Т БАНК

Платформа публичных API

Как подойти к разработке и развитию

Каравашкин Павел
Т-API

Павел Каравашкин

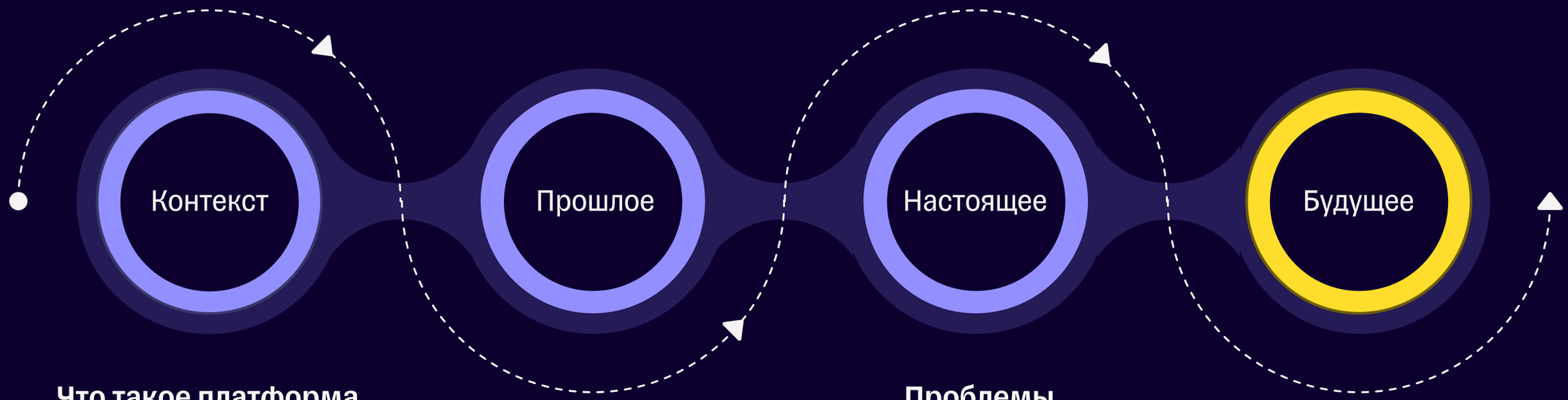
Teamlead платформы T-API

- Лидер профессии системного анализа
- Монолиты (ERP, CRM, SCM, PLM, ESB, BSS, BPMs, WMS, ABS)
- Лекции, преподавание, менторство, подкаст

- В IT 8 лет
- Консалтинг, финтех, производство, логистика, телеком
- В Т-Банк 2,5 года

О чем выступление?

О платформе публичных API Т-Банка



Что такое платформа публичных API

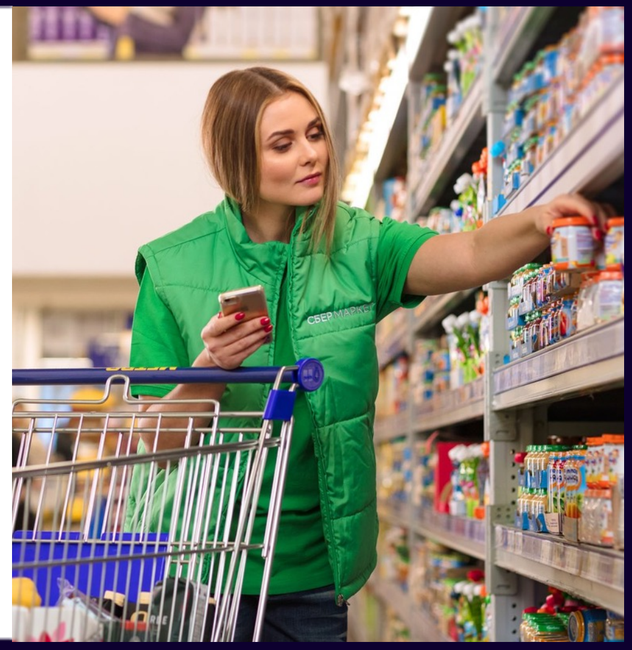
Как появляются платформы

Проблемы
Как их решали
К чему пришли

Будущее
Советы, C2A

Контекст (кейс)

Ашан



Бизнес карты

Карты для компаний,
с гибкими настройками лимитов

Публичное API

Интерфейс, предоставляющий клиентам возможность автоматизировать свою деятельность

Ручки для сайта – открыты, но не декларируемы

API МВД – декларируемый, но не открытый

Публичный – это открыто декларируемый

Мотивация

Нужно API для продукта

- Не нужны велосипеды
- Быстро выпустить продукт

Хотите платформу

- Реальный опыт
- Неочевидные проблемы
- А может не надо?

Просто послушать

- Как строятся платформы
- С какими вызовами сталкиваются

Продукт и платформа

API для продукта

Цель: Заработать

- Имеет определенные ФТ
- Решает потребности клиента
- Конечный результат разработки

Платформа для API

Цель: Обеспечить

- Имеет определенные НФТ
- Предоставляет инструменты для создания продуктов
- Решает потребности продуктовых команд

Начало

Задача

- Хотим публичный API к банковским продуктам
- **Облако вопросов:**

Требования

- Много стейкхолдеров
- Нельзя купить готовое
- Единая точка
- Юр. лица
- Бесплатные вызовы
- Стандартизация
- Открытость для расширения

Какой подход выбрать

- **Load Balancer**

  слишком просто

- **Service Mesh**

  слишком сложно

- **ESB**

  устарело

- **Gateway**

Какой подход выбрать

- **Load Balancer**
  слишком просто
- **Service Mesh**
  слишком сложно
- **ESB**
  устарело
- **Gateway**
  в самый раз

API Gateway

- Надежность (SCALA) и безопасность
- Лучшие практики API
- Человеческое администрирование
- Правила гибкие и простые
- Защищаем от запросов и ответов
- Все функции из коробки (далее)

