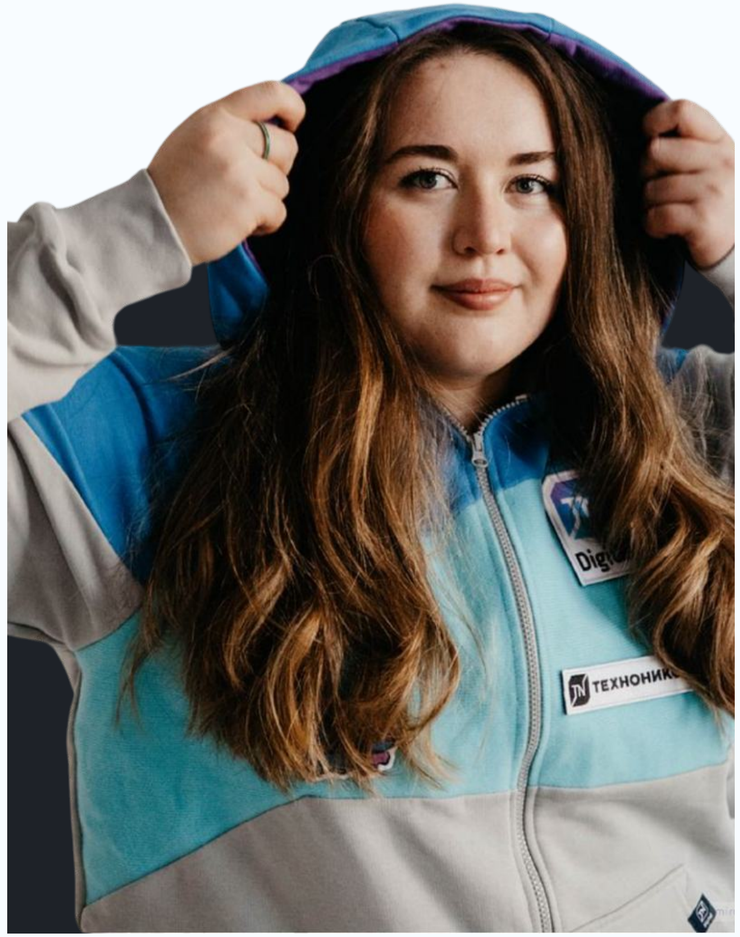
Секреты перехода к модульному монолиту

Причем тут REST API?

ТЕХНОНИКОЛЬ Digital



Альбина Бикбулатова

Discovery lead Project Manager

Станислав Балахонов

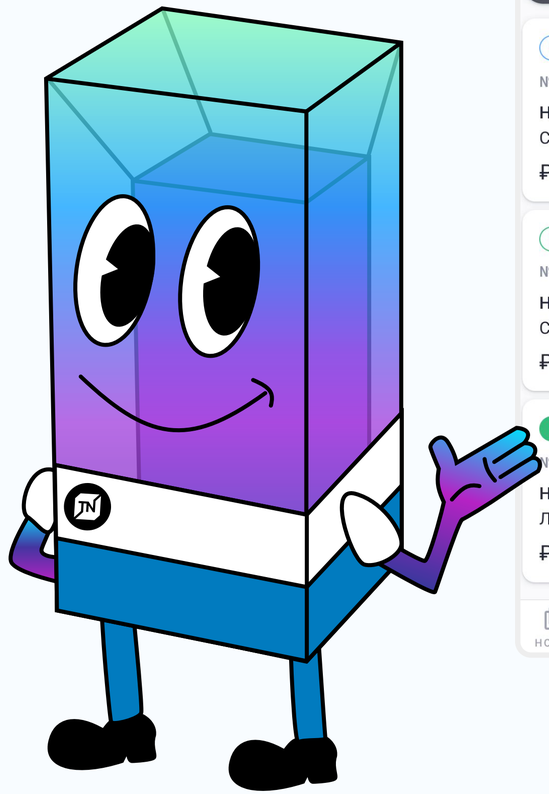
Delivery lead BE-developer

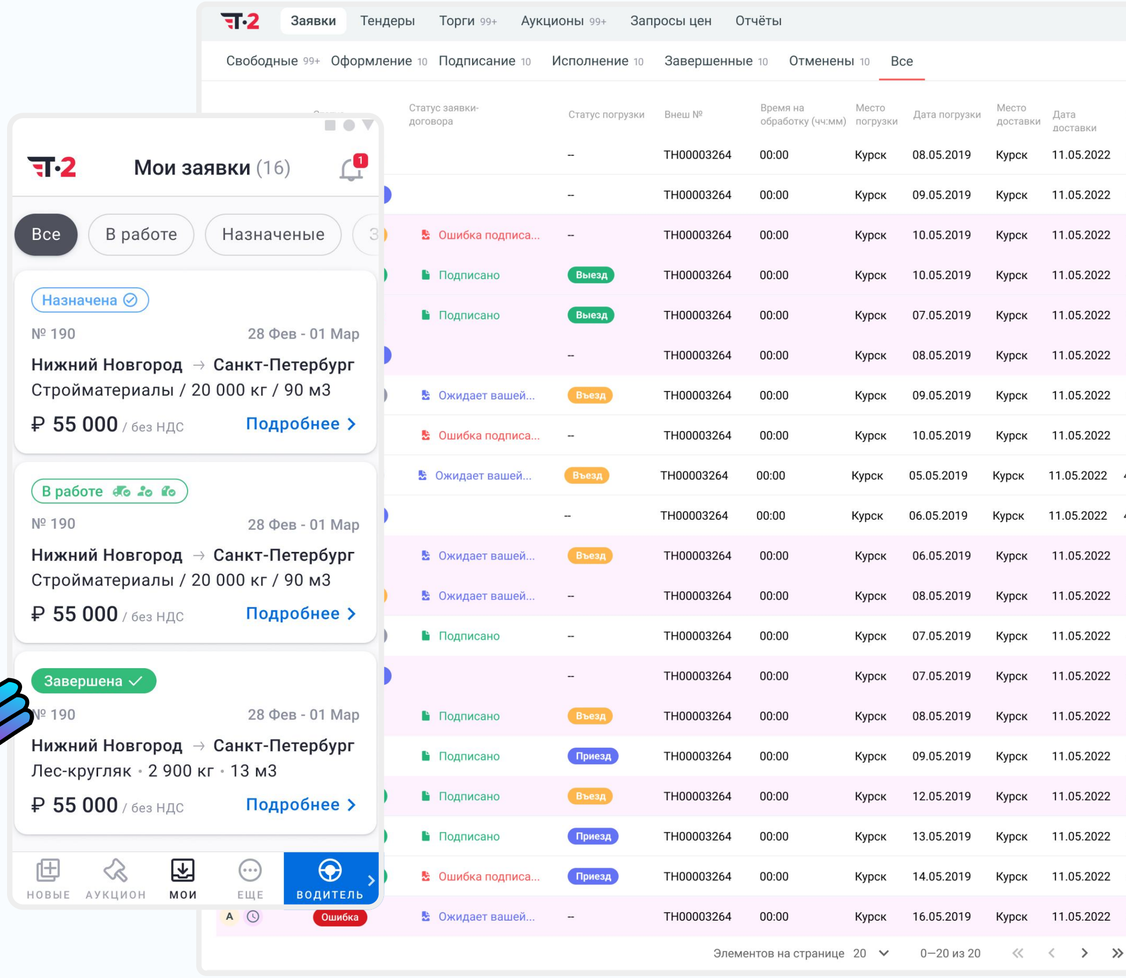
Бизнес процесс логистики ТЕХНОНИКОЛЬ и какое место мы занимаем

Автоматизация бизнес-процессов

Transportation Management System

ТЕХНОНИКОЛЬ





Гибридная команда

Почему REST API?

Принцип работы REST API

1С-TMS

Polling — паттерн асинхронного обмена по REST API.

