YATALKS

Как формировать структуру команд под запросы бизнеса

Александр Поломодов
Технический директор,
Тинькофф

YfD */

TINKOFF

План нашей беседы

- Ключевые подходы

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф
  - Старт новой инициативы

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф
  - Старт новой инициативы
  - Развитие продукта

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф
  - Старт новой инициативы
  - Развитие продукта
  - Масштабирование разработки вверх (scaling up)

План нашей беседы

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф
  - Старт новой инициативы
  - Развитие продукта
  - Масштабирование разработки вверх (scaling up)
  - Масштабирование разработки вниз (scaling down)

Ключевые подходы

YfD */

Ключевые подходы:

Backcasting

Backcasting is a planning method that starts with defining a desirable future and then works backwards to identify policies and programs that will connect that specified future to the present.

Ключевые подходы:

YfD */

Project

Project is a temporary endeavor undertaken to create a unique product, service or result

Ключевые подходы:

Kanban

Kanban is a lean method to manage and improve work across human systems. This approach aims to manage work by balancing demands with available capacity, and by improving the handling of system-level bottlenecks.

Work items are visualized to give participants a view of progress and process, from start to finish—usually via a kanban board. Work is pulled as capacity permits, rather than work being pushed into the process when requested.

Ключевые подходы:

Conway's Law

Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.

Conway's Law

Ключевые подходы:

Inverse Conway Maneuver

> The 'Inverse Conway Maneuver' recommends evolving your team and organizational structure to promote your desired architecture. Ideally your technology architecture will display isomorphism with your business architecture.

Ключевые подходы:

Team topologies

Team Topologies is a clear, easy-to-follow approach to modern software delivery with an emphasis on optimizing team interactions for flow.

Four fundamental types of team and three core team interaction modes combine with awareness of Conway's Law, team cognitive load, and responsive organization evolution to define a no-nonsense, team-friendly, humanistic approach to building and running software systems.

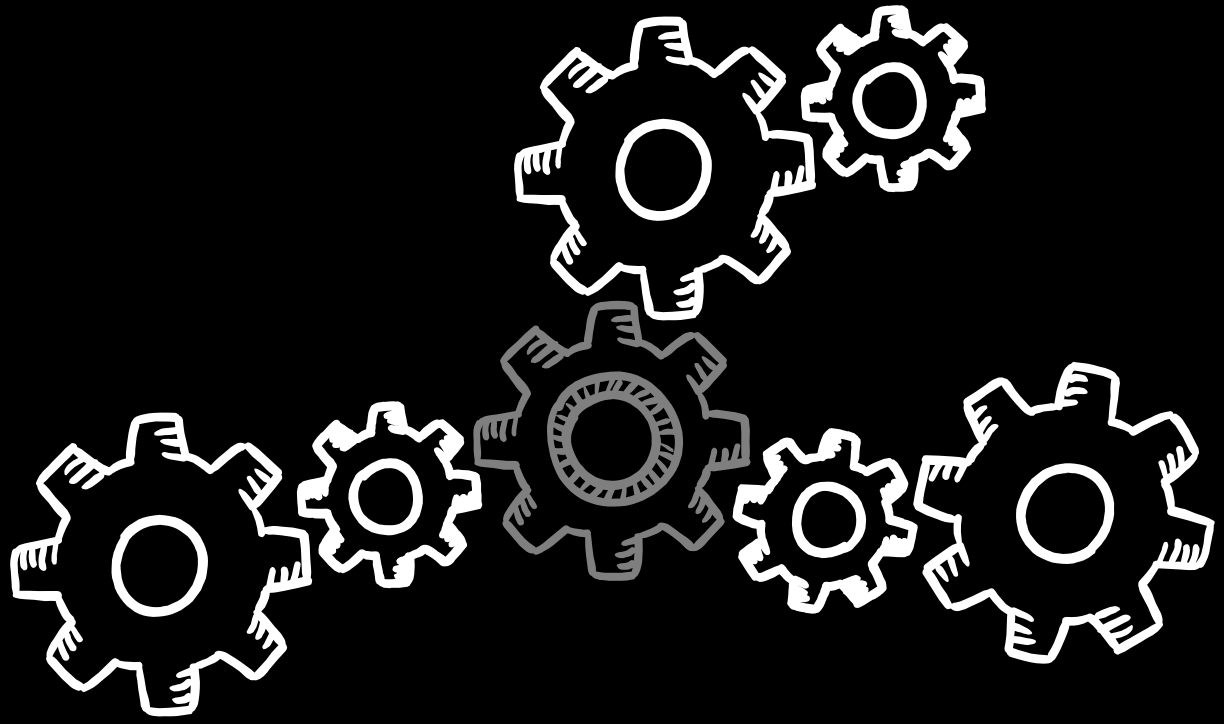
Ключевые подходы:

Change management

Change management is a systematic approach to dealing with the transition or transformation of an organization's goals, processes or technologies. The purpose of change management is to implement strategies for effecting change, controlling change and helping people to adapt to change.

Алгоритм

YfD */



Алгоритм

- Определить долговременную цель

Алгоритм

- Определить долговременную цель
- Понять целевую архитектура систем

Алгоритм

- Определить долговременную цель
- Понять целевую архитектура систем
- Применить обратный маневр Конвея

Алгоритм

- Определить долговременную цель
- Понять целевую архитектура систем
- Применить обратный маневр Конвея
- Использовать паттерны из team topologies

Алгоритм

- Определить долговременную цель
- Понять целевую архитектура систем
- Применить обратный маневр Конвея
- Использовать паттерны из team topologies
- Использовать менеджмент изменений, чтобы все спланировать и осуществить

Алгоритм

- Определить долговременную цель
- Понять целевую архитектуру систем
- Применить обратный манёвр Конвея
- Использовать паттерны из team topologies
- Использовать менеджмент изменений, чтобы все спланировать и осуществить

Nope!

Алгоритм

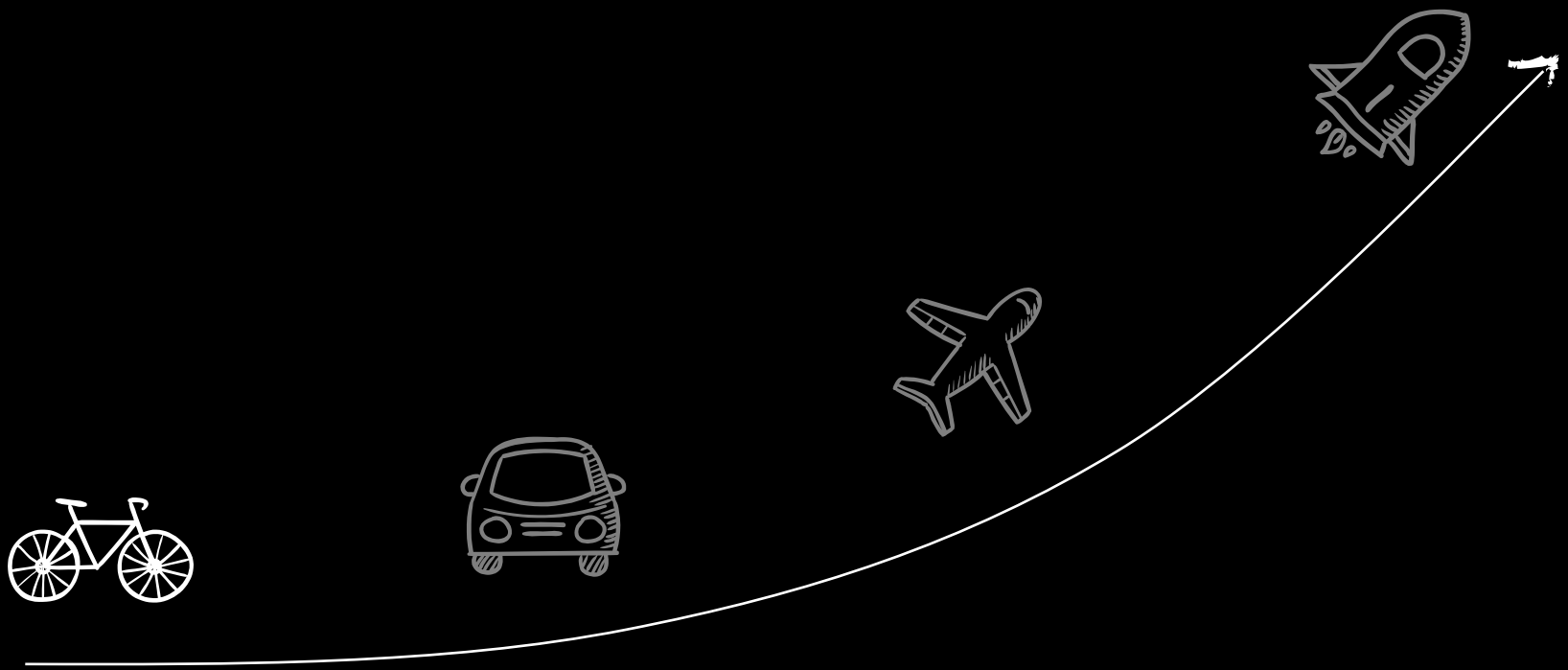
- Определить где горит и сначала починить
- Определить долговременную цель
- Понять целевую архитектура систем
- Применить обратный маневр Конвея
- Использовать паттерны из team topologies
- Использовать менеджмент изменений, чтобы все спланировать и осуществить

Примеры

1

Пример: проект

YfD */



1

Дано: переезд на новую версию Tinkoff.ru

YfD */

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы
  - Не работал SSR

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы
  - Не работал SSR
  - Не работала аналитика

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы
  - Не работал SSR
  - Не работала аналитика
  - Не работали многие механики на формах привлечения

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы
  - Не работал SSR
  - Не работала аналитика
  - Не работали многие механики на формах привлечения
  - Релизы занимали ~3 недели из-за ручного регресса

1 Дано: переезд на новую версию Tinkoff.ru

YfD */

- Разработка нового решения шла больше года
- Я пришел отвечать за привлечение: формы и лендинги
- Общая команда 50+ человек (3 человека на привлечение)
- До переезда оставалось 3 месяца и было 4 проблемы
  - Не работал SSR
  - Не работала аналитика
  - Не работали многие механики на формах привлечения
  - Релизы занимали ~3 недели из-за ручного регресса
- Переезд без решения проблем стоил бы компании сотни миллионов рублей допзатрат на привлечение

1 Решение: используем проектный подход

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)
  - Отдельная инфраструктура под app привлечения