- Все должны знать правила
- God Object
- Bottleneck
- Медленная доработка

Популярные решения

Kong

Amazon API Gateway

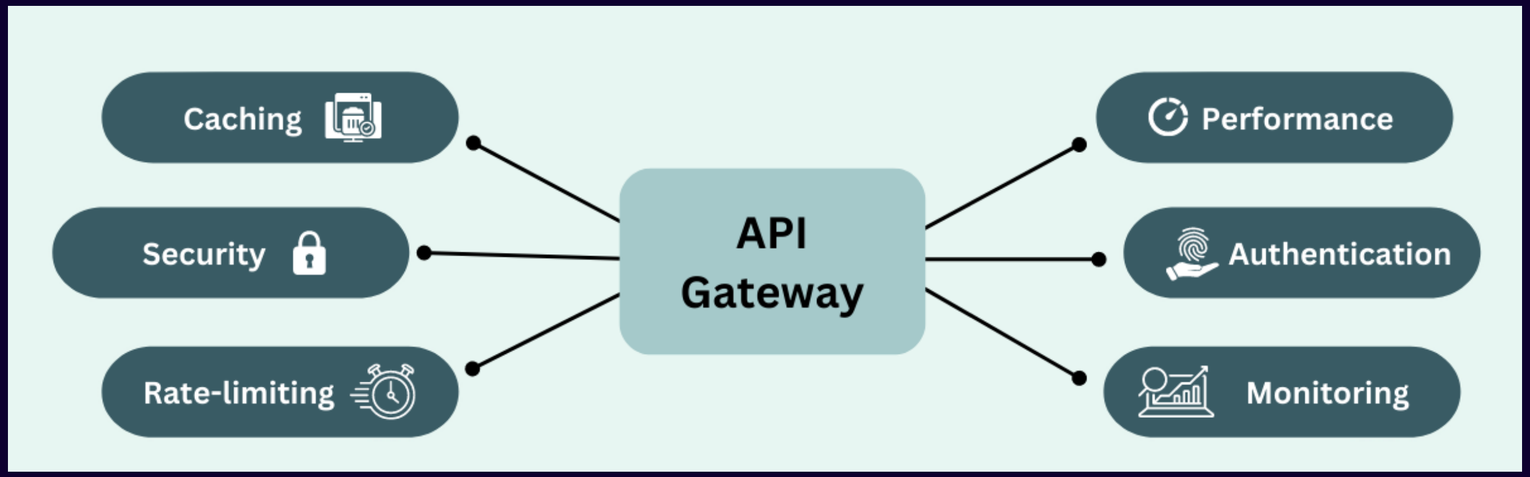
Gravitee.io

WSO2

KrakenD

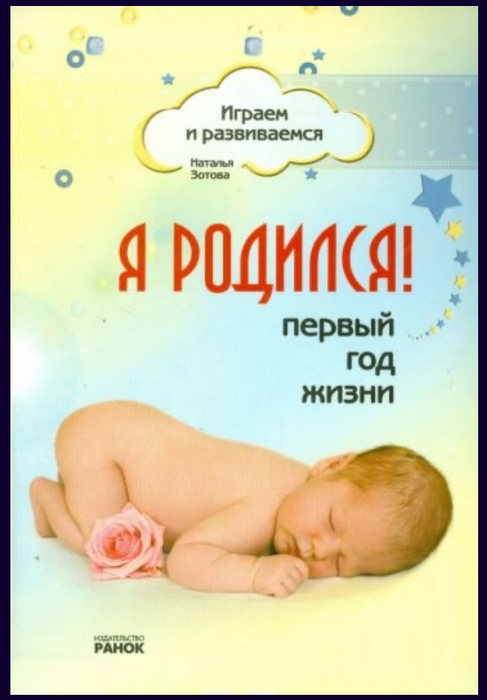
Tyk

Goku

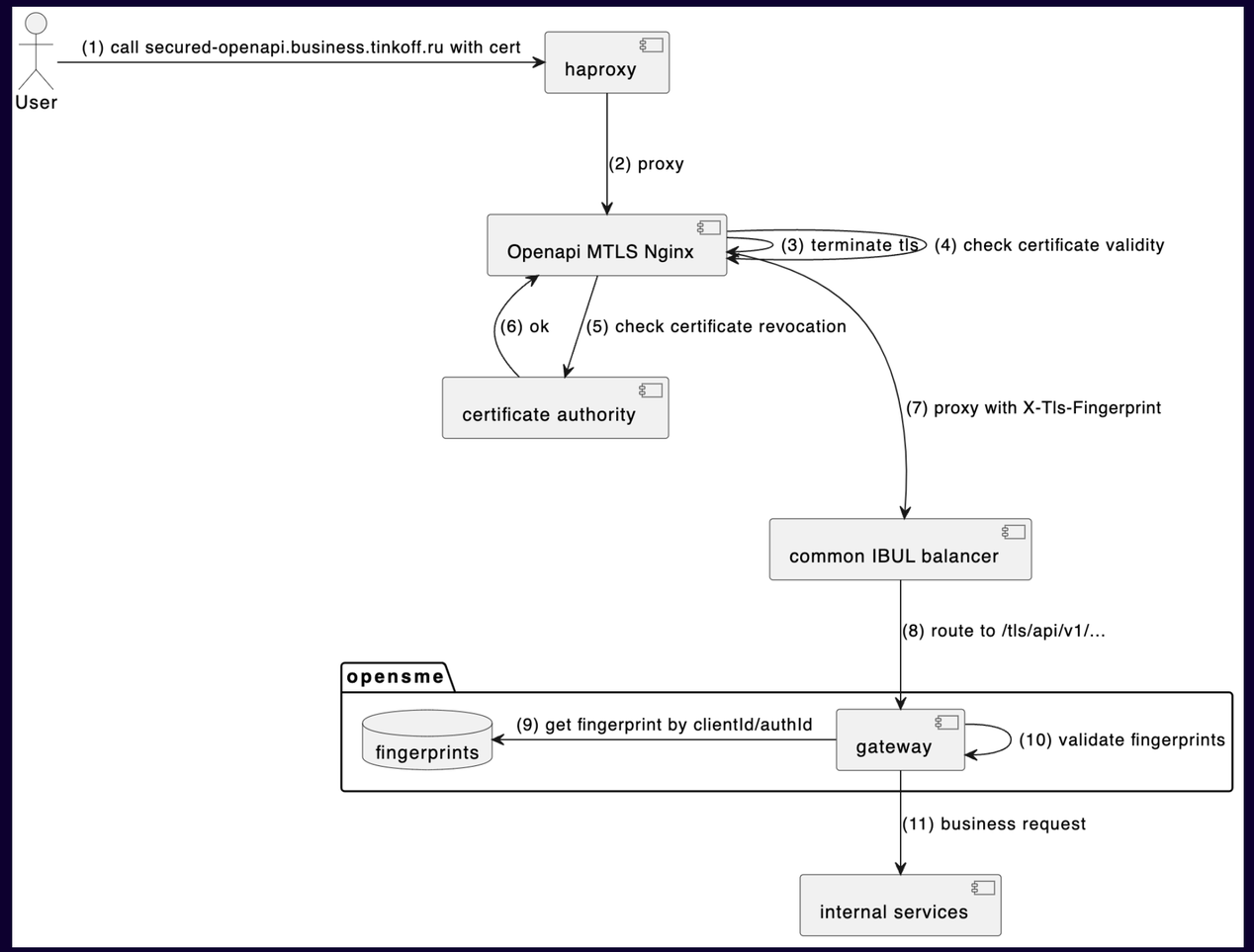


Рождение (2019)

- Монорепа
- SCALA



- mTLS
- ReDoc
- REST JSON



Что делать дальше?

Факапы 😱

1. Как мы сделали Instamart 😰

- Доработки под конкретного клиента
- Разный релизный цикл приложений
- Легаси
- Дополнительная точка наблюдения
- Дополнительная точка отказа

- Клиенту ”очень нужен” туннель
- Избавляемся от влияния клиентов друг на друга

- **Платформа ≠ продукт**

Любое допущение – это отсутствие стратегии

Факапы 😱

2. Как мы закопали и откопали функционал 😰😰

Да кому нужен 1С в 2021 году?