+ Плюсы

Простота разработки

Строгое разделение логики взаимодействия между системами

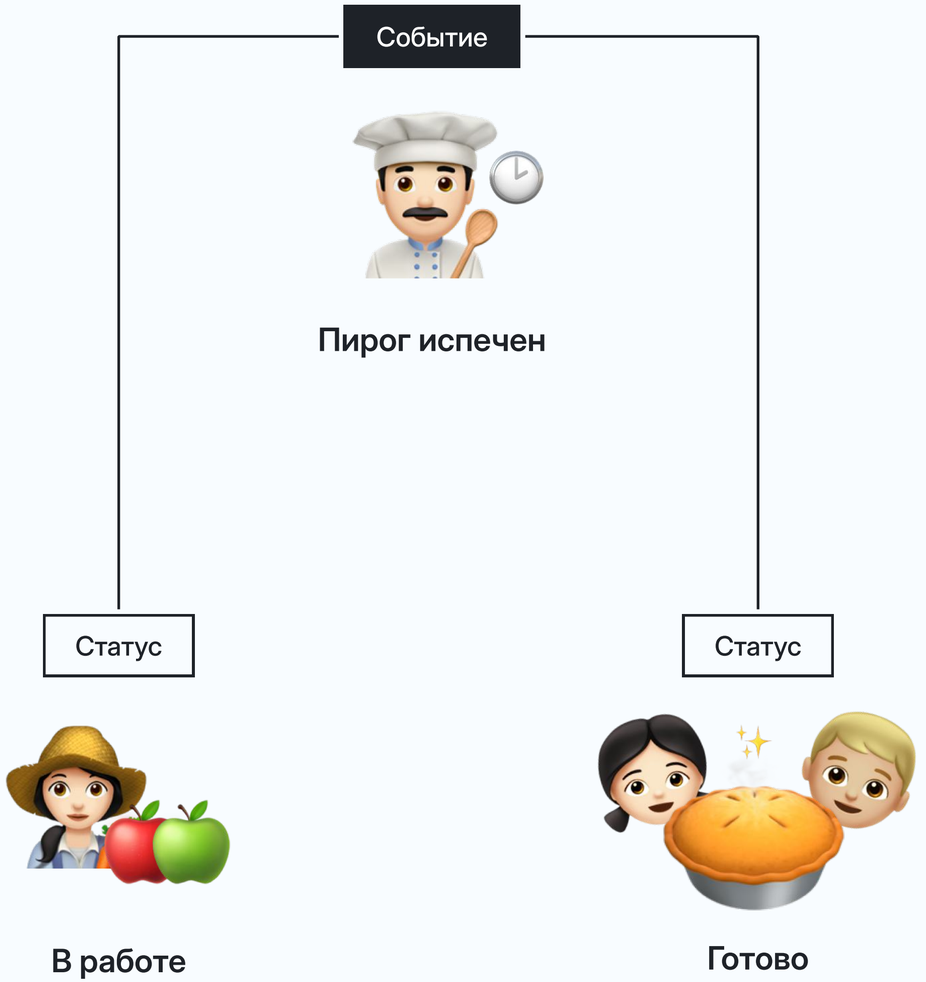
− Минусы

Большое количество запросов

Статусно-событийная схема сущности

Статус — это состояние или условие, в котором находится объект или процесс.

Событие — это результат действия или изменение, которое происходит в системе или программе.

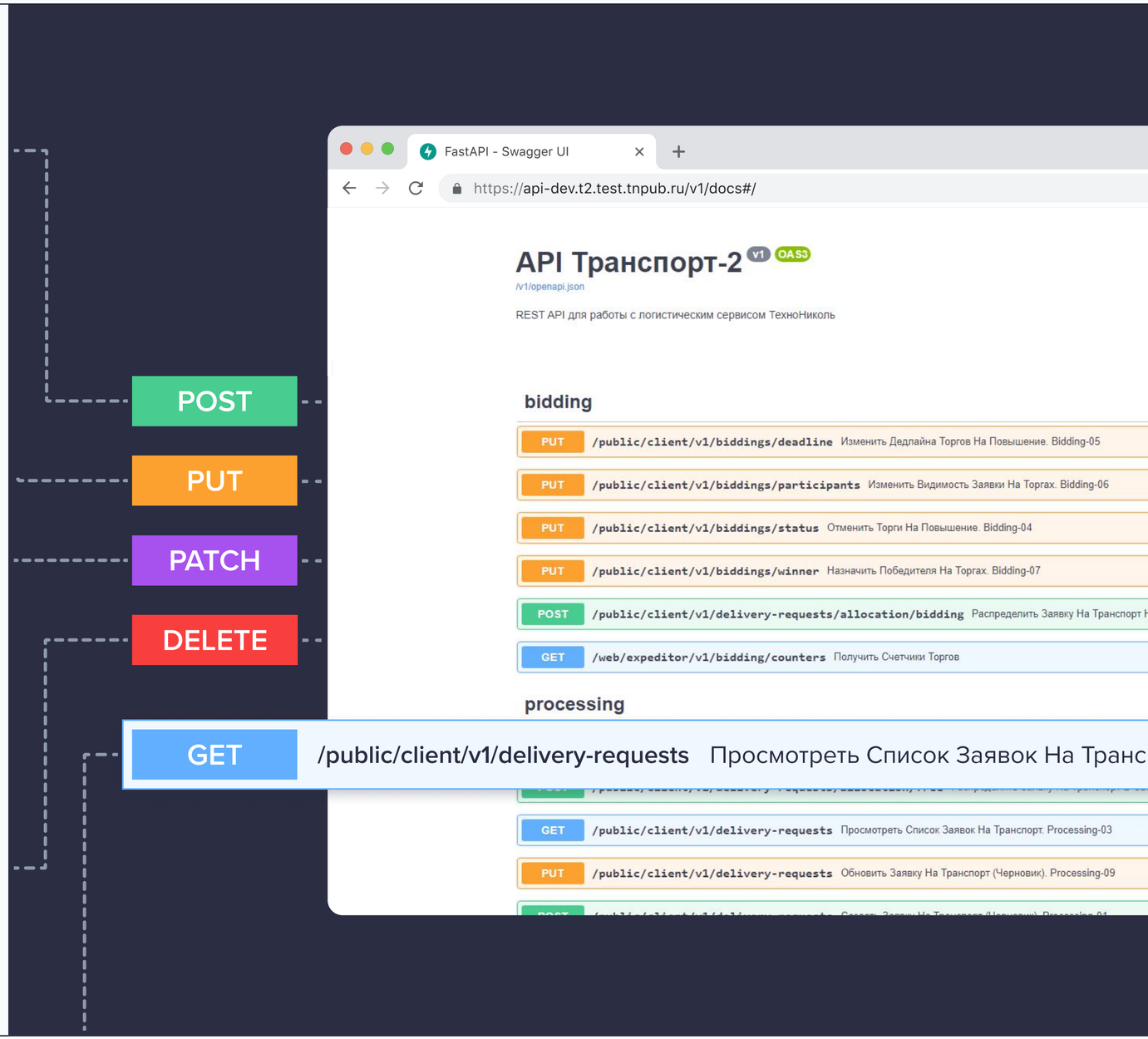


Новый функционал 1С-TMS

В процессе проектирования бизнес-логики обмена 1С-TMS мы заложили много нового функционала, который в прежнем обмене не был реализован.

Список Endpoints

Конечная точка (endpoint) — адрес,
по которому отправляется запрос.
Один и тот же эндпойнт может объединять
несколько действий.



Количество изменений

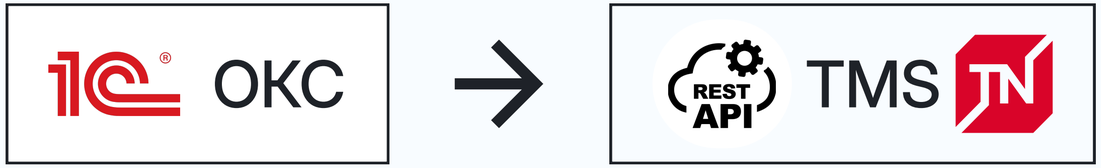
≈ 50 из 70

эндпоинтов влекли за собой либо коренное изменение логики работы, либо создание новой