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)
  - Отдельная инфраструктура под app привлечения
  - Собственный подход к работе с кодом (релиз из своей замороженной ветки + редкий мердж общих изменений)

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)
  - Отдельная инфраструктура под app привлечения
  - Собственный подход к работе с кодом (релиз из своей замороженной ветки + редкий мердж общих изменений)
- Костыльные решения на аналитику

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)
  - Отдельная инфраструктура под app привлечения
  - Собственный подход к работе с кодом (релиз из своей замороженной ветки + редкий мердж общих изменений)
- Костыльные решения на аналитику
- ASAP перенос основных механик со старых форм

1 Решение: используем проектный подход

YfD */

- Собираем проектную команду и фиксируем план того, что надо доделать
- Короткие итерации и частые релизы (2 раза в неделю)
  - Отдельная инфраструктура под app привлечения
  - Собственный подход к работе с кодом (релиз из своей замороженной ветки + редкий мердж общих изменений)
- Костыльные решения на аналитику
- ASAP перенос основных механик со старых форм
- SEO проблемы приняли как данность

1 Итоги

YfD */

1 Итоги

YfD */

- К переезду успели перевезти основные формы и страницы ± без потери эффективности

1 Итоги

YfD */

- К переезду успели перевезти основные формы и страницы ± без потери эффективности
- Команда научилась решать проблемы

1 Итоги

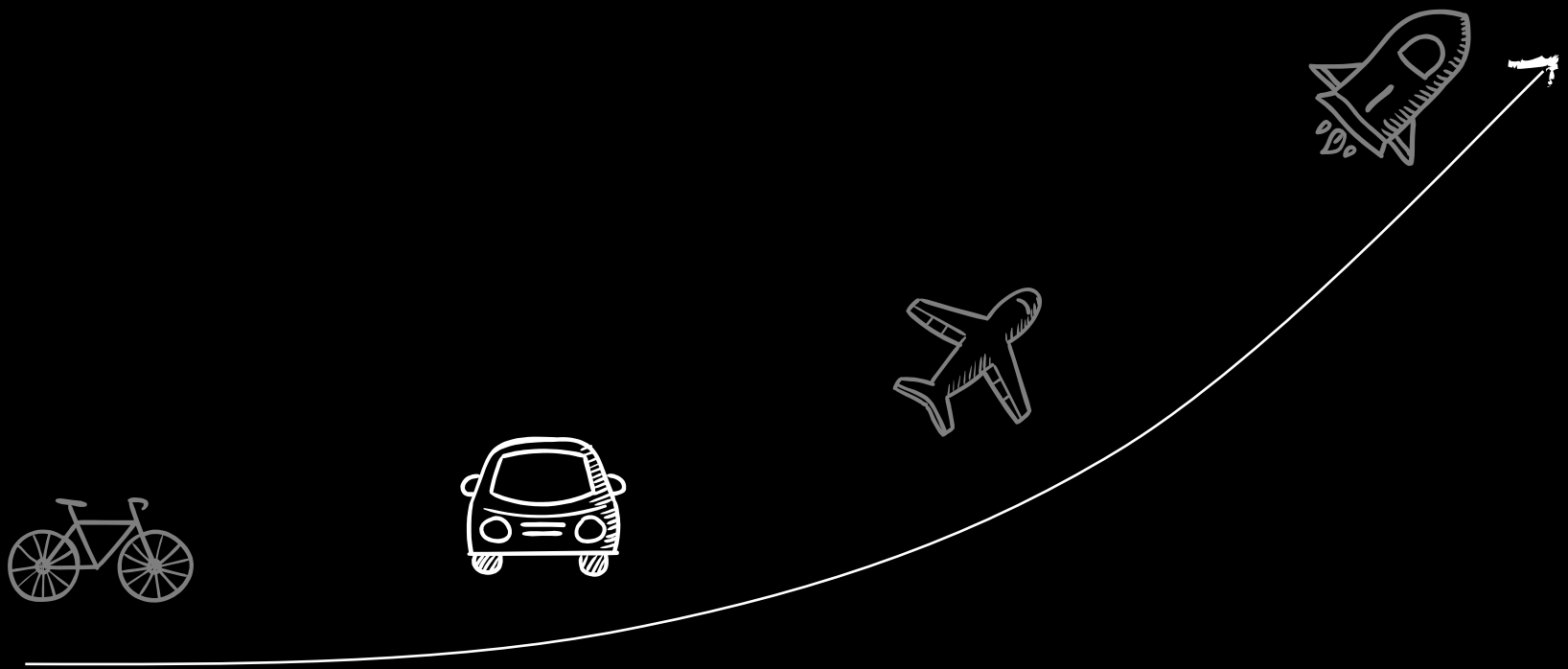
YfD */

- К переезду успели перевезти основные формы и страницы ± без потери эффективности
- Команда научилась решать проблемы
- Получившийся продукт готов к масштабированию

2

Пример: продукт

YfD */



2

Дано: переход к планомерному развитию онлайн-привлечения

YfD */

2 Дано: переход к планомерному развитию онлайн-привлечения

YfD */

- Основные формы и страницы запущены на новом движке

2 Дано: переход к планомерному развитию онлайн-привлечения

YfD */

- Основные формы и страницы запущены на новом движке
- Долговременные цели

2 Дано: переход к планомерному развитию онлайн-привлечения

YfD */

- Основные формы и страницы запущены на новом движке
- Долговременные цели
  - Нужно перенести оставшиеся страницы и формы на новые рельсы

2 Дано: переход к планомерному развитию онлайн-привлечения

- Основные формы и страницы запущены на новом движке
- Долговременные цели
  - Нужно перенести оставшиеся страницы и формы на новые рельсы
  - Нужна система server-driven UI

2 Дано: переход к планомерному развитию онлайн-привлечения

- Основные формы и страницы запущены на новом движке
- Долговременные цели
  - Нужно перенести оставшиеся страницы и формы на новые рельсы
  - Нужна система server-driven UI
  - Нужна система для проведения экспериментов

2 Дано: переход к планомерному развитию онлайн-привлечения

- Основные формы и страницы запущены на новом движке
- Долговременные цели
  - Нужно перенести оставшиеся страницы и формы на новые рельсы
  - Нужна система server-driven UI
  - Нужна система для проведения экспериментов
  - Нужна система для склейки продуктовых и рекламных страниц

2 Дано: переход к планомерному развитию онлайн-привлечения

- Основные формы и страницы запущены на новом движке
- Долговременные цели
  - Нужно перенести оставшиеся страницы и формы на новые рельсы
  - Нужна система server-driven UI
  - Нужна система для проведения экспериментов
  - Нужна система для склейки продуктовых и рекламных страниц
- Примерная схема взаимодействия систем дальше