Давайте возьмем чужой модуль

Внезапно много клиентов и доработок

Нехватка разработчиков

Нехватка продуктового управления

Стресс, паника, выгорание

Осознанное владение

Если у вас есть функционал – то вы не должны о нем забывать

Продукт можно закрыть, но не поставить на паузу

Факапы 😱

3. SCALA контракты 😰😰😰

Команды будут описывать контракты на SCALA

Не все смогли

Появилась логика внутри контрактов

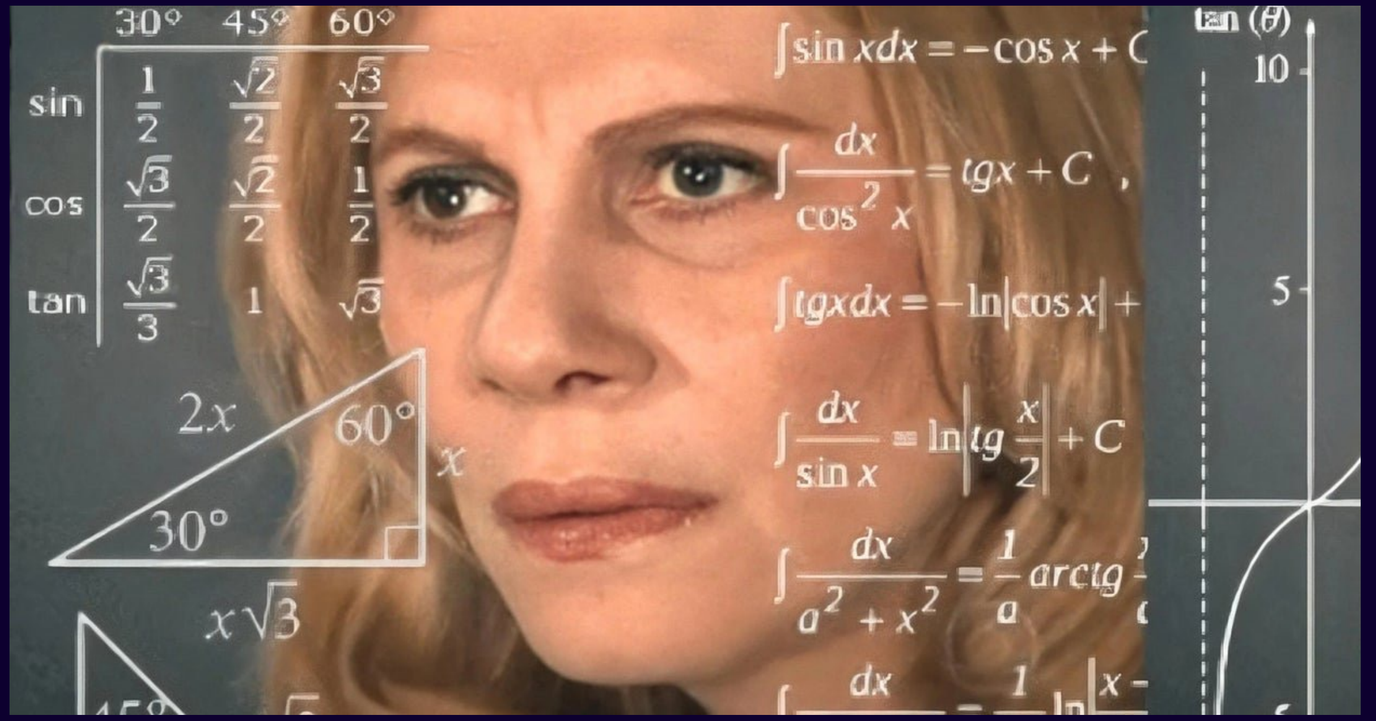
Переделали хорошо

Но логику не переделать

CX/DX станет важен

Неудобная платформа погибнет

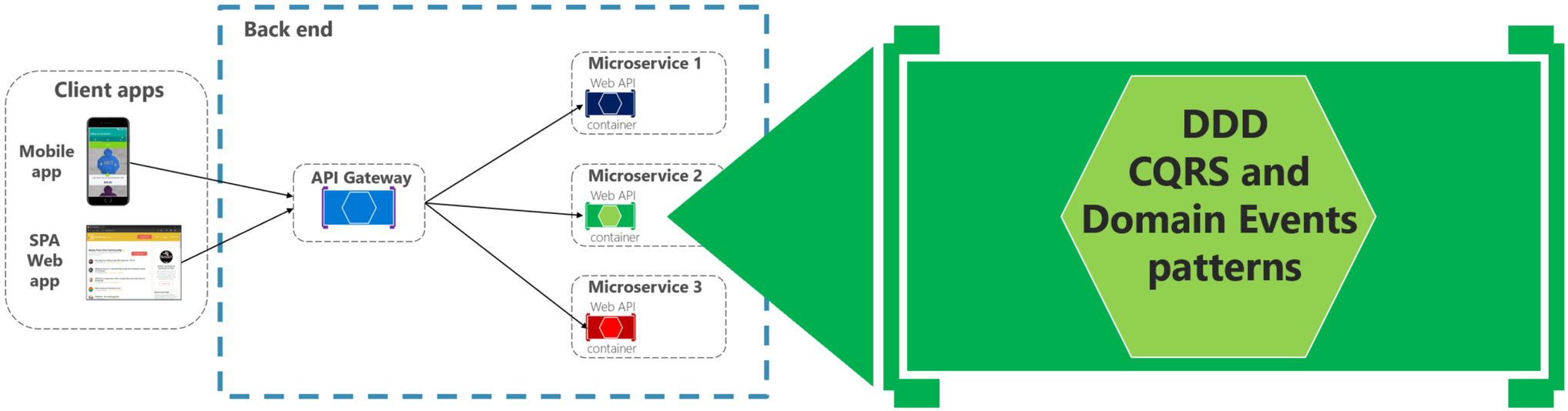
Думаем


30° 45° 60°
sin 1/2 √2/2 √3/2
cos √3/2 √2/2 1/2
tan √3/3 1 √3
2x
60°
30°
x
x√3
∫ sin x dx = −cos x + C
∫ dx / cos² x = tg x + C ,
∫ tg x dx = −ln|cos x| +