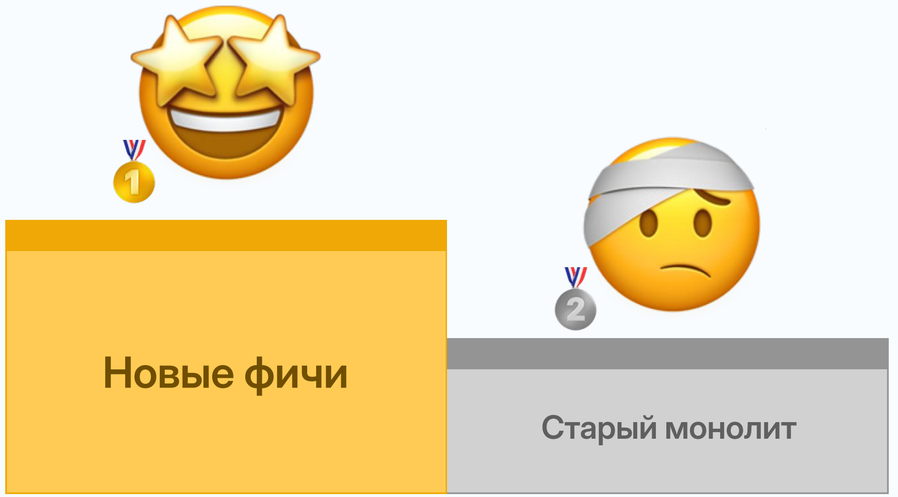
Работа двух интеграций одновременно

Асинхронная интеграция

Синхронная интеграция

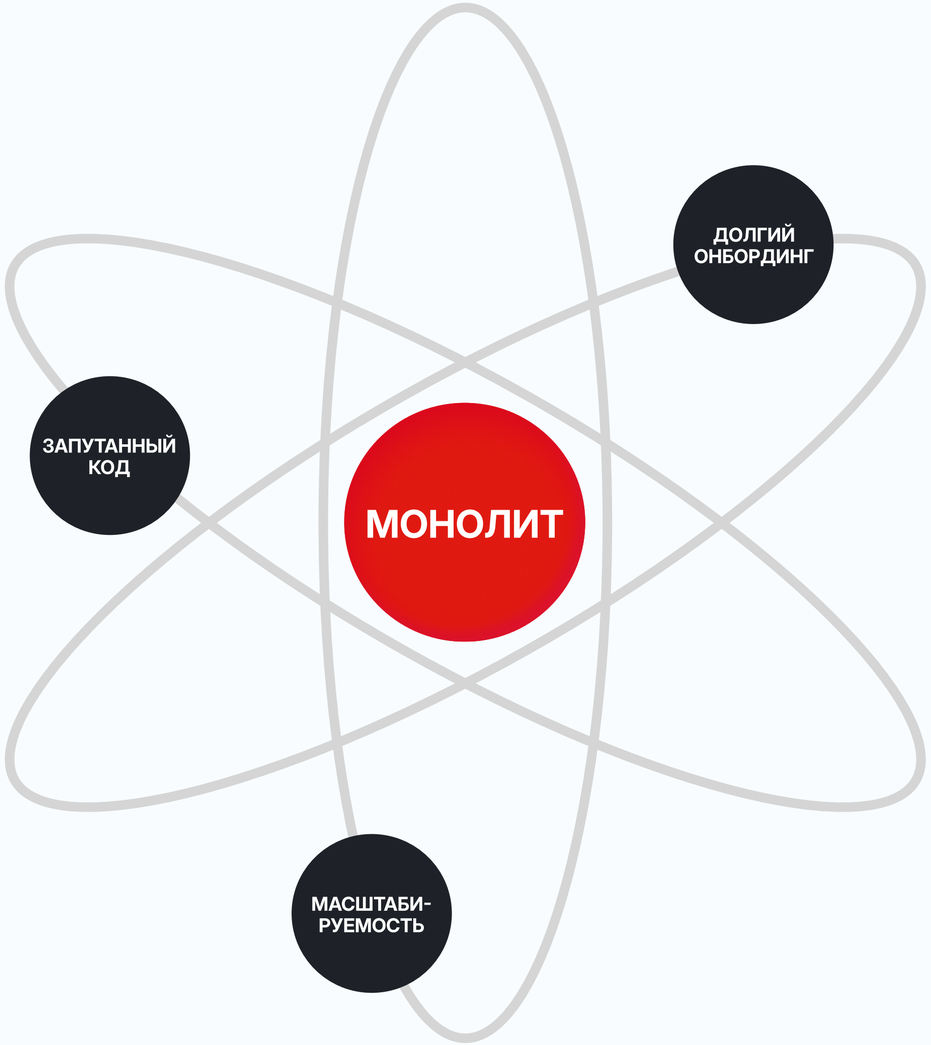


Приоритеты разработки



Боли монолита

Он оброс многими болезнями: большая связность кода, плохая масштабируемость, длительный онбординг новых участников и т.д.

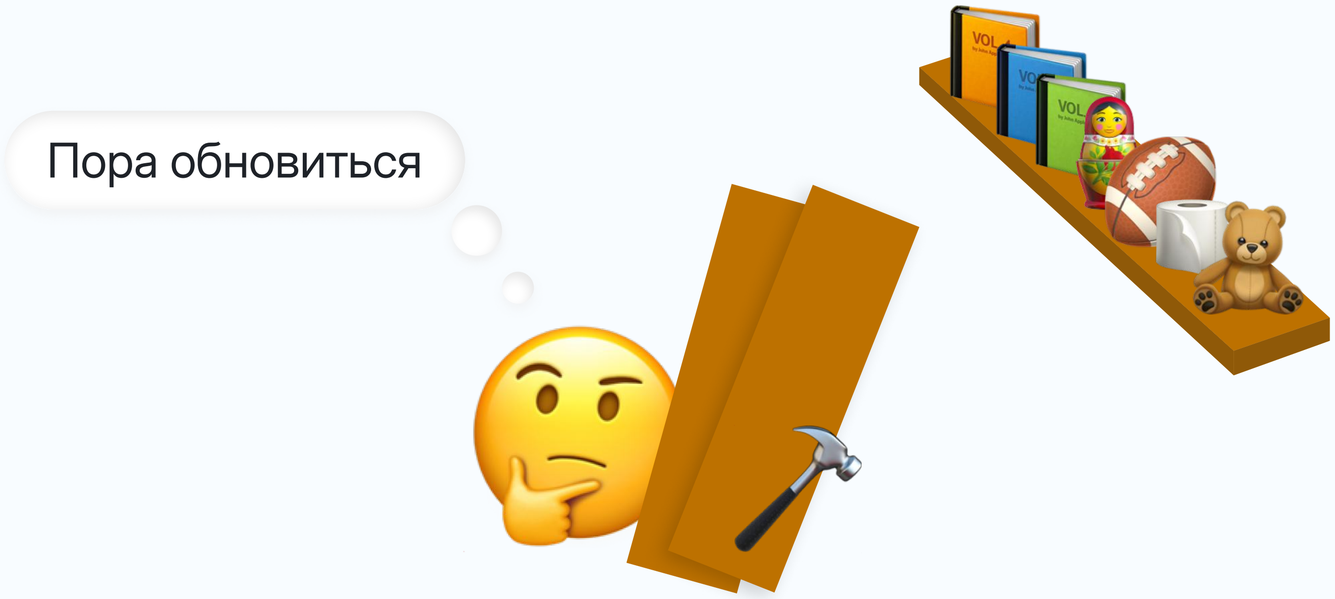


Хаос в бизнес-логике

И в эндпойнтах, и в функциях-обработчиках,
и в ORM — всюду распределена бизнес-логика. Отсутствие порядка
в структурировании кода пагубно влияло на внесение изменений,
онбординг и масштабируемость.

Переход к новому монолиту

Мы приняли решение переходить на новую архитектуру приложения — построить новый монолит, но монолит модульный!



Преимущества монолита и микросервисов

Монолит

- Проще релиз
- Быстрее вызовы между модулями
- Проще в управлении
- Проще эксплуатация
- Проще поддерживать согласованность данных
- Легче проводить рефакторинг

Микросервисы

- Возможно обновление по частям
- Более низкий TTM
- Независимость релизов
- Независимое масштабирование
- Независимая разработка микросервисов
- Высокая доступность отдельных модулей
- Независимое тестирование микросервисов

Monolith first

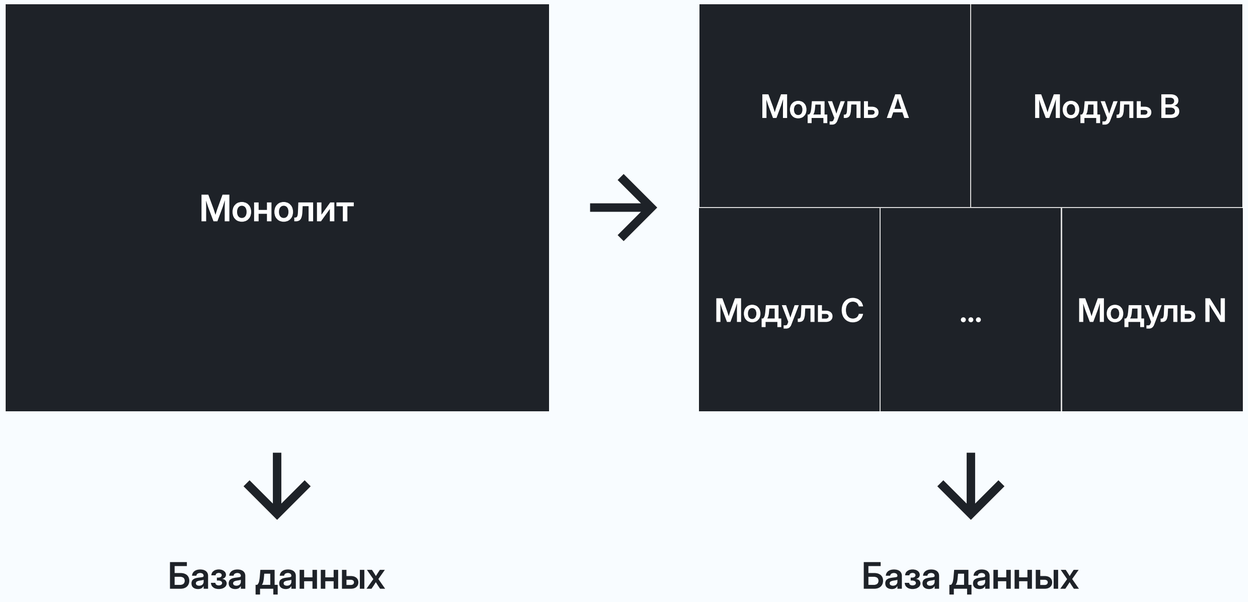
Monolith first потому что Модульный монолит включает плюсы и того и другого.