User

YfD */

User Browser

Tinkoff.ru

User

Browser

Load balancing

Tinkoff.ru

YfD */

User

Browser

Load balancing

Frontend

Tinkoff.ru

Url: Tinkoff.ru

User

Browser

Load balancing

Frontend

Server-driven UI API

Tinkoff.ru

Url: Tinkoff.ru

Get content for Tinkoff.ru

User

Browser

Load balancing

Frontend

Server-driven UI API

A/b platform

YfD */

Tinkoff.ru

Url: Tinkoff.ru

Get content for Tinkoff.ru

Get a/b test for Tinkoff.ru

User

Browser

Load balancing

Frontend

Server-driven UI API

A/b platform

YfD */

Tinkoff.ru

Url: Tinkoff.ru

Get content for Tinkoff.ru

Get a/b test for Tinkoff.ru

Do a/b test magic

User | Browser | Load balancing | Frontend | Server-driven UI API | A/b platform

Tinkoff.ru

Url: Tinkoff.ru

Get content for Tinkoff.ru

Get a/b test for Tinkoff.ru

Do a/b test magic

a/b test for page Tinkoff.ru

User

Browser

Load balancing

Frontend

Server-driven UI API

A/b platform

YfD */

Tinkoff.ru

Url: Tinkoff.ru

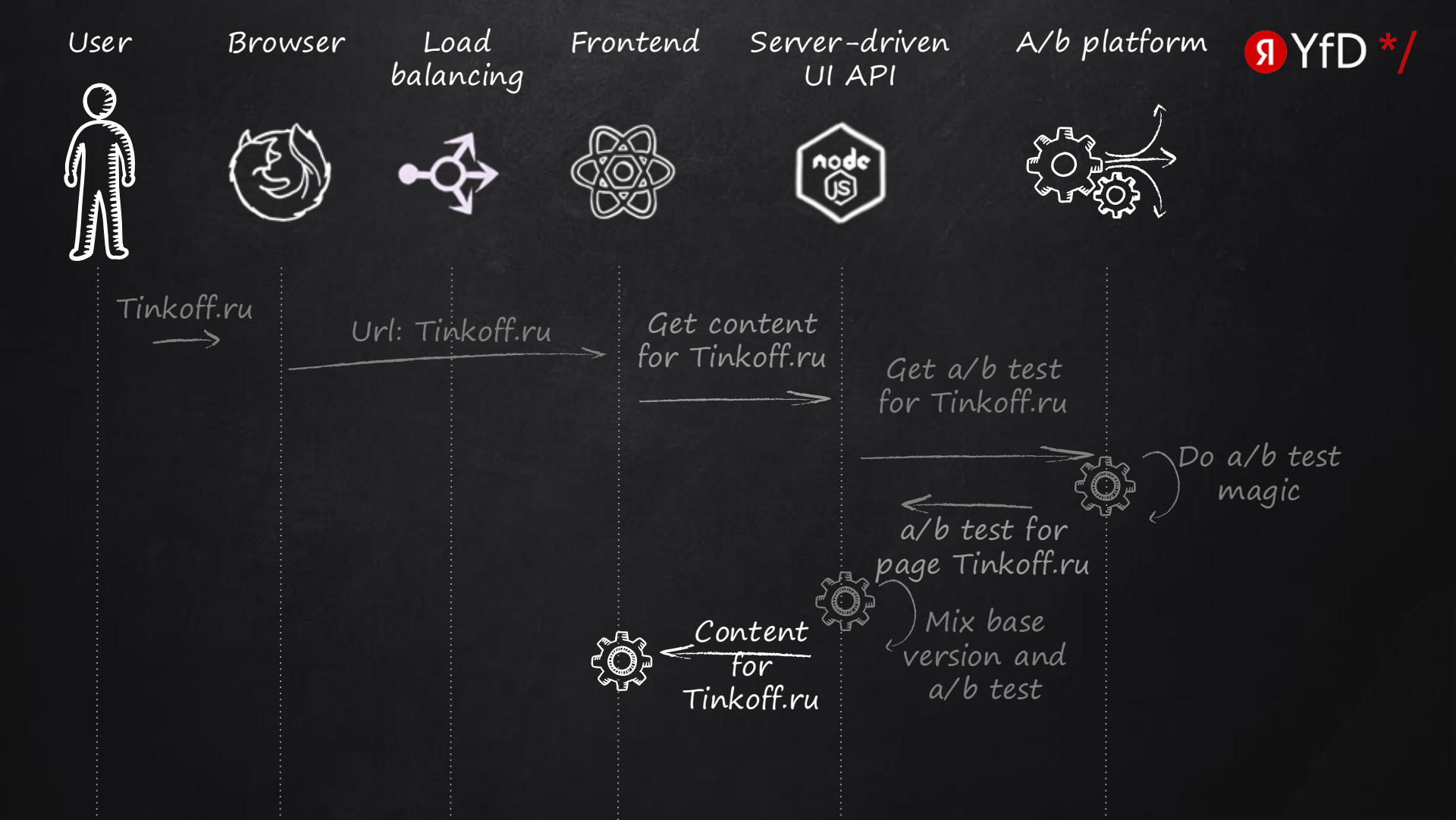
Get content for Tinkoff.ru

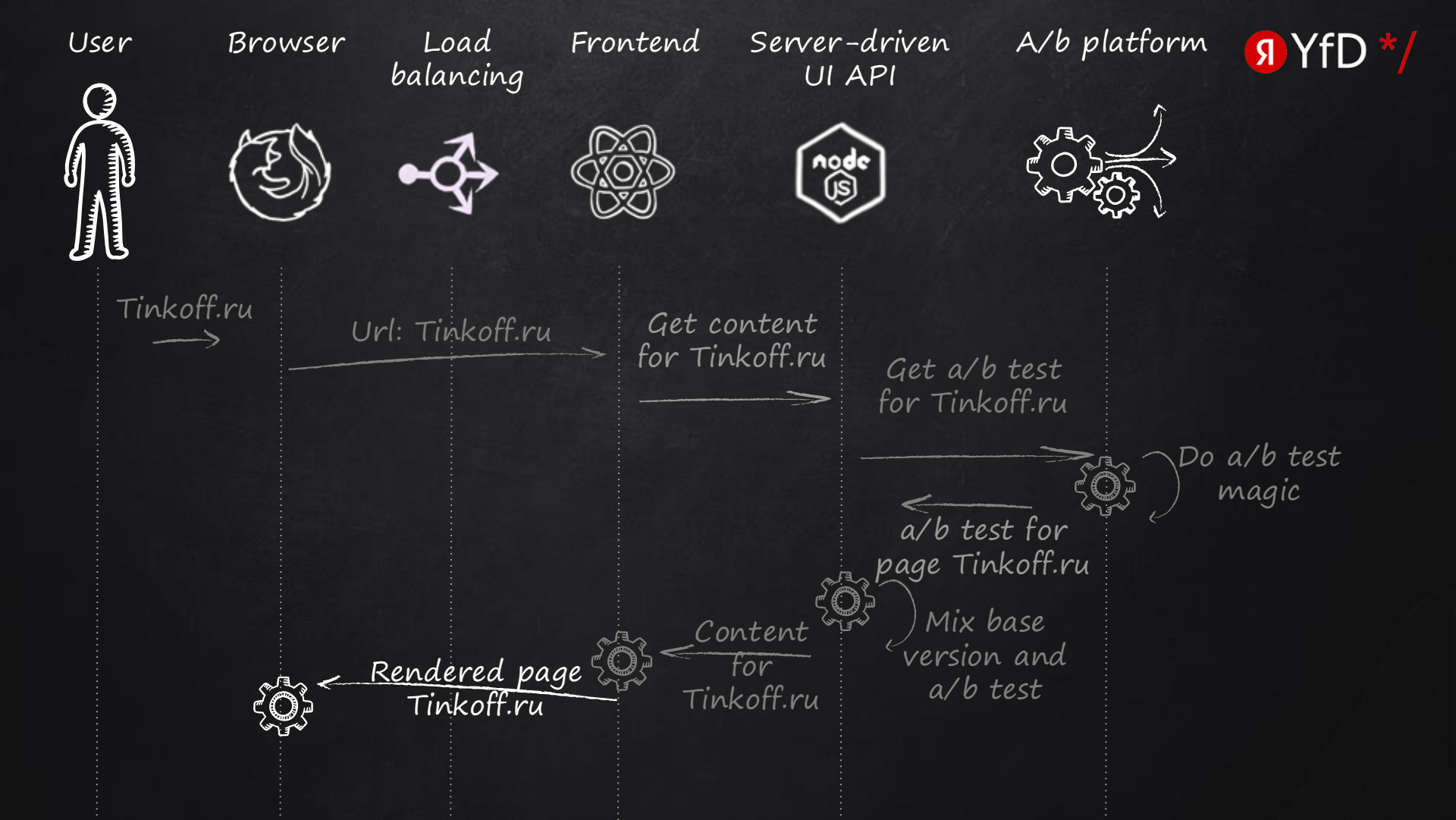
Get a/b test for Tinkoff.ru

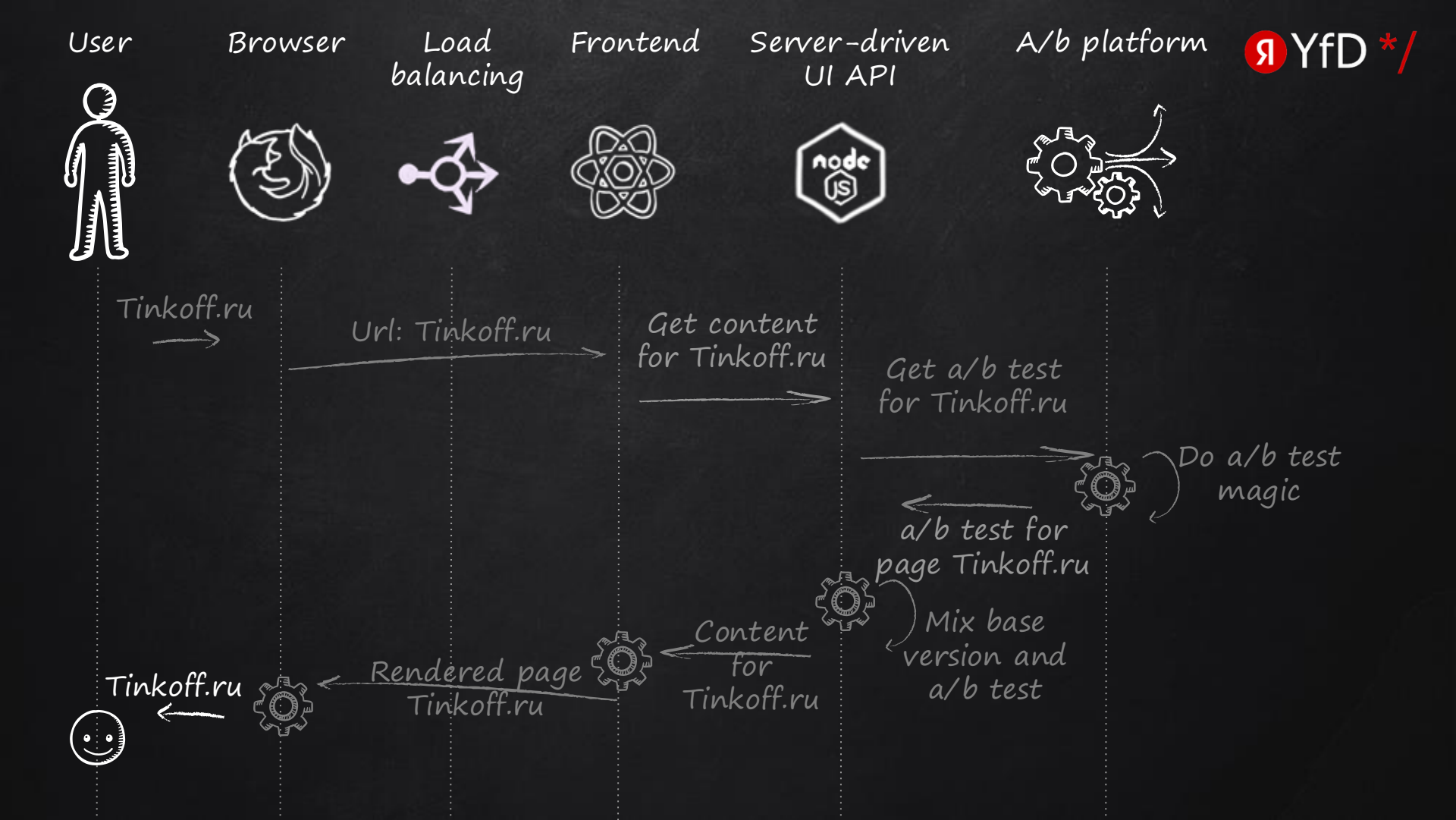
Do a/b test magic

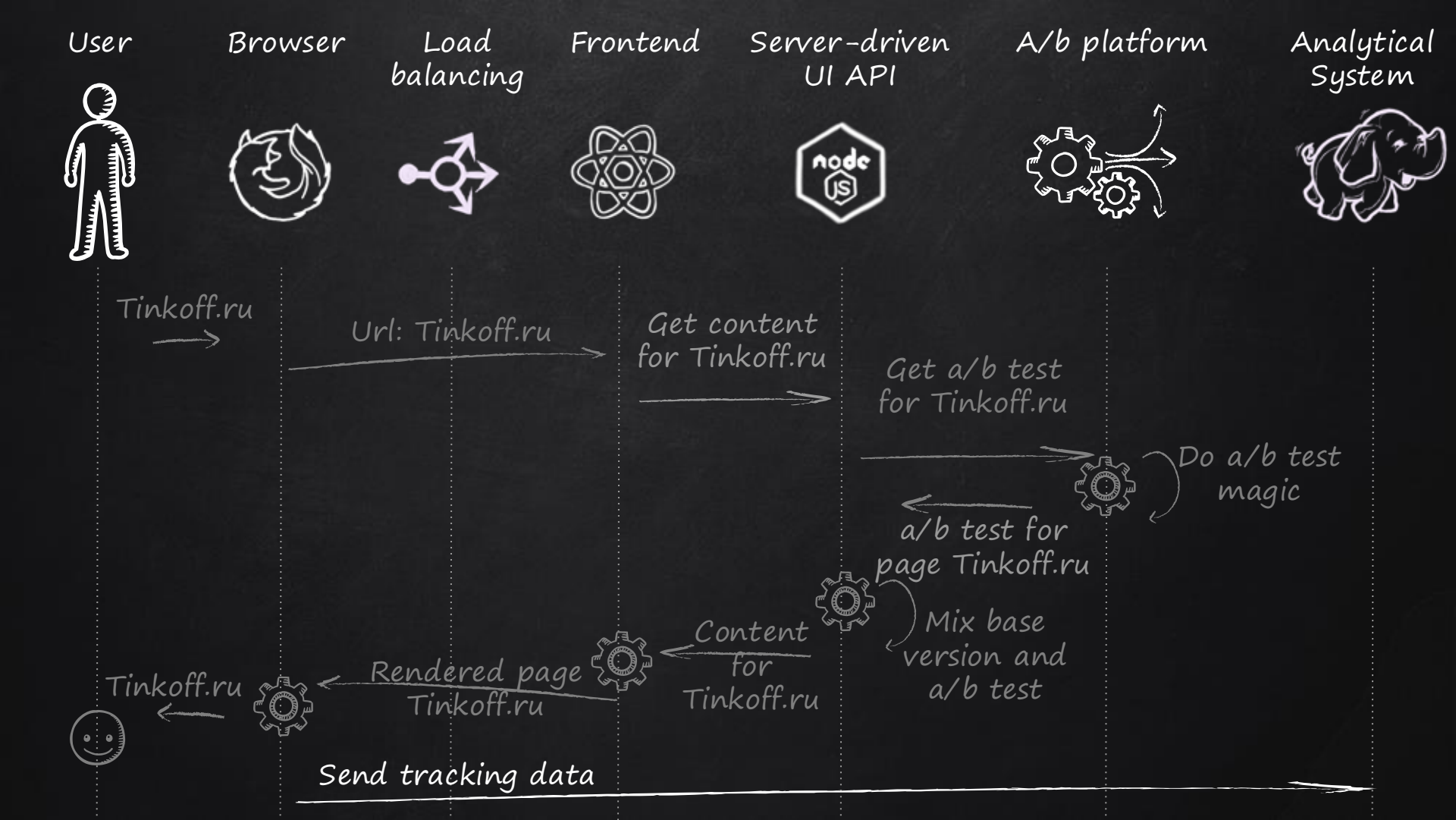
a/b test for page Tinkoff.ru

Mix base version and a/b test


User
Browser
Load balancing
Frontend
Server-driven UI API
A/b platform
YfD */
Tinkoff.ru
Url: Tinkoff.ru
Get content for Tinkoff.ru
Get a/b test for Tinkoff.ru
Do a/b test magic
a/b test for page Tinkoff.ru
Mix base version and a/b test
Content for Tinkoff.ru


User
Browser
Load balancing
Frontend
Server-driven UI API
A/b platform
YfD */
Tinkoff.ru
Url: Tinkoff.ru
Get content for Tinkoff.ru
Get a/b test for Tinkoff.ru
Do a/b test magic
a/b test for page Tinkoff.ru
Mix base version and a/b test
Content for Tinkoff.ru
Rendered page Tinkoff.ru


User
Browser
Load balancing
Frontend
Server-driven UI API
A/b platform
YfD */
Tinkoff.ru
Url: Tinkoff.ru
Get content for Tinkoff.ru
Get a/b test for Tinkoff.ru
Do a/b test magic
a/b test for page Tinkoff.ru
Mix base version and a/b test
Content for Tinkoff.ru
Rendered page Tinkoff.ru
Tinkoff.ru


User
Browser
Load balancing
Frontend
Server-driven UI API
A/b platform
Analytical System
Tinkoff.ru
Url: Tinkoff.ru
Get content for Tinkoff.ru
Get a/b test for Tinkoff.ru
Do a/b test magic
a/b test for page Tinkoff.ru
Mix base version and a/b test
Content for Tinkoff.ru
Rendered page Tinkoff.ru
Tinkoff.ru
Send tracking data

2 Решение: используем продуктовый подход

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд
  - Фронтовой команды

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд
  - Фронтовой команды
