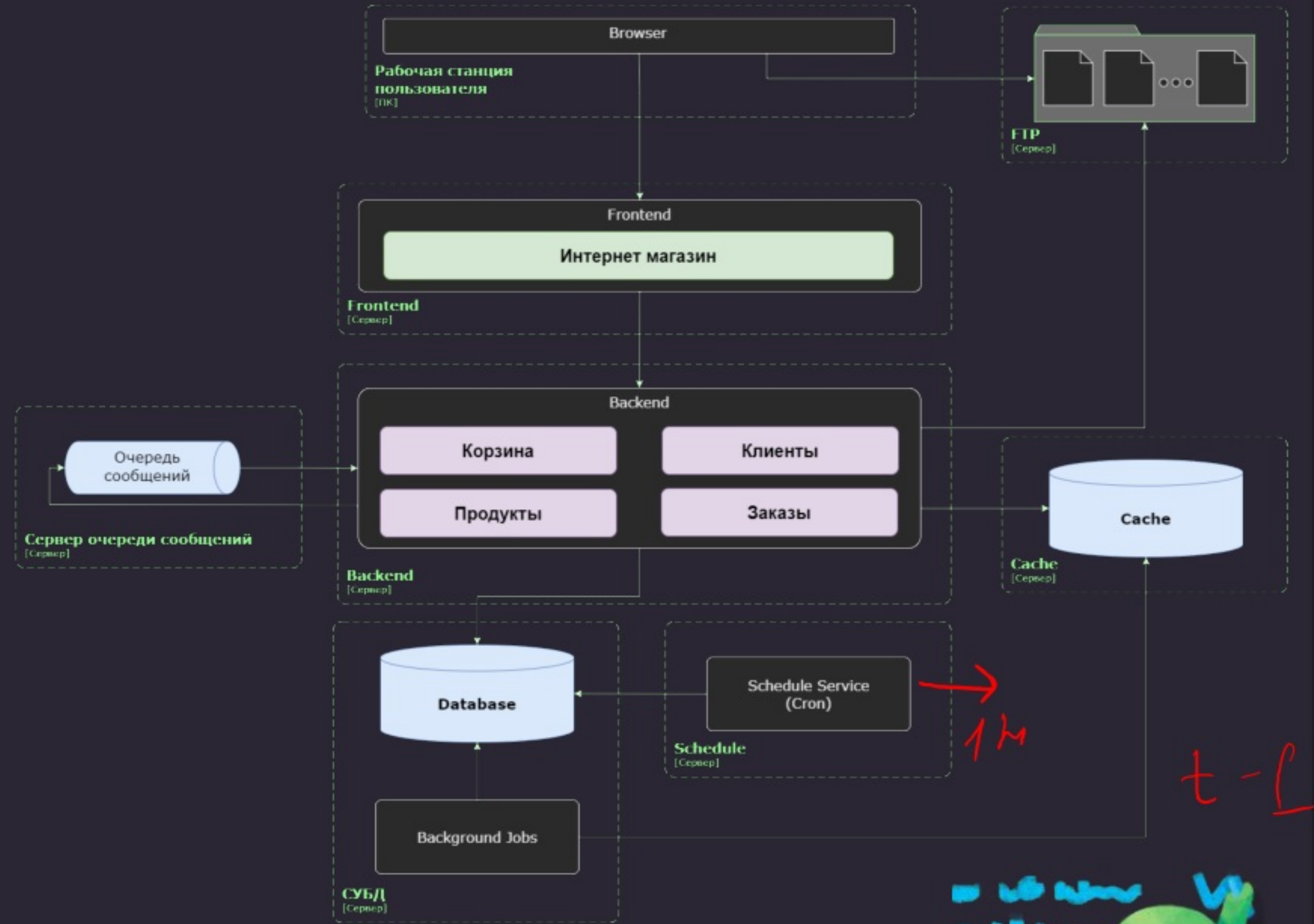


Анализ монолита

Timing – Когда и с какой задержкой

Уведомления – раз в час

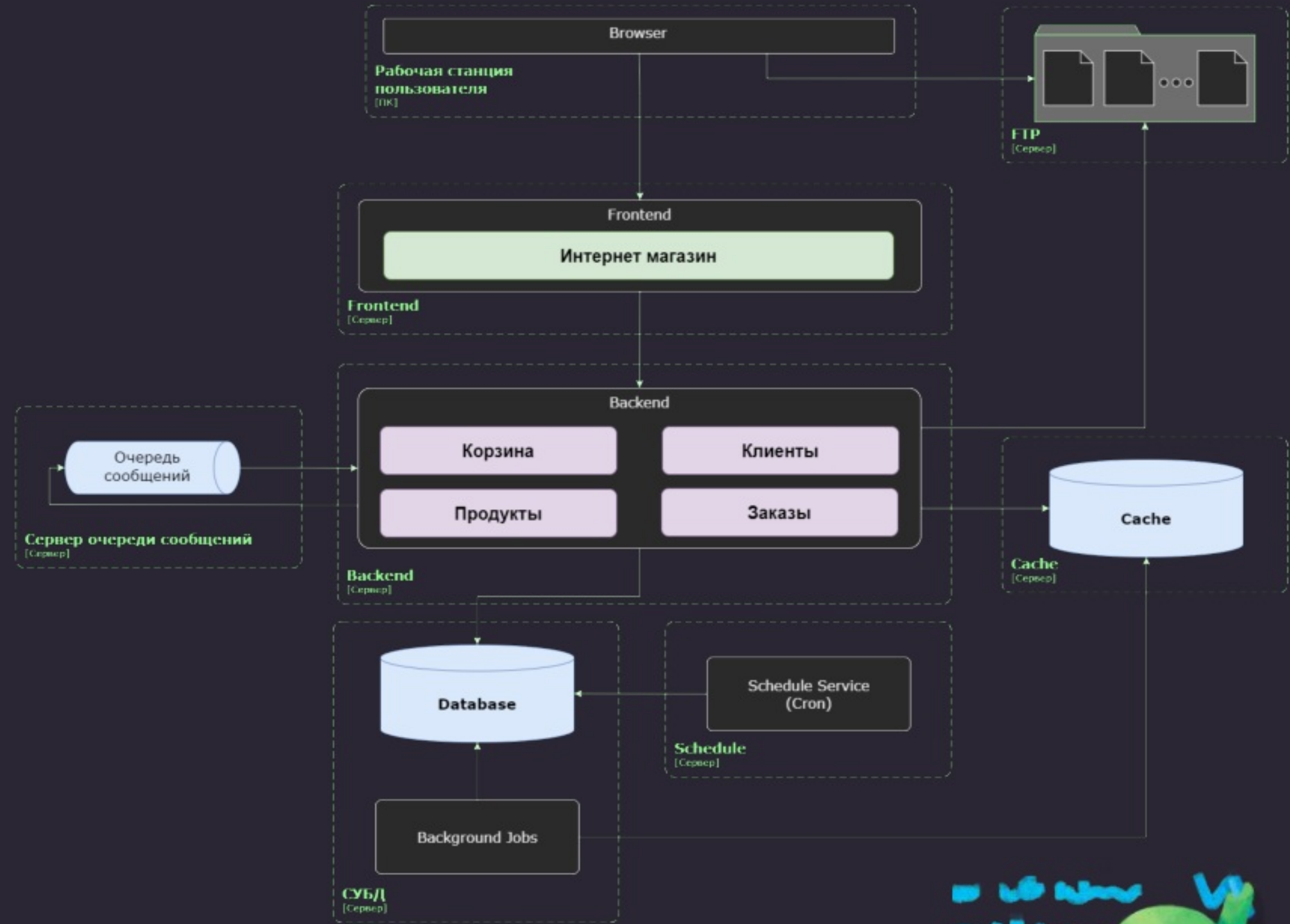
Синхронизация – раз в день



Анализ монолита

Technology – стек, протоколы

C#,
MSSQL,
Redis,
Kafka,
FTP,
React
HTTPS,
RESP,
TCP



Вопросы от спикера

Какие типы MVC были названы в лекции? +1

Что делать при смене контракта? +1

Когда можно\удобней вынести бизнес логику на сторону клиента? +2

Какая важная деталь была упущена в архитектуре? + все баллы из DC

Анализ монолита