- Проще релиз
- Быстрее вызовы между модулями
- Легче проводить рефакторинг
- Проще в управлении
- Проще поддерживать согласованность данных
- Проще эксплуатация
- Независимая разработка модулей
- Независимое тестирование модулей
- Более низкий TTM

*Если вы смогли написать хороший модульный монолит, то с высокой вероятностью сделаете хорошую микросервисную архитектуру.

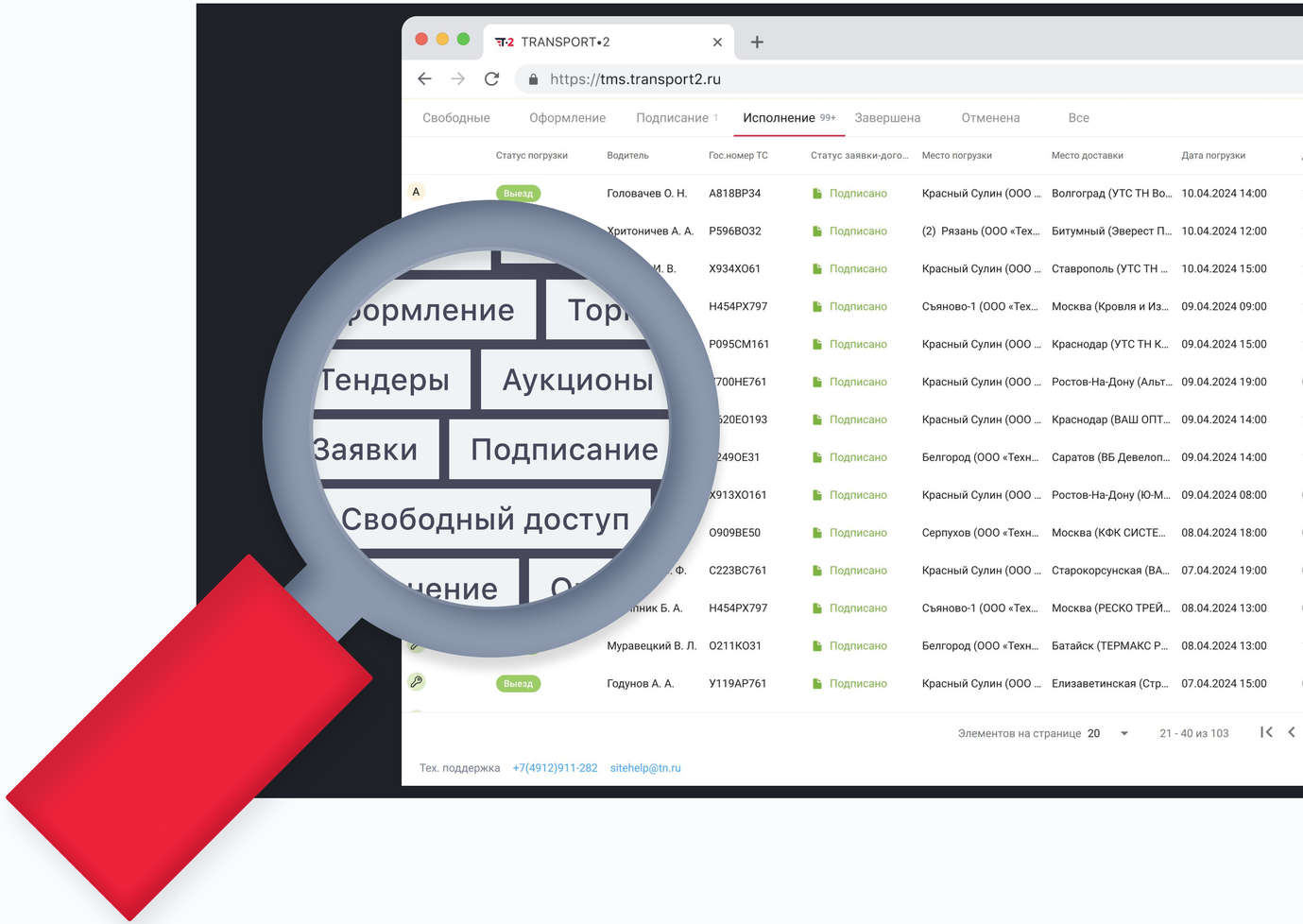
Модульный монолит

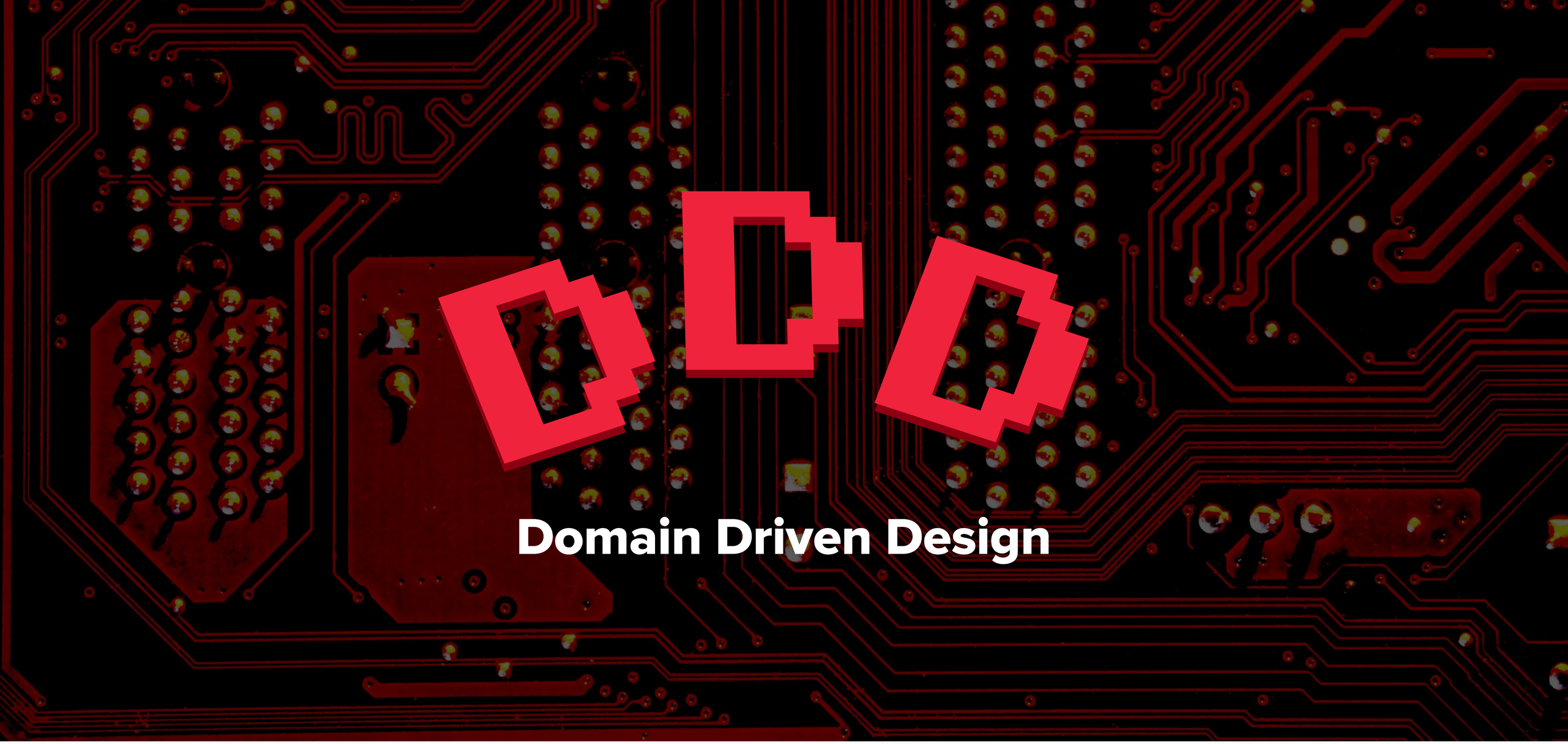
Модульный монолит — это такой монолит, где каждый модуль может функционировать отдельно. Т.е. представляет собой предметную подобласть или бизнес-подпроцесс.



Domain Driven Design

DDD (с англ. Предметно-Ориентированное Проектирование) — это подход к разработке программного обеспечения, нацеленный на изучение предметной области предприятия или его отдельных бизнес-процессов.