  - Content сервисов для server driven UI

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд
  - Фронтовой команды
  - Content сервисов для server driven UI
  - a/b платформы

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд
  - Фронтовой команды
  - Content сервисов для server driven UI
  - a/b платформы
- Стыкуем их через точки интеграции (API)

2 Решение: используем продуктовый подход

YfD */

- Явно определяем ответственность команд
  - Фронтовой команды
  - Content сервисов для server driven UI
  - a/b платформы
- Стыкуем их через точки интеграции (API)
- Организуем их работу по Kanban

2 Итоги

YfD */

2 Итоги

YfD */

- Команды понимают свои зоны ответственности

2 Итоги

YfD */

- Команды понимают свои зоны ответственности
- Процессы в командах выстроены

2 Итоги

YfD */

- Команды понимают свои зоны ответственности
- Процессы в командах выстроены
  - Работа визуализирована и понятна заказчикам

2 Итоги

YfD */

- Команды понимают свои зоны ответственности
- Процессы в командах выстроены
  - Работа визуализирована и понятна заказчикам
  - У команд предсказуемая пропускная способность

2 Итоги

YfD */

- Команды понимают свои зоны ответственности
- Процессы в командах выстроены
  - Работа визуализирована и понятна заказчикам
  - У команд предсказуемая пропускная способность
  - Понятно как ее повышать

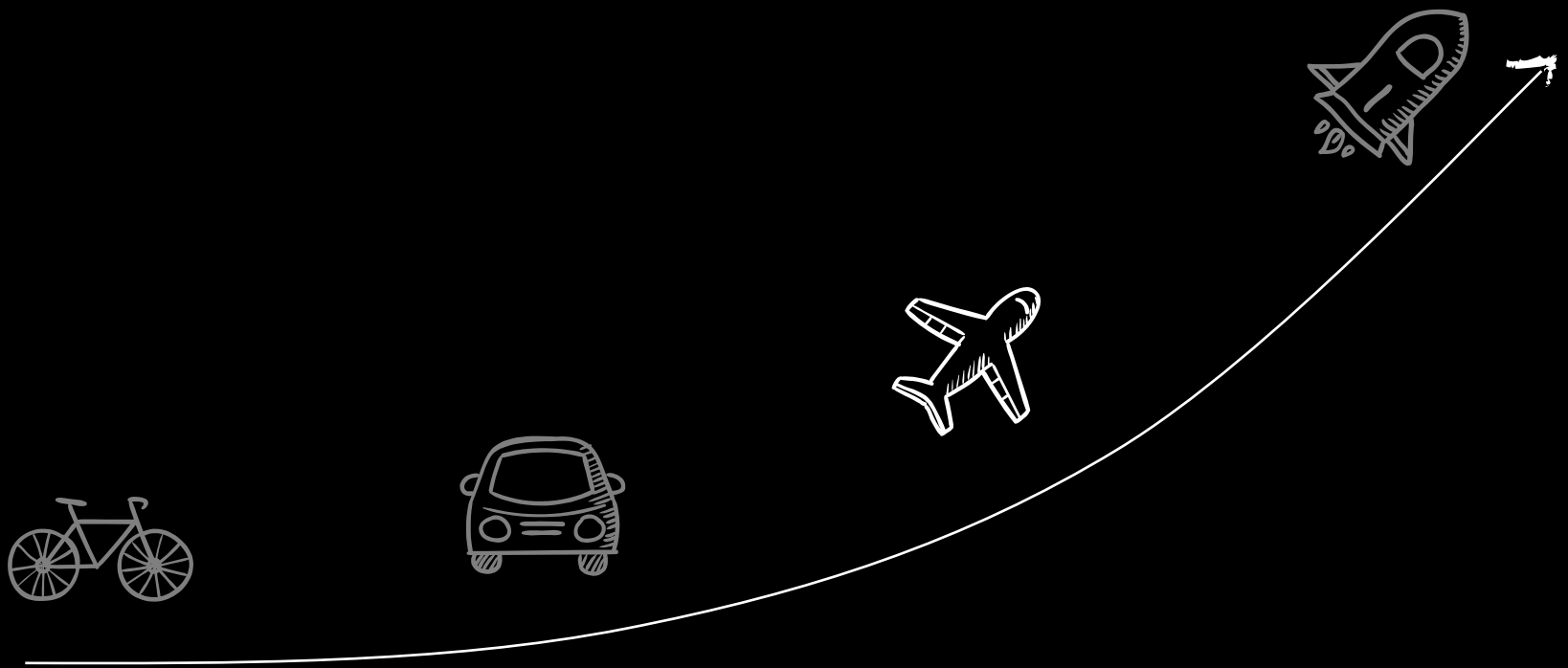
2 Итоги

YfD */

- Команды понимают свои зоны ответственности
- Процессы в командах выстроены
  - Работа визуализирована и понятна заказчикам
  - У команд предсказуемая пропускная способность
  - Понятно как ее повышать
- Получившиеся продуктовые готов к масштабированию

3 Пример: масштабирование

YfD */



3

Дано: масштабирование фронтовых продуктовых команд

3 Дано: масштабирование фронтовых продуктовых команд

YfD */

- Команды выстроены (фронт, server driven UI, a/b платформа)

3 Дано: масштабирование фронтовых продуктовых команд

YfD */

- Команды выстроены (фронт, server driven UI, a/b платформа)
- Долговременные цели

3

Дано: масштабирование фронтовых продуктовых команд

YfD */

- Команды выстроены (фронт, server driven UI, a/b платформа)
- Долговременные цели
  - Нужно научиться масштабировать фронтовую разработку за счет создания отдельных автономных продуктовых команд

3 Дано: масштабирование фронтовых продуктовых команд

YfD */

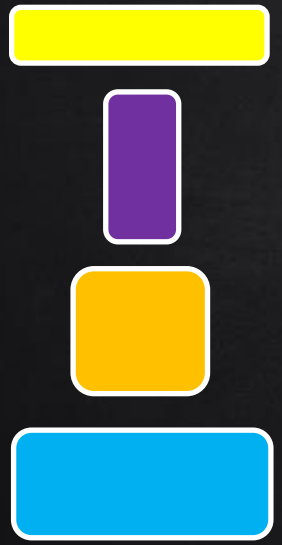
- Команды выстроены (фронт, server driven UI, a/b платформа)
- Долговременные цели
  - Нужно научиться масштабировать фронтовую разработку за счет создания отдельных автономных продуктовых команд
  - Общие компоненты нужно централизовать, экономя на масштабе и стандартизации