∫ dx / sin x = ln|tg x/2| + C
∫ dx / (a² + x²) = 1/a arctg
tan (θ)
10
5

Реархитектура. Попытка в DDD.

External architecture per application

Internal architecture per microservice



- External microservice patterns
- API Gateway
- Resilient communication
- Pub/Sub and event driven

Internal DDD patterns in addition to SOLID principles and Dependency Injection

Требования

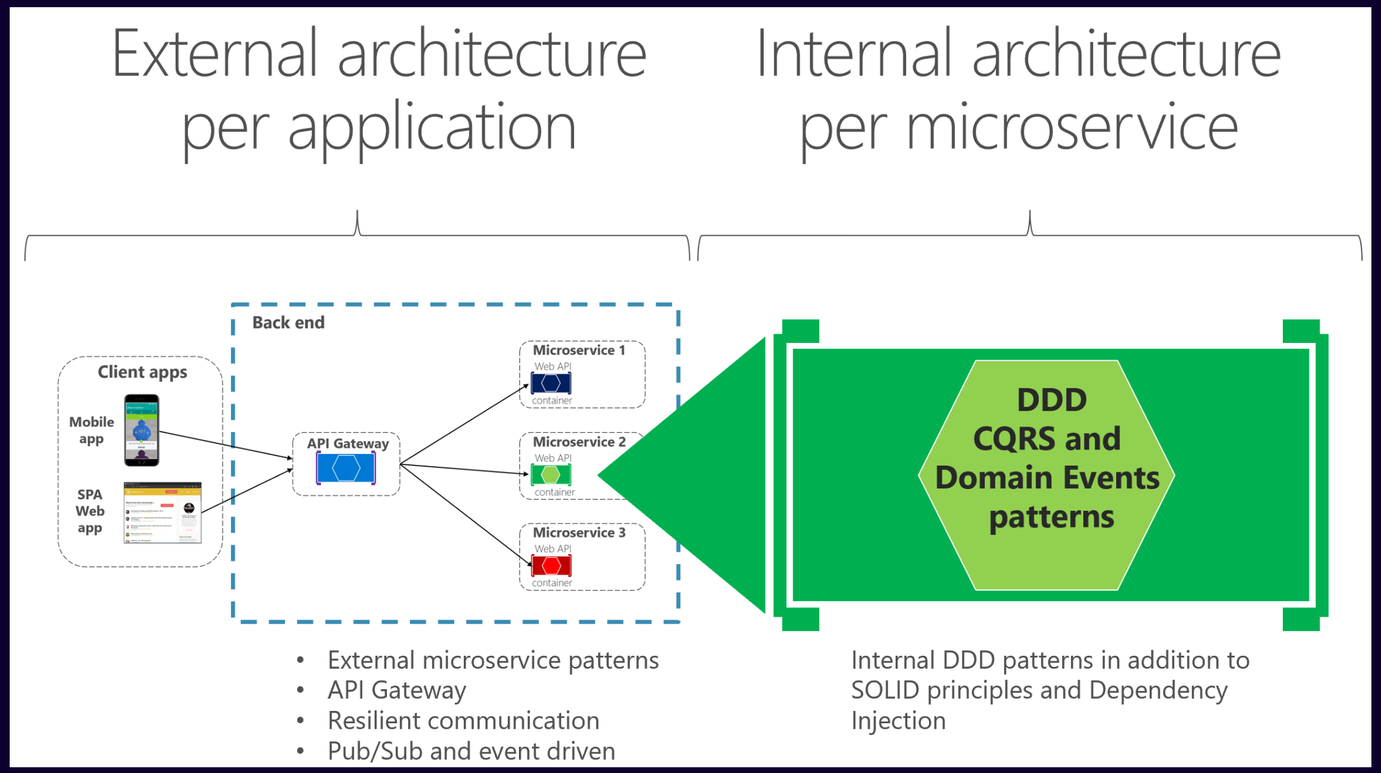
- Много стейкхолдеров
- Нельзя купить готовое
- Единая точка
- Юр. лица
- Бесплатные вызовы
- Стандартизация
- Открытость для расширения

Попытка в DDD

~~Бизнес продукты~~

Не получилась

Домена нет



Хотим универсальное решение

«Третий интерфейс Т-Банк»

UI/UX/CX/DX

Вопрос: **кто** и что делает в вашей системе?