Почему DDD?

Ограниченный контекст

Единый язык

Опять на своем птичьем объясняешь!

Практика Event Storming

EventStorming — это воркшоп с простой и четкой структурой, позволяющий собрать вместе экспертов различных компетенций для быстрого, совместного исследования сложной предметной области.

События

Воркшоп начинается с коллективного поиска решений
для событий, связанных с исследуемой предметной областью, которые
расставляют в хронологическом порядке.

Команды

Затем добавляем команды. Они, в свою очередь, описывают операции системы, в результате которых происходят события. Глобально — это методы, которые совершают манипуляции над объектами.

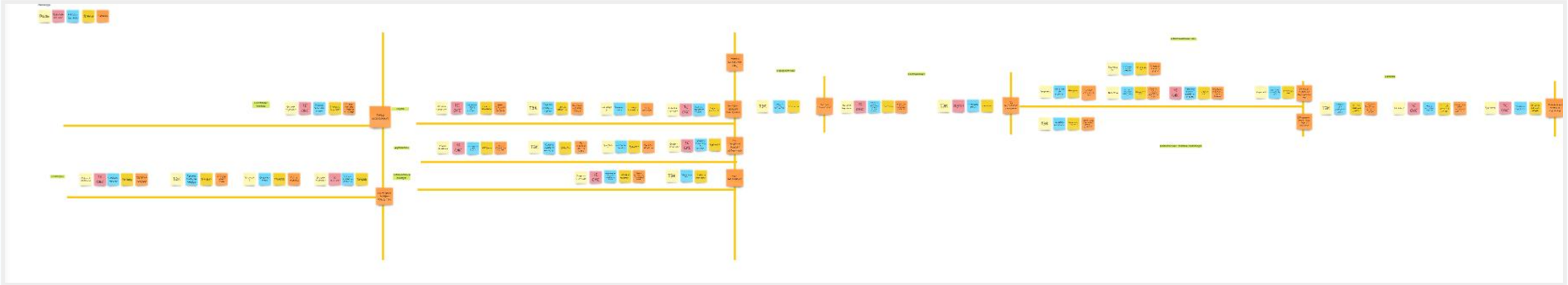
Акторы

Мы добавляли на карту акторов — инициаторов событий в системе.

Модели

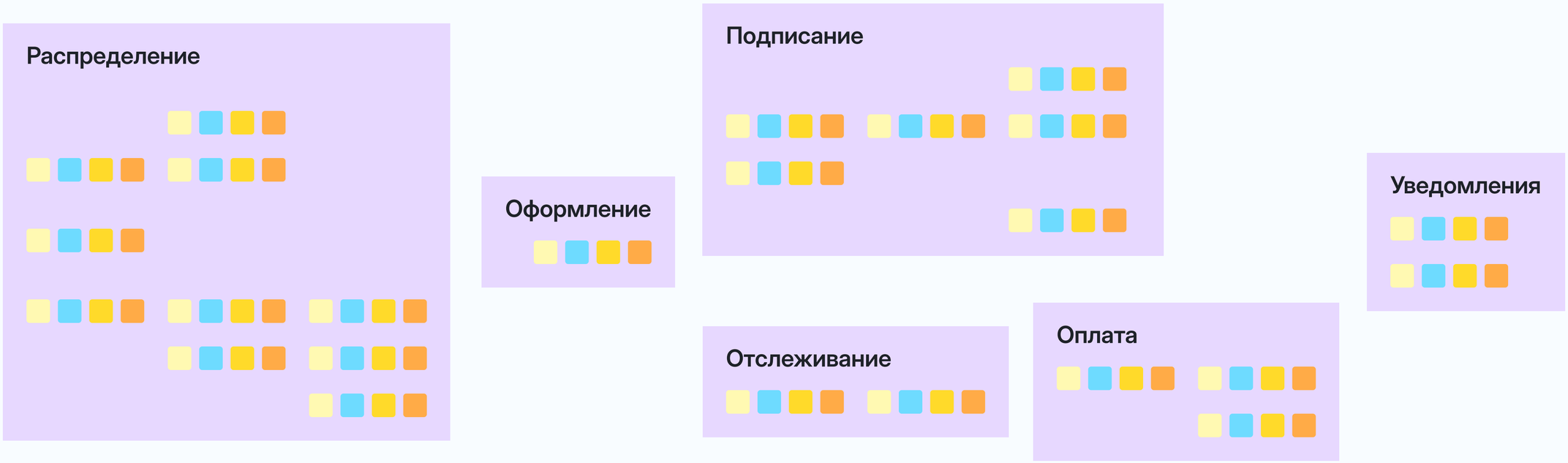
Модель — это олицетворение бизнес-сущности в программном коде.

Event Storming заявки на перевозку



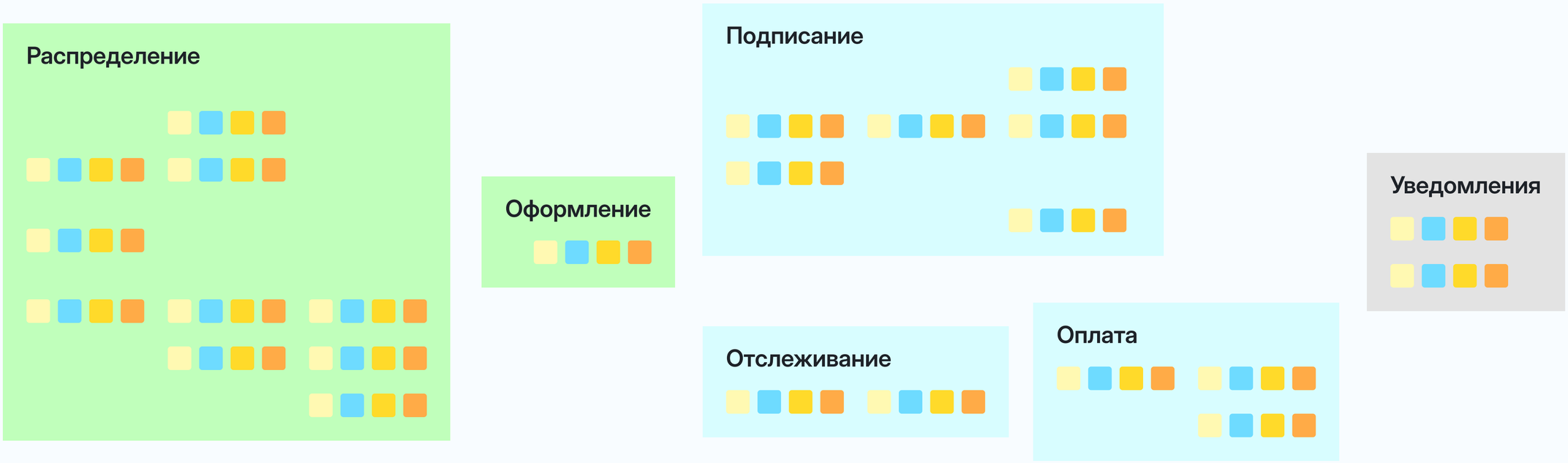
Результаты Event Storming

Мы собираем связанные концепции в **домены**,
которые позволяют разбить монолит на **модули/сервисы**.