3 Решение: используем team topologies

3 Решение: используем team topologies

YfD */

4 типа команд



Stream-aligned team

Enabling team

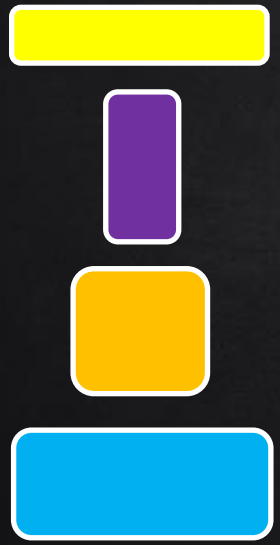
Complicated-subsystem team

Platform team

3 Решение: используем team topologies

YfD */

4 типа команд



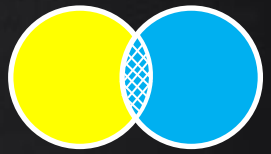
Stream-aligned team

Enabling team

Complicated-subsystem team

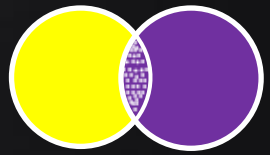
Platform team

3 типа взаимодействий



Collaborating

X-as-a-Service



Facilitating

3 Итоги

YfD */

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами
  - Сделали свой фреймворк поверх React с DI

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами
  - Сделали свой фреймворк поверх React с DI

3 Итоги

YfD */

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами
  - Сделали свой фреймворк поверх React с DI
  - Реализовали общие компоненты, подключаемые через DI

3 Итоги

YfD */

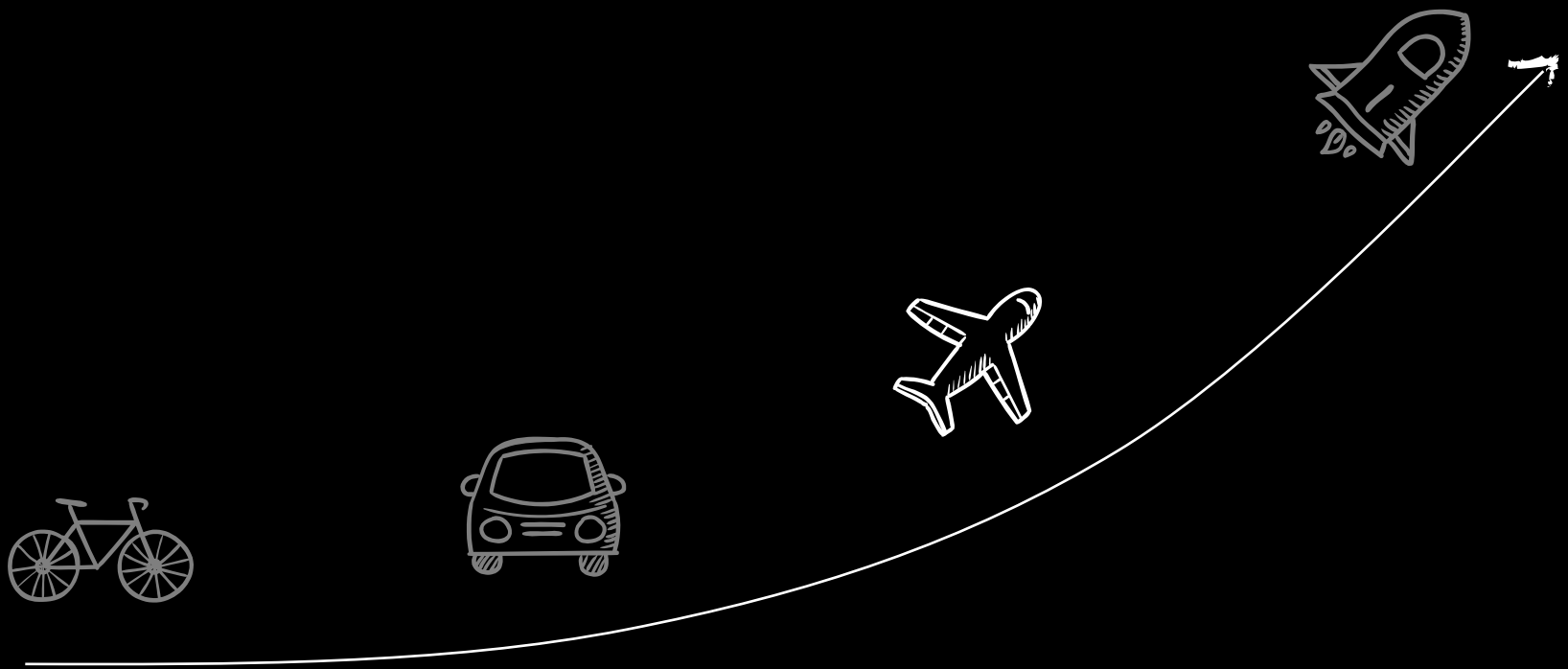
- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами
  - Сделали свой фреймворк поверх React с DI
  - Реализовали общие компоненты, подключаемые через DI
- Успешно переехали на микрофронты

3 Итоги

- Разделили команды на продуктовые и платформенные
- Продуктовые команды активно пилили фичи
- Платформенные занимались общими инструментами
  - Сделали свой фреймворк поверх React с DI
  - Реализовали общие компоненты, подключаемые через DI
- Успешно переехали на микрофронты
- Развитие всей неавторизованной части Tinkoff.ru перешло к этим командам

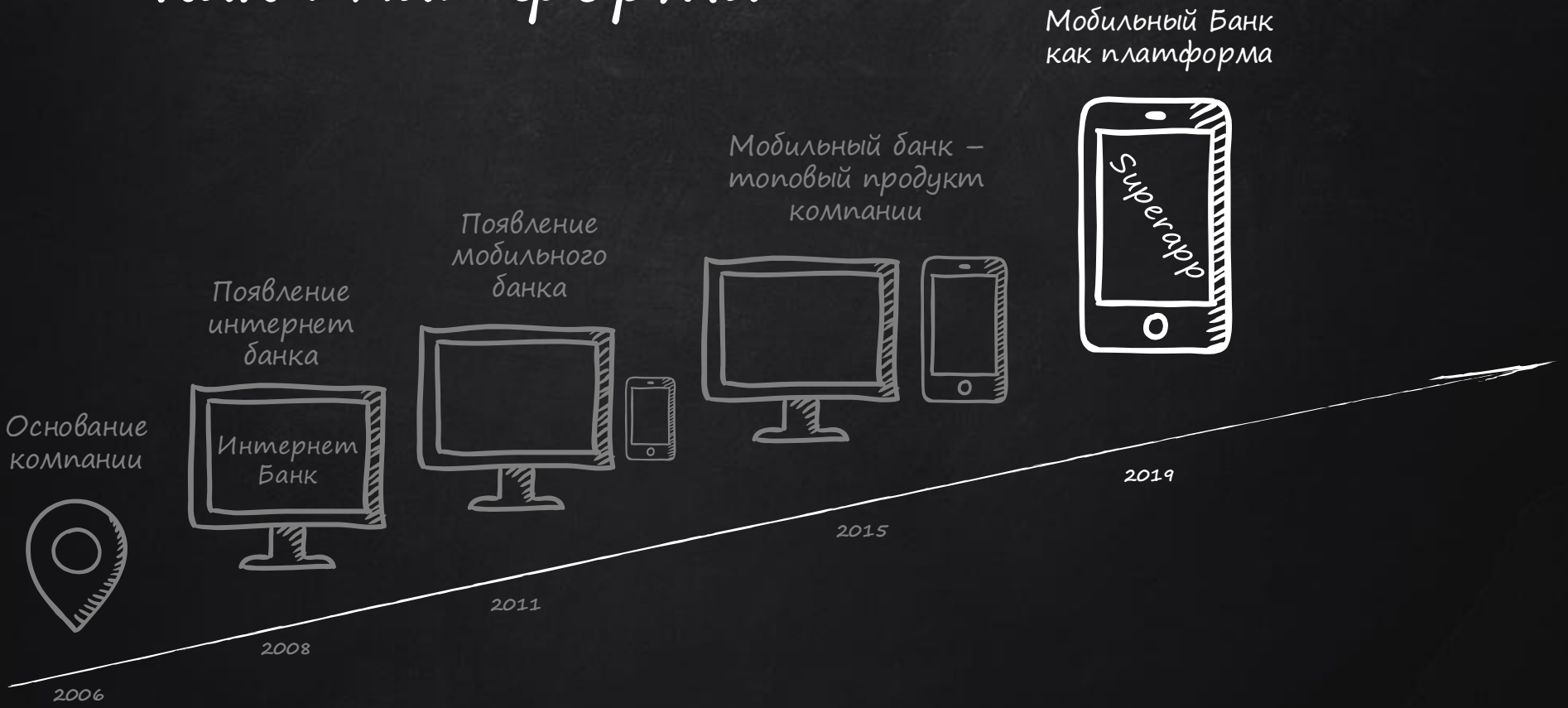
4 Пример: платформизация

YfD */



4 Дано: мобильный банк как платформа

YfD */



4 Super app не монолитен

YfD */

Superapp

4 Super app не монолитен

YfD */

Superapp

Сервисы внутри

- Банковские продукты
- Платежи и переводы
- Нефинансовые сервисы
- Страховые продукты
- ...

4 Super app не монолитен

YfD */

Superapp

Сервисы внутри

- Банковские продукты
- Платежи и переводы
- Нефинансовые сервисы
- Страховые продукты
- ...