Владельцы

Продакты

ЛПР

Sec

Стейкхолдеры

Технологи

SDE

Разные языки

Поддержка

DevOps

QA

Интеграторы

SA

Архитекторы

Продажи

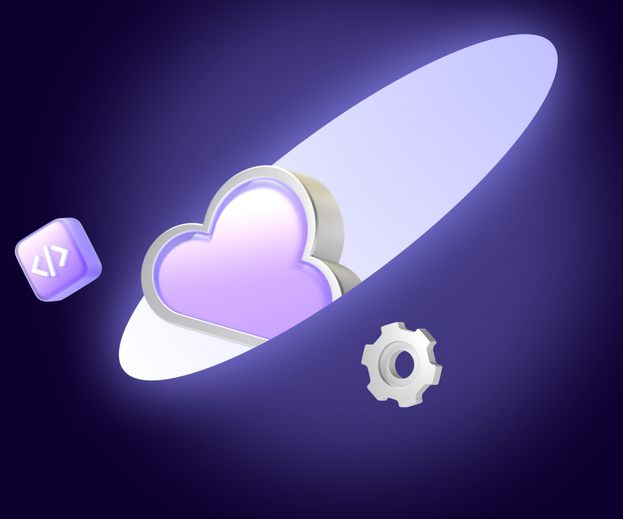
Маркетинг

UI/UX/CX/DX

Вопрос: кто и **что делает** в вашей системе?

Клиент

- **Как-то нас находит**
- **Принимает решение об интеграции**
- **Читает документацию**
- **Пишет код (но хочет SDK)**
- **Тестирует**
- **Запускается**



UI/UX/CX/DX

Вопрос: кто и **что делает** в вашей системе?

Продуктовая команда

- **Как-то нас находит (но возможно мы ее)**
- **Думает о бизнес смысле**
- **Реализует контракт**
- **Пишет код**
- **Заезжает на платформу**
- **Согласует все со всеми**
- **Меняет контракты**
- **Уведомляет клиентов**

Выбираем подход к API

Предпосылки: API становится значимым интерфейсом

Code First

Приложение – это продукт

Сначала создается приложение, затем API

+ Традиционный подход
+ Легко начать
+ Можно менять код без API
– Проблемы масштабирования
– Проблемы документации

API First

API – это продукт

Все общение через API Любые изменения сначала в API

Пользователь: внешний

+ Прозрачность архитектуры
+ Масштабируемость
+ Параллельная разработка
– Сложность ЖЦ
– Потенциальное отсутствие документации

API Design First

API + документация – это продукт

Одна документация для человека и машины

Пользователь: внутренний

+ Прозрачность для заказчика
+ Качество продукта
+ Стандартизация инструментов
– Сложность внедрения подхода

API Driven Development

Платформа API – это продукт

Фокус на DX

Автоматизация Dev, QA, CI/CD, SDK

Пользователь: все

+ Наивысшая автоматизация
+ Качество и надежность
– Сложность внедрения подхода
– По сути отдельная система

Черпаем мудрость

Сообщества

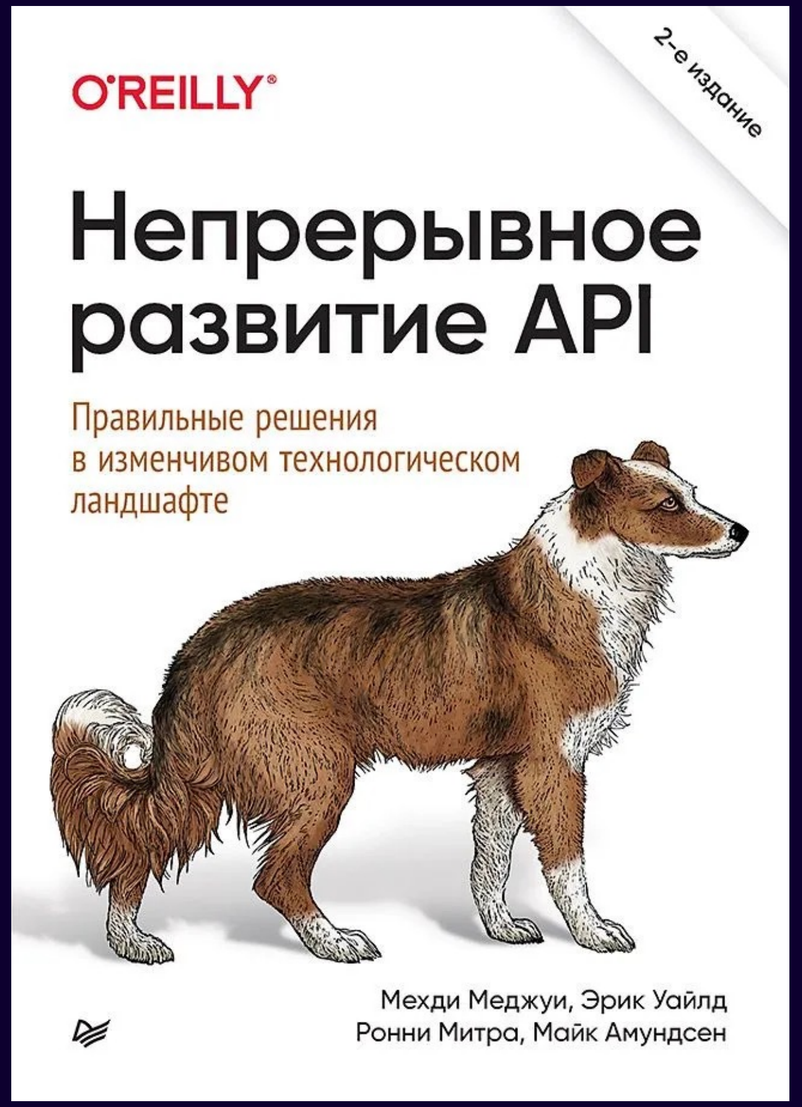
Успешные кейсы

”Устав Безоса” – AWS, начало 200х

https://tcrn.ch/2R5dh9n

Непрерывное развитие – 10 столпов

1. Стратегия
2. Дизайн
3. Документация
4. Разработка
5. Тестирование
6. Развертывание
7. Безопасность
8. Мониторинг
9. Обнаружение
10. Управление изменениями