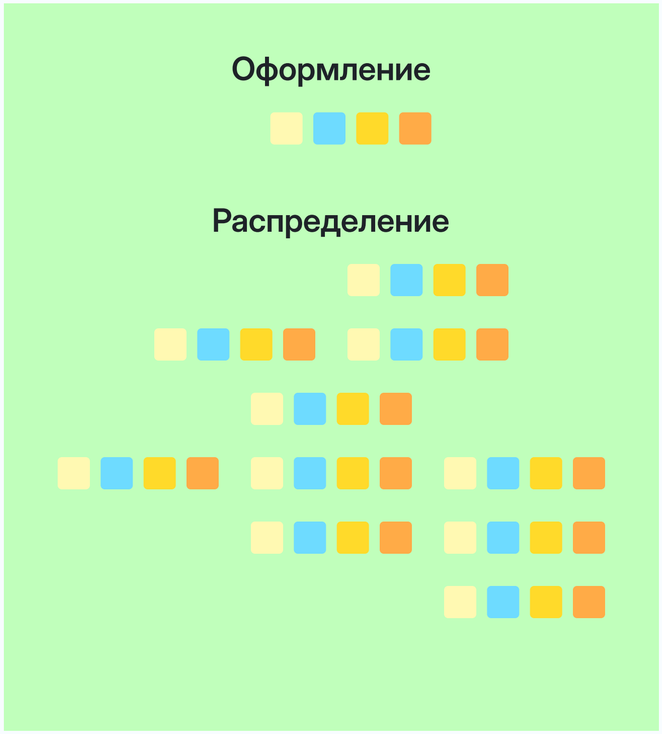


Разбиение домена

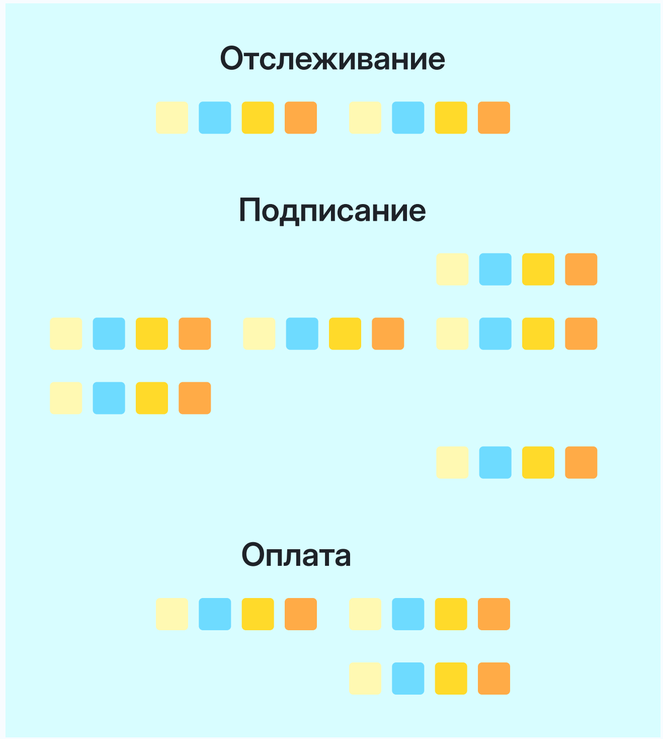
Программные реализации поддоменов являются автономными модулями системы.



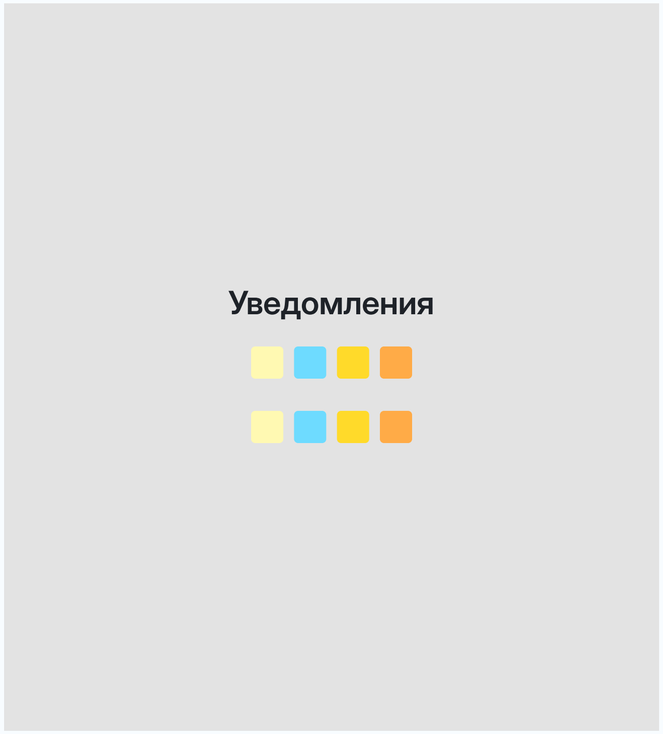
Разбиение домена



Смысловое ядро



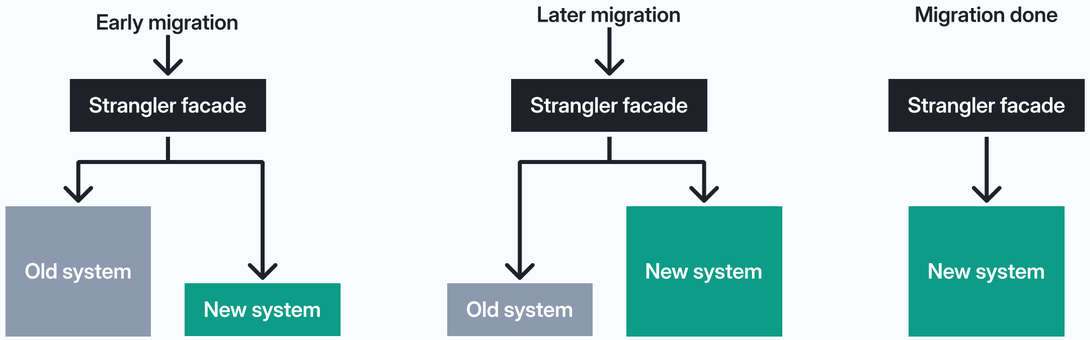
Вспомогательные домены



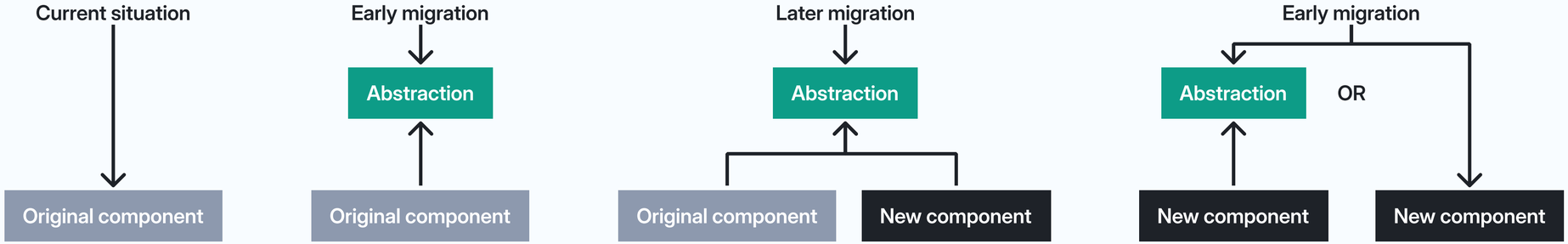
Домены общего назначения

Примеры паттернов

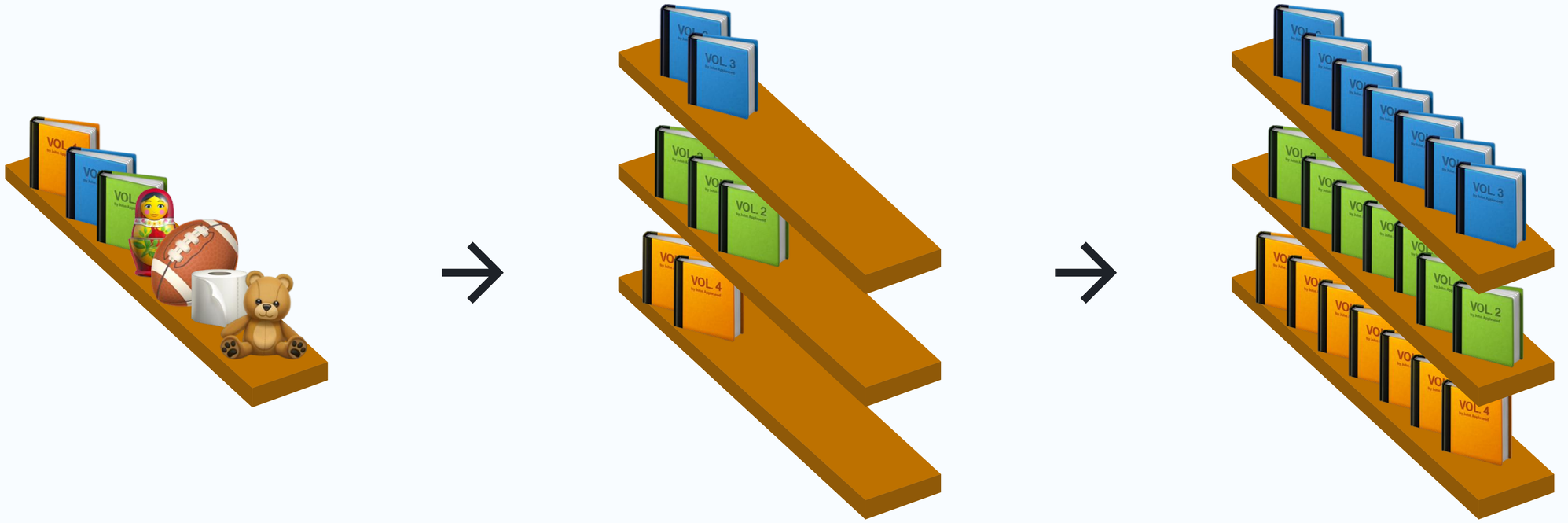
Strangler Fig (Душитель)