И всех их нужно параллельно развивать, поэтому нужны были изменения

4 Решение: используем все инструменты сразу

4 Решение: используем все инструменты сразу

- Закон Конвея

4 Решение: используем все инструменты сразу

YfD */

- Закон Конвея
- Обратный маневр Конвея

4 Решение: используем все инструменты сразу

YfD */

- Закон Конвея
- Обратный маневр Конвея
- Team Topologies

4 Решение: используем все инструменты сразу

YfD */

- Закон Конвея
- Обратный маневр Конвея
- Team Topologies
- Change management

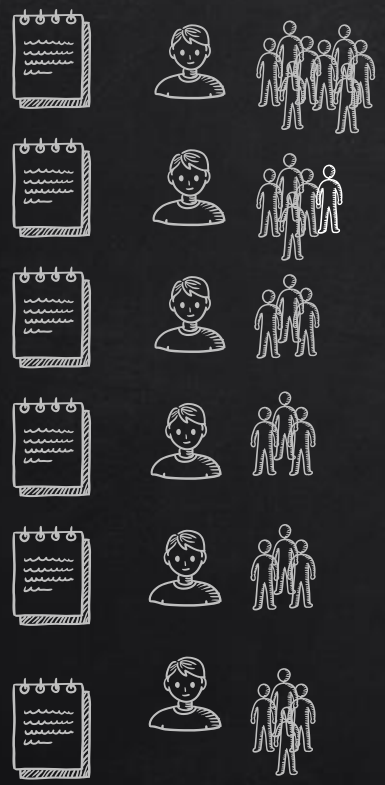
4 Новая структура команды

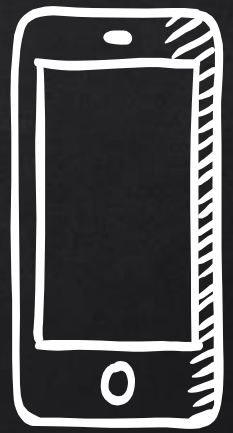
YfD */

Business

- Banking products
- Payments
- Insurance products
- SME products
- Invest products
- NonFinancial Services

Business feature teams





Platform teams

- Release
- Platform
- Design Excellence
- Performance
- Reliability

4 Архитектура (что было)

YfD */

Общая команда → Layered Architecture

Presentation Layer

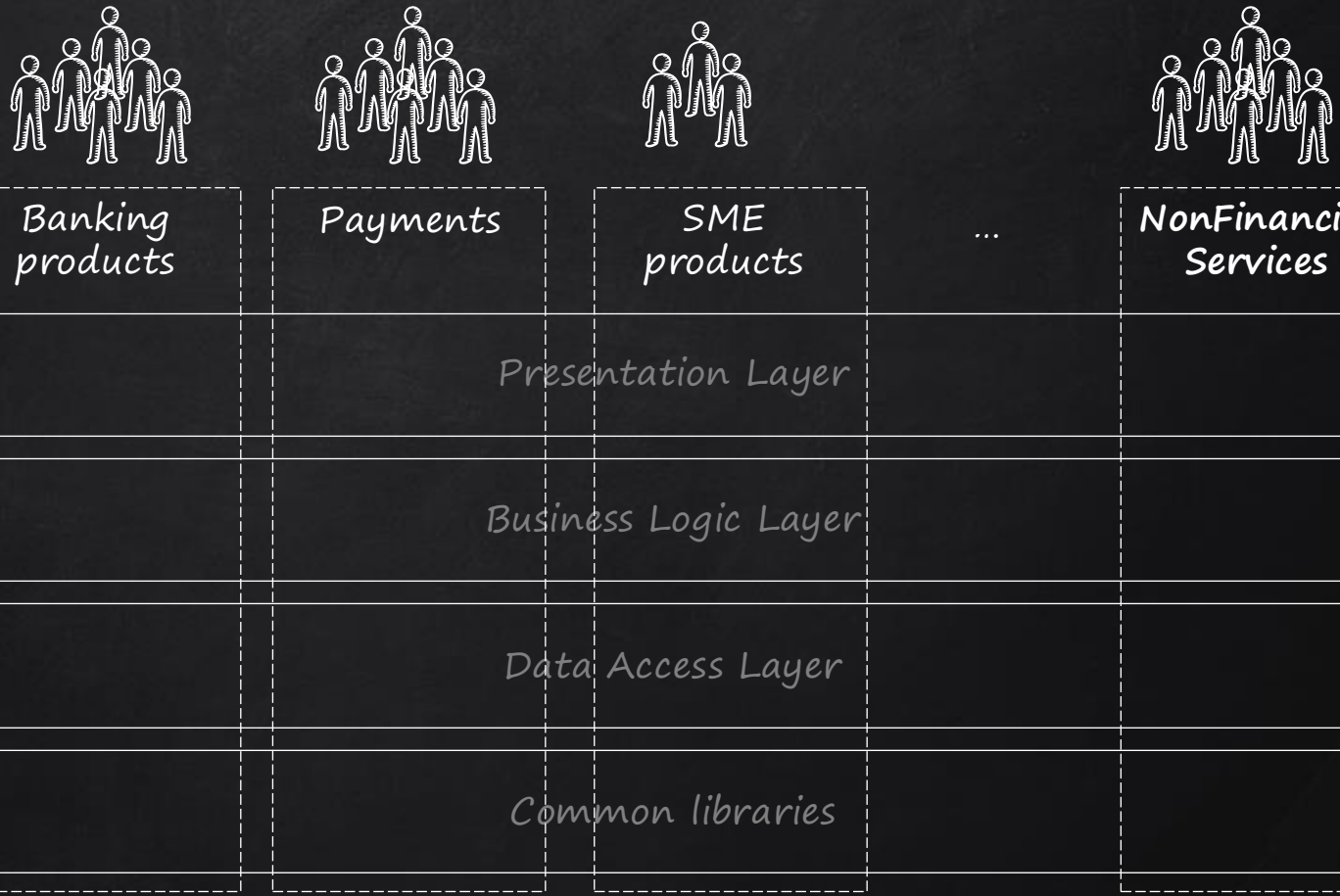
Business Logic Layer

Data Access Layer

Common libraries

4 Архитектура (виртуальное разделение)

Отдельные автономные команды с отдельными бизнес-доменами



4 Архитектура (виртуальное разделение)

Отдельные автономные команды с отдельными бизнес-доменами

И своими модулями и модульной архитектурой в общем




Banking products	Payments	SME products	...	NonFinancial Services
	Presentation Layer			
	Business Logic Layer			
	Data Access Layer			
	Common libraries			

4 Release trains

YfD */

Команды должны работать автономно
Релизы привязаны к датам, а не заранее запланированной функциональности

4 Инженерная культура

YfD */

4 Инженерная культура

YfD */

- Процессы разработки в канве Kanban-подхода формализируются и внедряются при помощи Delivery Managers

4 Инженерная культура

YfD */

- Процессы разработки в канве Kanban-подхода формализируются и внедряются при помощи Delivery Managers
- Автоматизированное тестирование обязательно для новых фич

4 Инженерная культура

YfD */

- Процессы разработки в канве Kanban-подхода формализируются и внедряются при помощи Delivery Managers
- Автоматизированное тестирование обязательно для новых фич
- Активная работа над сокращением времени регресса за счет автоматизации

4 Инженерная культура

YfD */

- Процессы разработки в канве Kanban-подхода формализируются и внедряются при помощи Delivery Managers
- Автоматизированное тестирование обязательно для новых фич
- Активная работа над сокращением времени регресса за счет автоматизации
- Использование fitness functions для контроля архитектуры с использованием Danger в шагах CI/CD

4 Инженерная культура

YfD */

- Процессы разработки в канве Kanban-подхода формализируются и внедряются при помощи Delivery Managers
- Автоматизированное тестирование обязательно для новых фич
- Активная работа над сокращением времени регресса за счет автоматизации
- Использование fitness functions для контроля архитектуры с использованием Danger в шагах CI/CD
- Улучшение инфраструктуры для сборок приложения и прогонов тестов

4 Структура команд SuperApp

YfD */

Feature team

Platform team

Руководители платформы

CPO

CTO

Delivery Manager

SuperApp

4 Итоги

YfD */

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app
  - Наладили работу над надежностью приложения (mobile SRE)

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app
  - Наладили работу над надежностью приложения (mobile SRE)
  - Оптимизировали производительность приложения и ее мониторинг

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app
  - Наладили работу над надежностью приложения (mobile SRE)
  - Оптимизировали производительность приложения и ее мониторинг
  - Улучшили инфраструктуру (CI/CD) вместе с coretech

4 Итоги

YfD */

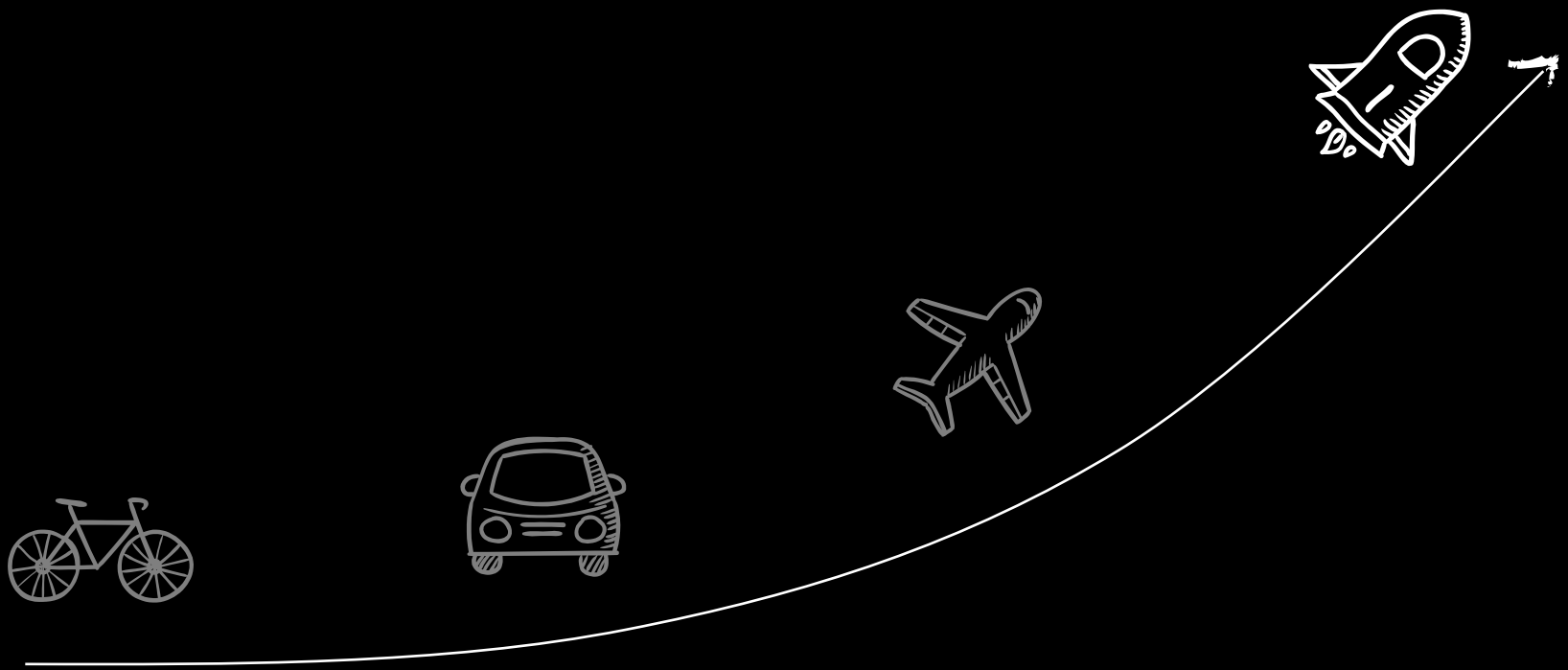
- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app
  - Наладили работу над надежностью приложения (mobile SRE)
  - Оптимизировали производительность приложения и ее мониторинг
  - Улучшили инфраструктуру (CI/CD) вместе с coretech
  - Ускорили частоту релизов в несколько раз