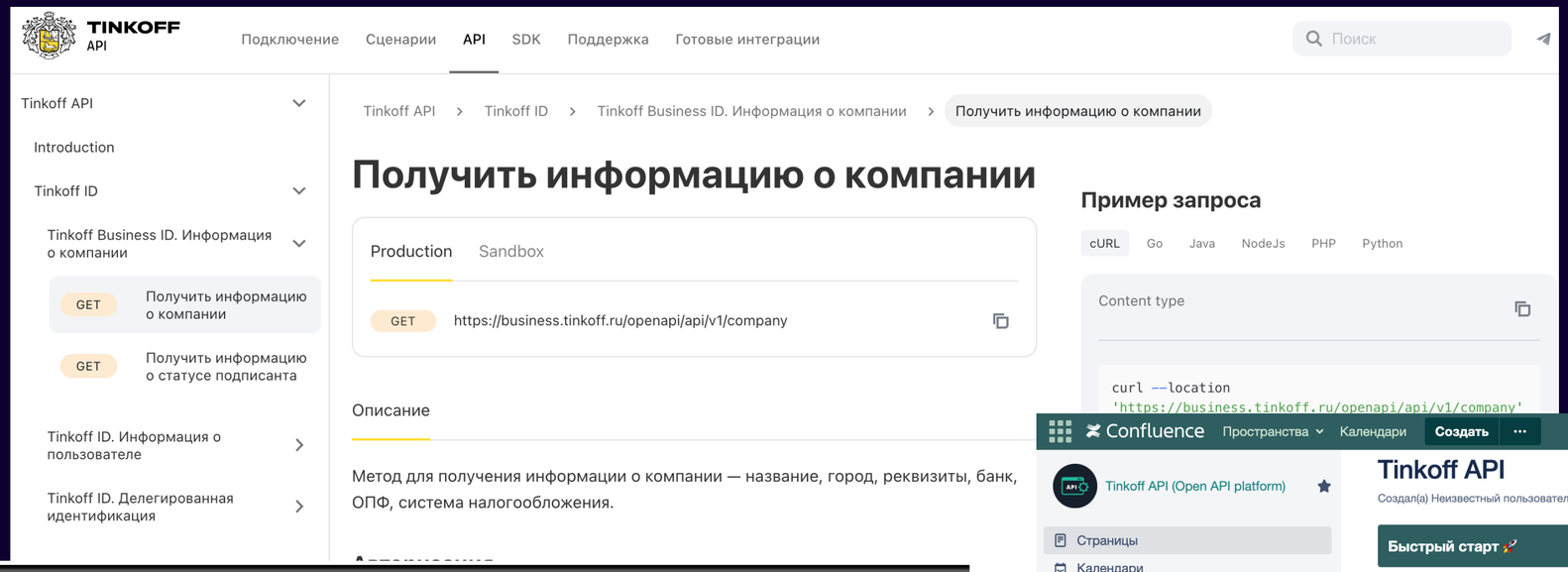


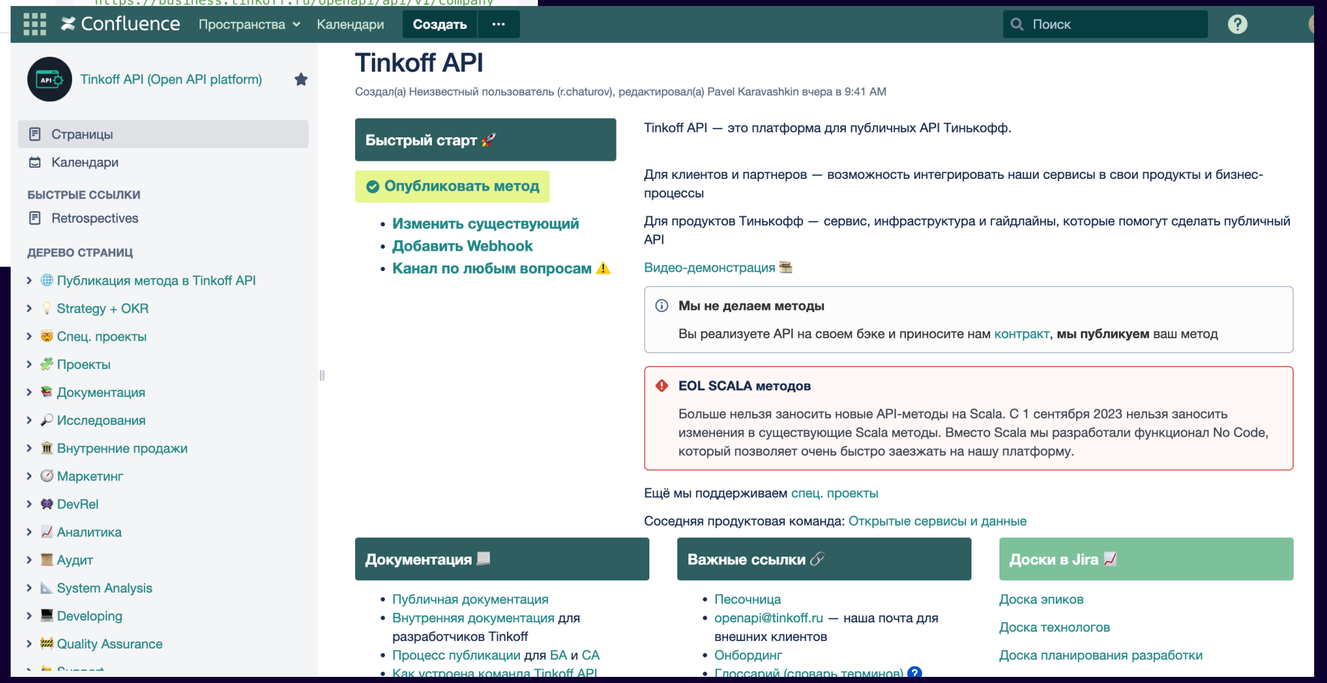
Итоги. Услуги платформы

Итоги. Услуги платформы

1. Стратегия
2. Дизайн
3. Документация
4. Разработка
5. Тестирование
6. Развертывание
7. Безопасность
8. Мониторинг
9. Обнаружение
10. Управление изменениями

Документация – это база





TW + SA + QA
это сила!