Branch by abstraction



Паттерн душитель



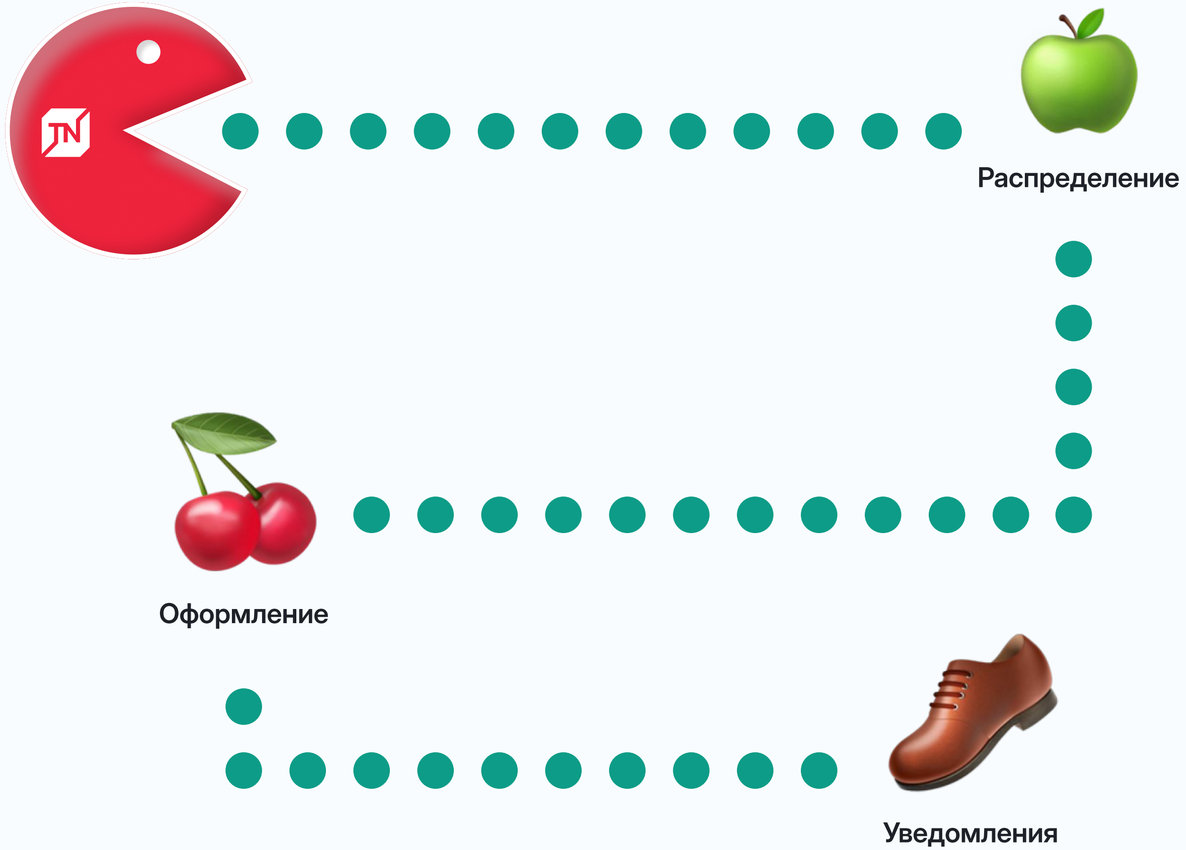
Старый монолит

Переход к модульному монолиту

Модульный монолит

Сохранение экспертизы

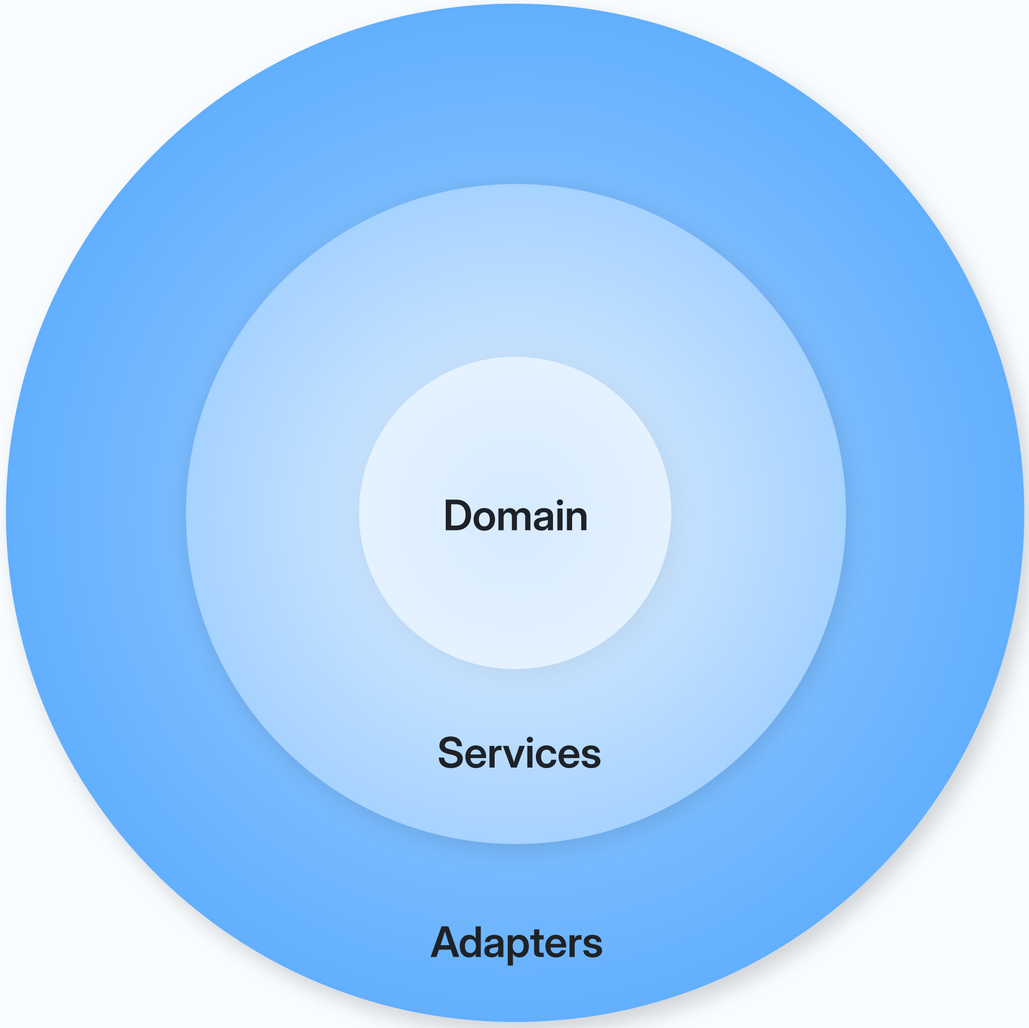
Смысловое ядро — это конкурентное преимущество компании, экспертиза по которому всегда должна оставаться внутри корпорации.



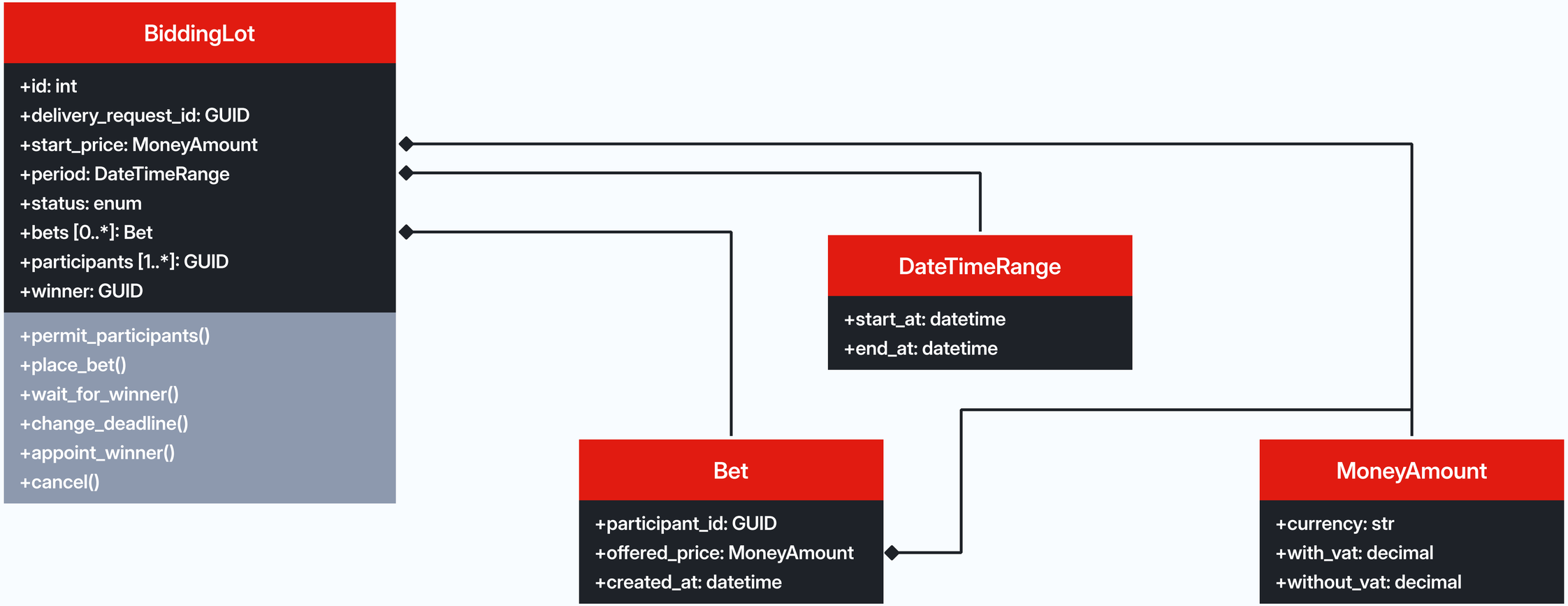
Архитектурная схема

Архитектура домена

Домен разделен на три слоя:
Domain — логика бизнес-сущности
Services — логика приложения
Adapters — инфраструктура