4 Итоги

YfD */

- Каждая бизнес-вертикаль получила автономную мобильную команду, что пилит свой бэклог
- Появились платформенные команды, которые
  - Внедрили новые архитектурные подходы к модуляризации app
  - Наладили работу над надежностью приложения (mobile SRE)
  - Оптимизировали производительность приложения и ее мониторинг
  - Улучшили инфраструктуру (CI/CD) вместе с coretech
  - Ускорили частоту релизов в несколько раз
- Процессы работы описаны и внедрены в командах, ответственность за улучшение процессов лежит на команде

5 Пример: сквозные stream-aligned команды



5

Дано: нужны улучшения в сквозном процессе поставке ценности

YfD */

5 Дано: нужны улучшения в сквозном процессе поставке ценности

YfD */

- Внутри стеков есть продуктовые и платформенные команды

5 Дано: нужны улучшения в сквозном процессе поставке ценности

- Внутри стеков есть продуктовые и платформенные команды
- Нужны сквозные stream-aligned команды, вокруг value streams

5 Дано: нужны улучшения в сквозном процессе поставке ценности

- Внутри стеков есть продуктовые и платформенные команды
- Нужны сквозные stream-aligned команды, вокруг value streams
  - Снижение TTM

5 Дано: нужны улучшения в сквозном процессе поставке ценности

- Внутри стеков есть продуктовые и платформенные команды
- Нужны сквозные stream-aligned команды, вокруг value streams
  - Снижение TTM
  - Снижение когнитивной сложности

5 Дано: нужны улучшения в сквозном процессе поставке ценности

- Внутри стеков есть продуктовые и платформенные команды
- Нужны сквозные stream-aligned команды, вокруг value streams
  - Снижение TTM
  - Снижение когнитивной сложности
  - Улучшение сквозной архитектуры решения, его эффективности и надежности

5 Решение: используем все инструменты сразу

5 Решение: используем все инструменты сразу

YfD */

- Закон Конвея

5 Решение: используем все инструменты сразу

- Закон Конвея
- Обратный маневр Конвея

5 Решение: используем все инструменты сразу

YfD */

- Закон Конвея
- Обратный маневр Конвея
- Team Topologies

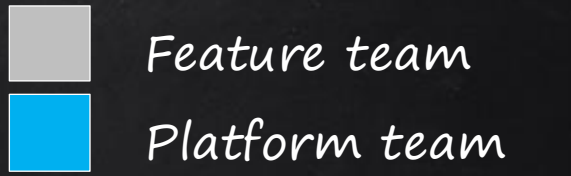
5 Решение: используем все инструменты сразу

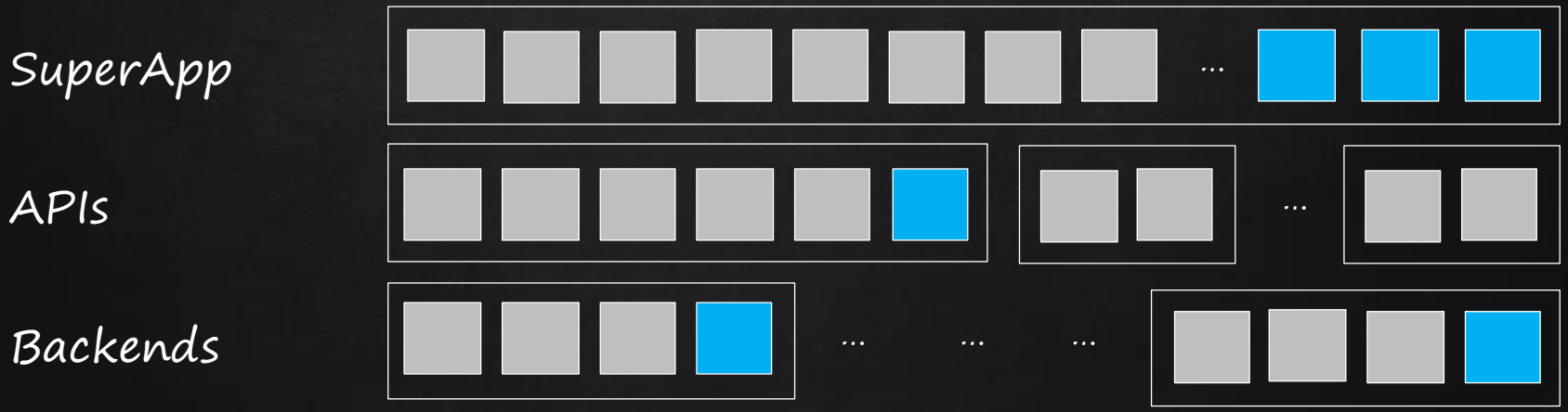
YfD */

- Закон Конвея
- Обратный маневр Конвея
- Team Topologies
- Change management

5 Слоеная структура команд

YfD */





5 Выделенное бизнес-направление

CPO

Feature team

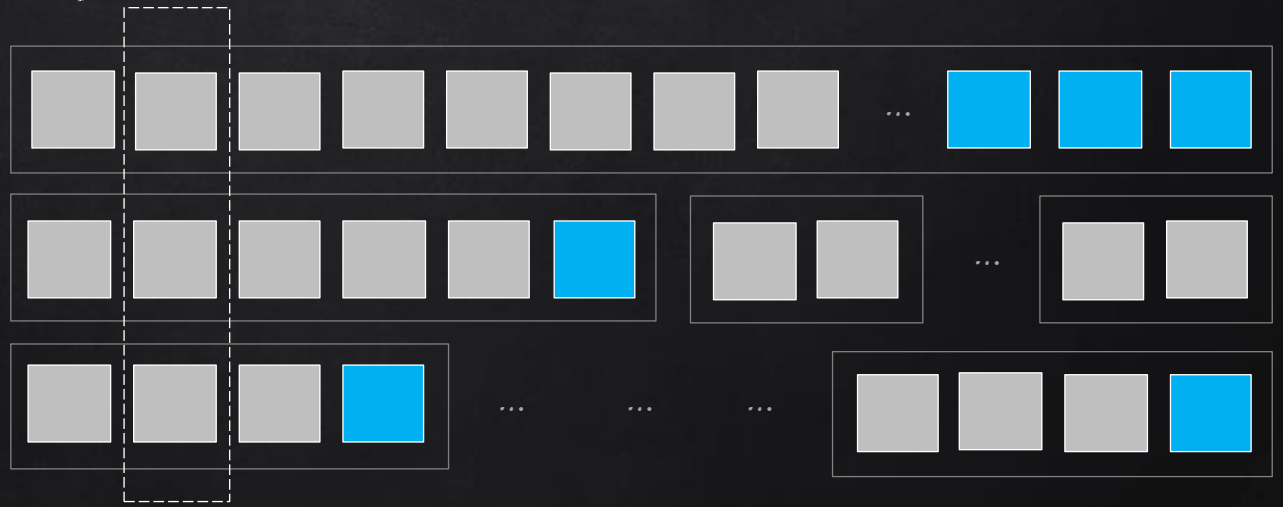
Platform team

Feature teams выделенного направления

SuperApp

APIs

Backends



5 Stream aligned team

Stream aligned team

Feature team

Platform team

Stream aligned team

SuperApp

APIs

Backends

5 Stream aligned team in details


CPO

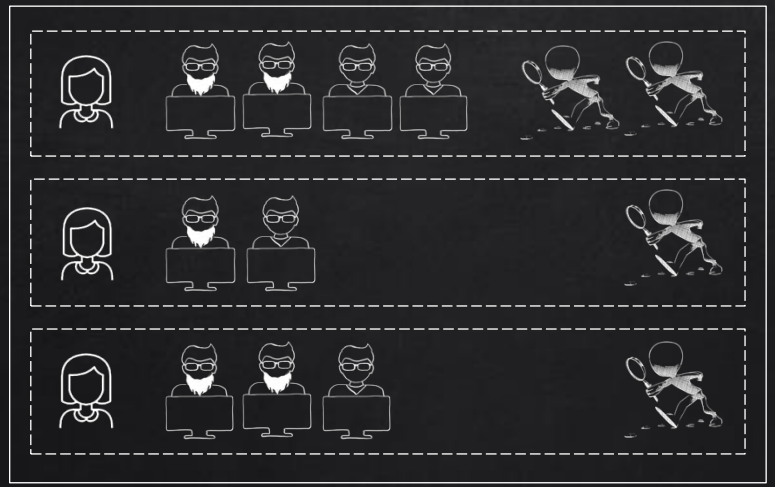

CTO


Delivery Manager

SuperApp

APIs

Backends



Analyst Developers QA Engineer

5 Сквозное проектирование

YfD */


CPO

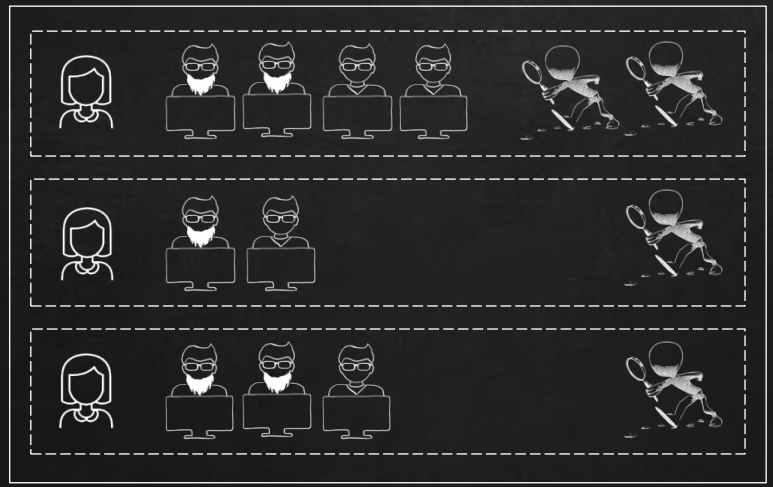

CTO


Delivery Manager

SuperApp

APIs

Backends



Analyst Developers QA Engineer

- Проектируем по всему стеку (ответственность CTO)
- Меняем подход к требованиям на user stories вместо поэкранных описаний
- **И аналитики пишут требования по всему стеку**