Неочевидные кейсы (незаданные вопросы)

- Версионирование API
- Общие классы и дубли классов
- Сценарии совместного использования API
- Огромные тела
- Скрытые методы
- Release Notes
- Песочница, вызовы с фронта
- Платные методы
- Мониторинг
- Утилизация API
- YAML конфиг > 1мб

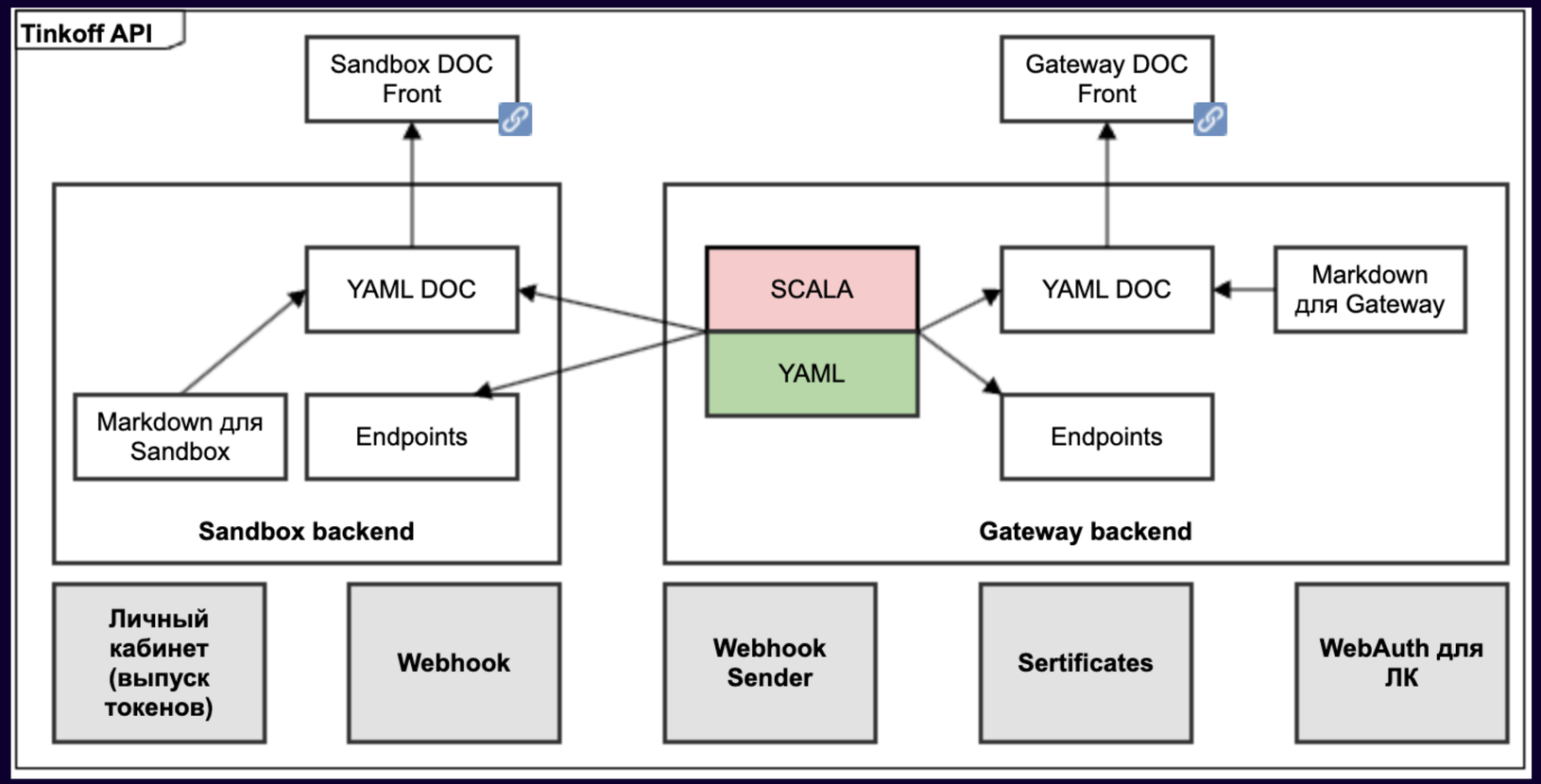
Факты

- **Выпуск mTLS сертификата – 12 минут**
- **Публикация нового метода – от 1 суток**
- 230+ методов, 24+ продуктов
- **>10 топ корп. клиентов**
- 50к Клиентов (ЮЛ)
- 4кк запросов в сутки, пиковый RPS 500
- Фактический SLI 99,95%

Т БАНК

Лучшая API платформа

Контракты – это данные



- CRUD
- Ролевая модель
- Принадлежат продуктам
- Имеют мета параметры
- Понятный формат
- Понятная структура