Моделирование бизнес-сущеностей



История проекта

Тише едешь — дальше релиз 😢

Апрель 2023

Проектирование

Август 2023 - Декабрь 2023

Разработка

Декабрь 2023 - 22 марта 2024

Отладка и интеграционное тестирование 1С-Т2-SignTN

9 апреля 2024

Переключение Выборг, Нижний Новгород и Рыбинск на REST

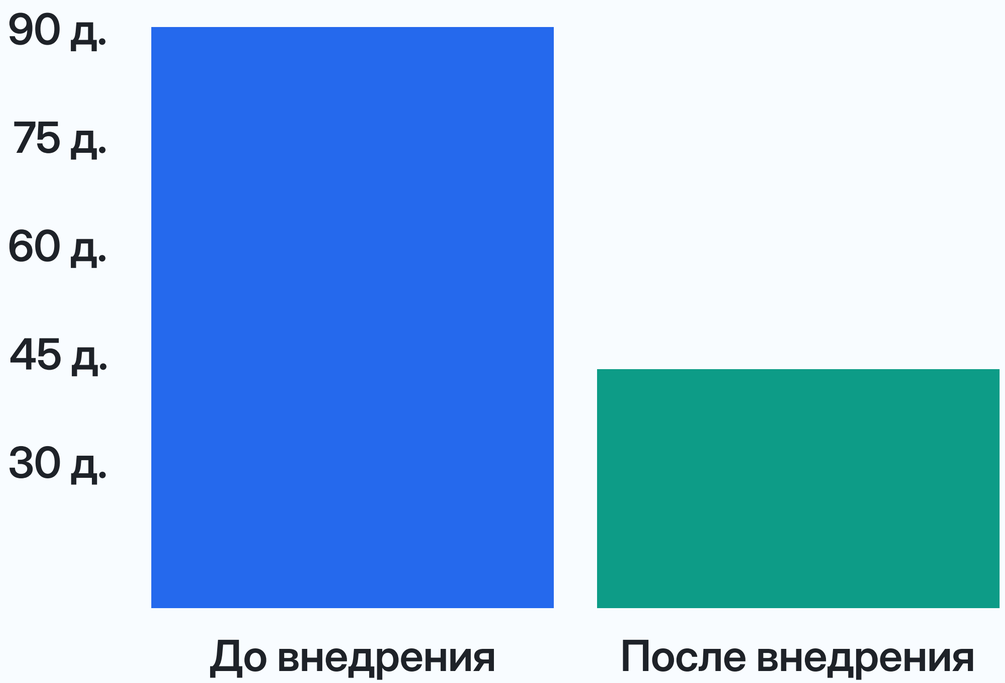
14 мая 2024

Доработка функционала

25 сентября 2024

5 заводов
+ 1 площадка

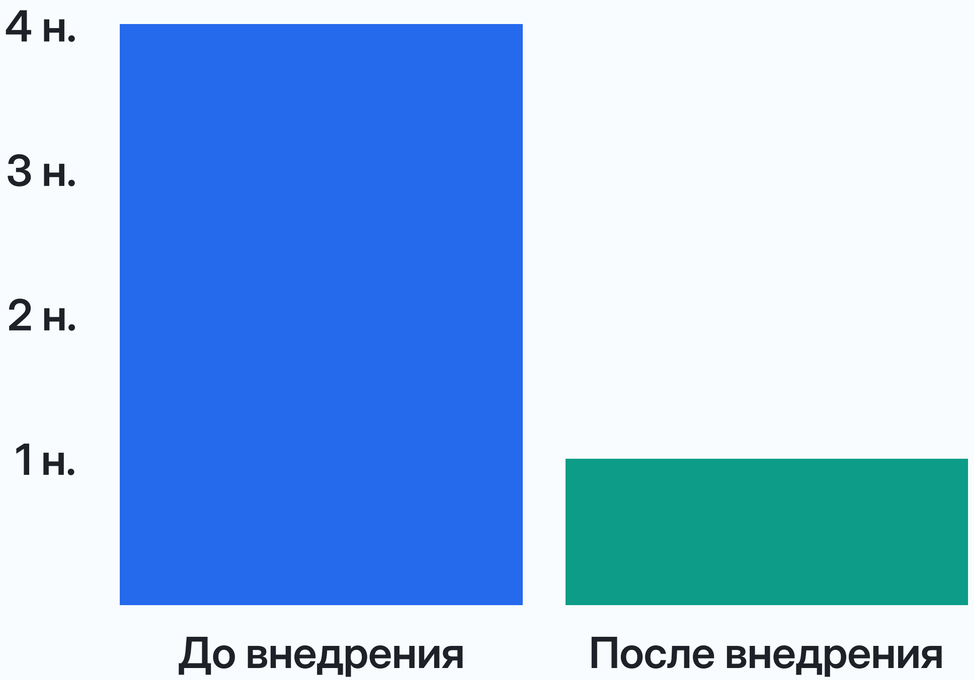
Что мы получили?



Уменьшилось время
разработки фич ЖД
и контейнерных перевозок

В 2 раза

Что мы получили?



Уменьшилось время
разработки счетчиков
в торгах и аукционах

В 4 раза

Что получили от внедрения новой архитектуры и применения DDD:

Бизнес и разработка говорят на одном языке: термины бизнеса находят прямое отражение в программном коде

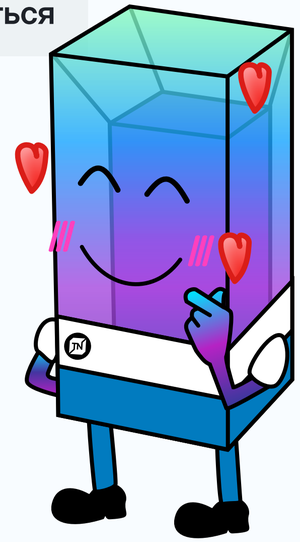
Гибкость в выборе языков программирования, баз данных и фреймворков

Снизилось время на онбординг

Ключевая экспертиза остаётся внутри компании

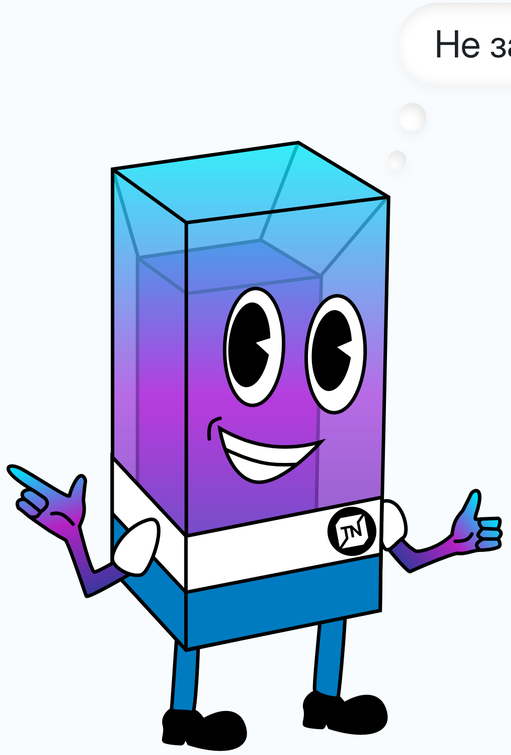
Ускорилась разработка

Разработка стала лучше параллелиться



Статусно-событийная схема сущности

Практика Event Storming



Список endpoints

Архитектурная схема

Стас Балахонов

balahonov@tn.ru +7 996 721-00-33

Альбина Бикбулатова

bikbulatova.a@tn.ru +7 963 155-56-10

ТЕХНОНИКОЛЬ Digital