AS-IS: YAML в GitLab / TO-BE: Структуры в БД + API MS

Принципы

OpenAPI

Мировой формат описания и проектирования API

APIv3

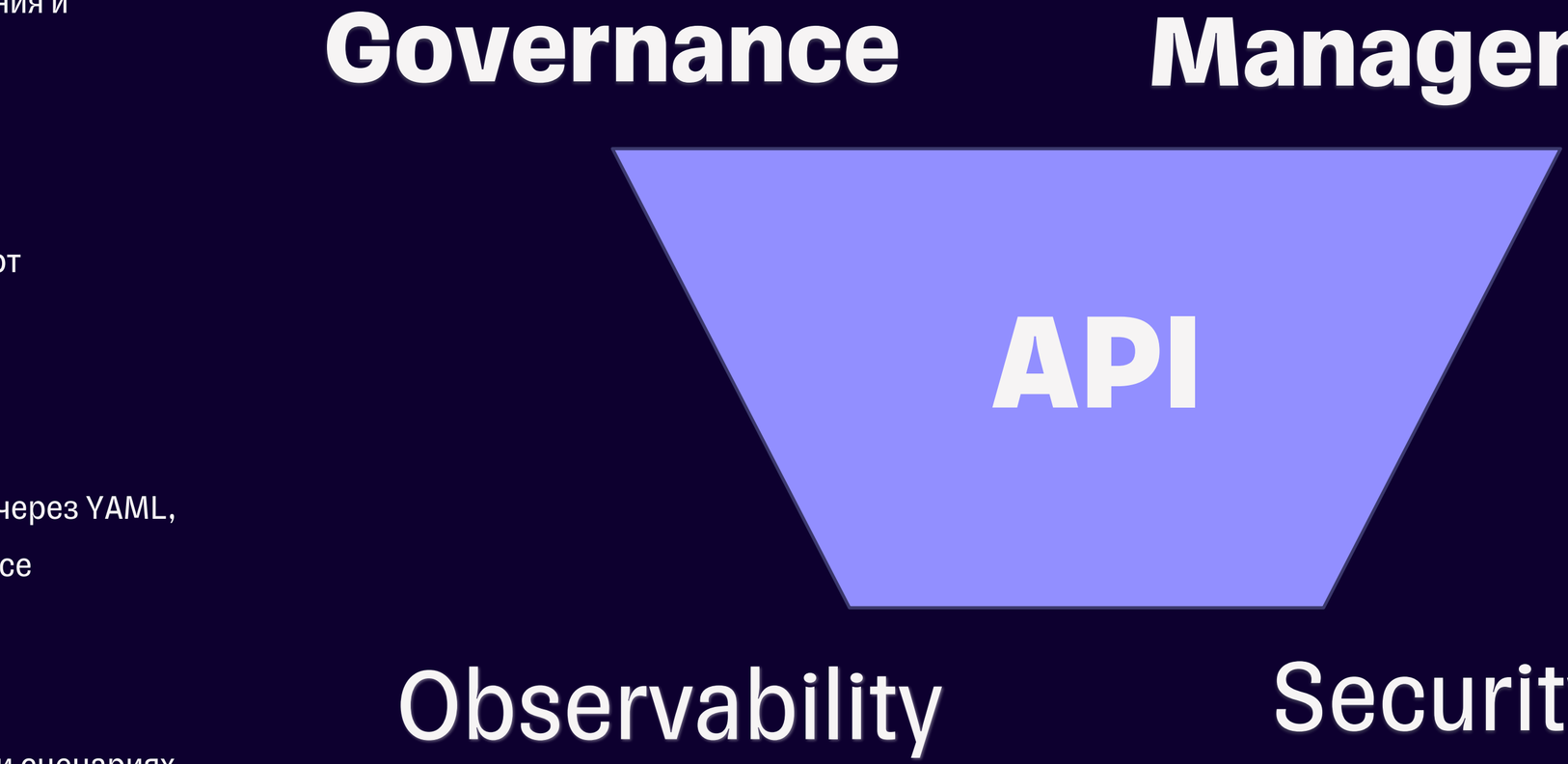
Корпоративный стандарт

NoCode

Без знания языков, все через YAML, максимальный SelfService

Design API First

Фокусировка на UX, CX и сценариях использования, а не на реализации

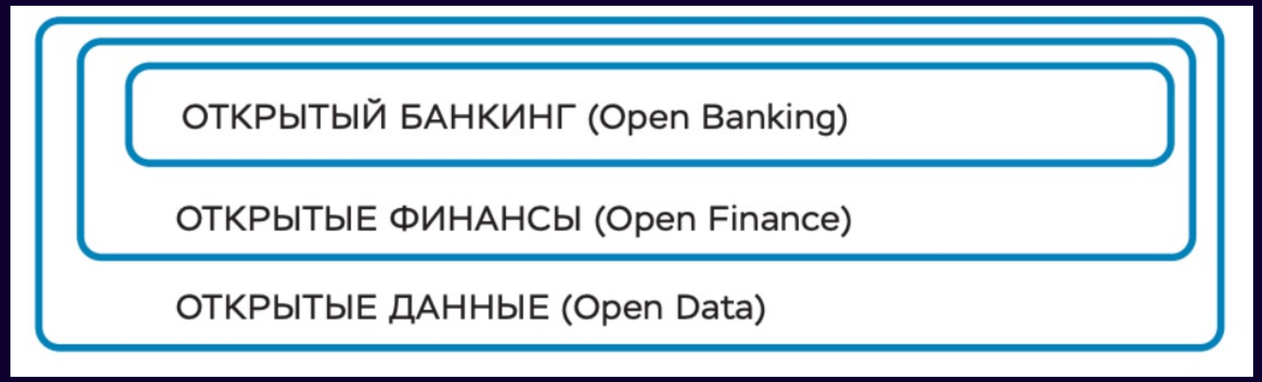


Планы (до конца 2025)

- ЛК Разработчика
- API Management System
- Бандлы и синергия продуктов
- Inner Source -> Open Source
- DevRel и PR

Создание команды технического развития

Extra: Что говорит ЦБ



- **Открытые API** – публично утвержденные программные интерфейсы для обмена между финтехами

- **Открытый банкинг** – модель обмена банковскими и платежными данными
- **Открытые финансы** – еще страховые, пенсионные, инвестиционные и пр. данные
- **Открытые данные** – расширение сфер участников модели + госы, бюджетники, ОпСОСы и пр.

https://openbankingrussia.ru/

«В идеале у нас должна быть система, когда владелец данных о себе – сам человек или бизнес – может перемещать свои данные свободно из одной организации в другую.»

Эльвира Набиуллина

Председатель Центрального банка Российской Федерации

ПЕРЕЙТИ К ИСТОЧНИКУ

Три вывода

- Сообщество – это благо, вы не одни
- Продукт ≠ Платформа
- API – полноценный интерфейс? Значит API First!

Одно действие

Делаешь платформу? Прочитай книгу!

Вопросы?

Павел Каравашкин

Teamlead T-API

p.karavashkin@tbank.ru

@pkaravashkin

developer.tbank.ru

Скоро будет профильный Т-Банк SA канал...и кое-что еще!