5 Сквозные процессы

YfD */


CPO

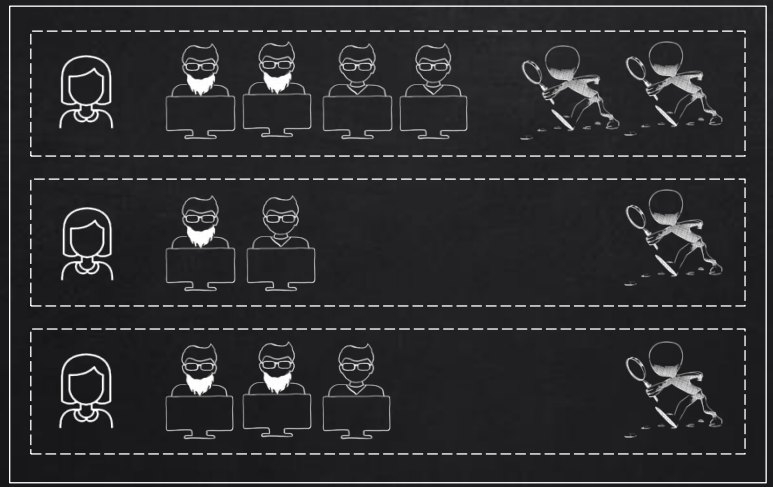

CTO


Delivery Manager

SuperApp

APIs

Backends



Analyst Developers QA Engineer

- Строим сквозные процессы с помощью Delivery Manager
- Метрики по сквозному процессу TTM, lead time, cycle time
- Работа как с downstream, так и с upstream

5 Сквозная архитектура

YfD */


CPO

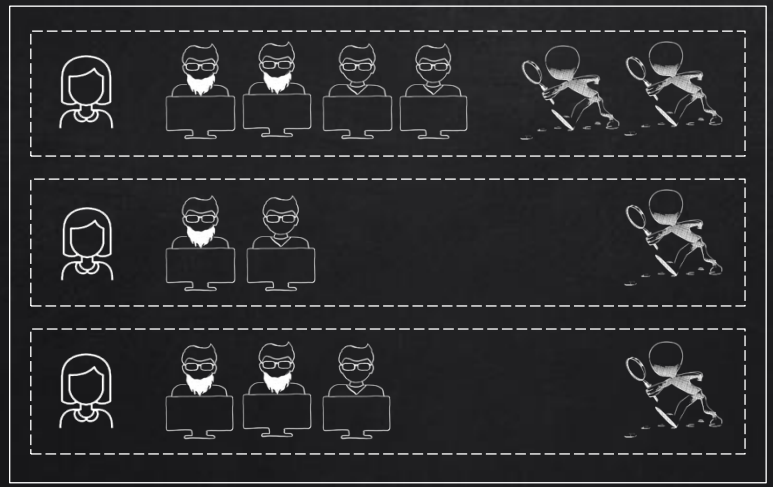

CTO


Delivery Manager

SuperApp

APIs

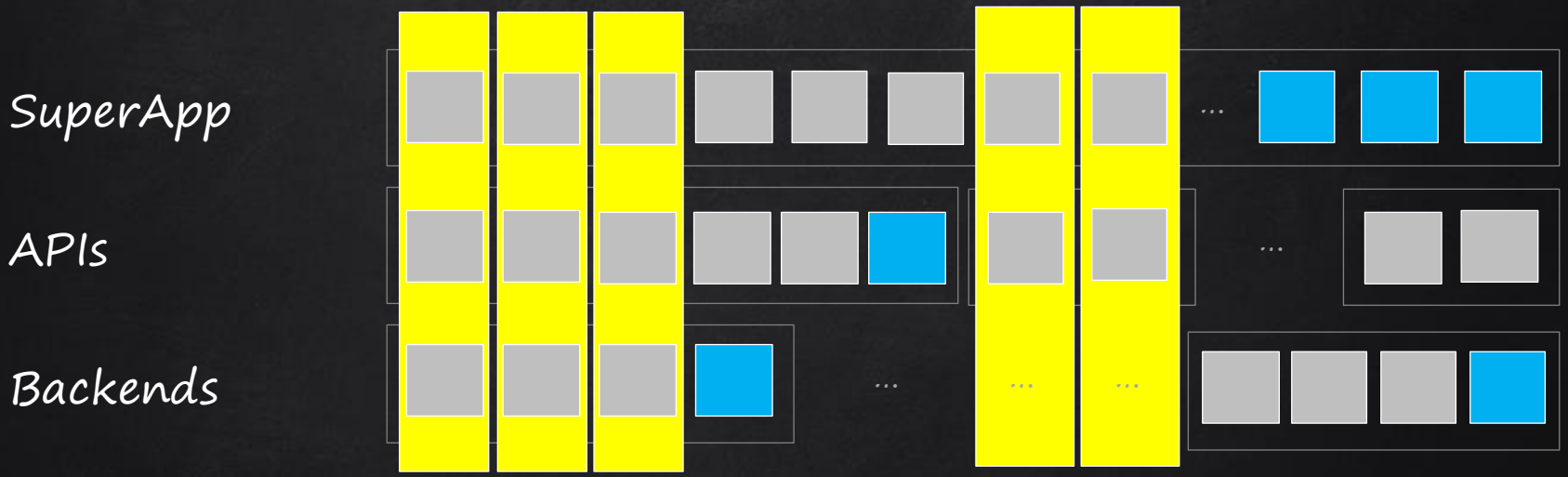
Backends



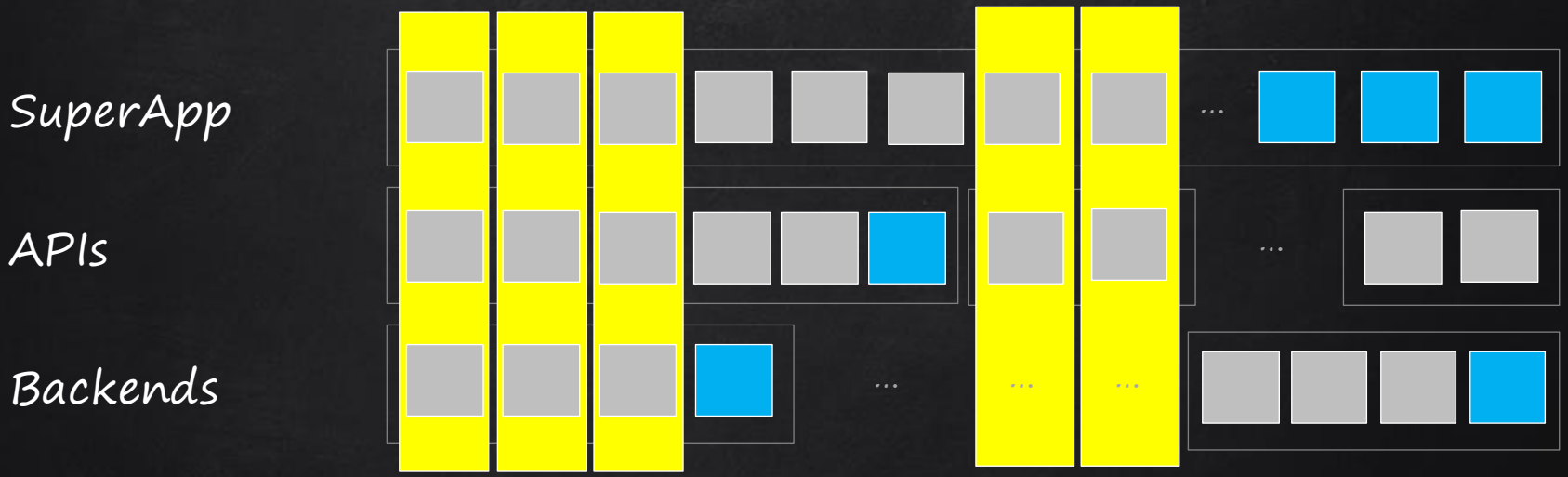
Analyst Developers QA Engineer

- Движемся от API в стиле «собери себе данные сам» к BFF для мобильных приложений
- Думаем над переходом к сборке read-моделей на основе потока событий

5 У нас много stream-aligned команд



5 У нас много stream-aligned команд



И как в этом случае осуществлять платформенный technical governance?

5 Обязательные правила для команд SuperApp описывают

YfD */

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты
- Как выглядит процесс разработки, которому они должны следовать

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты
- Как выглядит процесс разработки, которому они должны следовать
  - Фаза development, когда создаются фичи

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты
- Как выглядит процесс разработки, которому они должны следовать
  - Фаза development, когда создаются фичи
  - Фаза deployment, когда идет подготовка к развертыванию и развертывание

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты
- Как выглядит процесс разработки, которому они должны следовать
  - Фаза development, когда создаются фичи
  - Фаза deployment, когда идет подготовка к развертыванию и развертывание
- Как выглядит процесс эксплуатации, например, как обрабатываем SD или как разбираем баги с production

5 Обязательные правила для команд SuperApp описывают

YfD */

- Что такое команда и какие у нее должны быть атрибуты
- Как выглядит процесс разработки, которому они должны следовать
  - Фаза development, когда создаются фичи
  - Фаза deployment, когда идет подготовка к развертыванию и развертывание
- Как выглядит процесс эксплуатации, например, как обрабатываем SD или как разбираем баги с production
- Правила должны проверяться автоматически и показывать уровень здоровья команд

5 Метрики, на которые смотрят руководители платформы

YfD */

5 Метрики, на которые смотрят руководители платформы

YfD */

Отслеживаем эти метрики, чтобы понимать, как работа платформенных команд и изменения процессов влияют на

5 Метрики, на которые смотрят руководители платформы

YfD */

Отслеживаем эти метрики, чтобы понимать, как работа платформенных команд и изменения процессов влияют на

- Производительность разработчиков (MR, LoC)

5 Метрики, на которые смотрят руководители платформы

YfD */

Отслеживаем эти метрики, чтобы понимать, как работа платформенных команд и изменения процессов влияют на

- Производительность разработчиков (MR, LoC)
- Качество продукта (prod bugs, hotfixes and cherry picks)

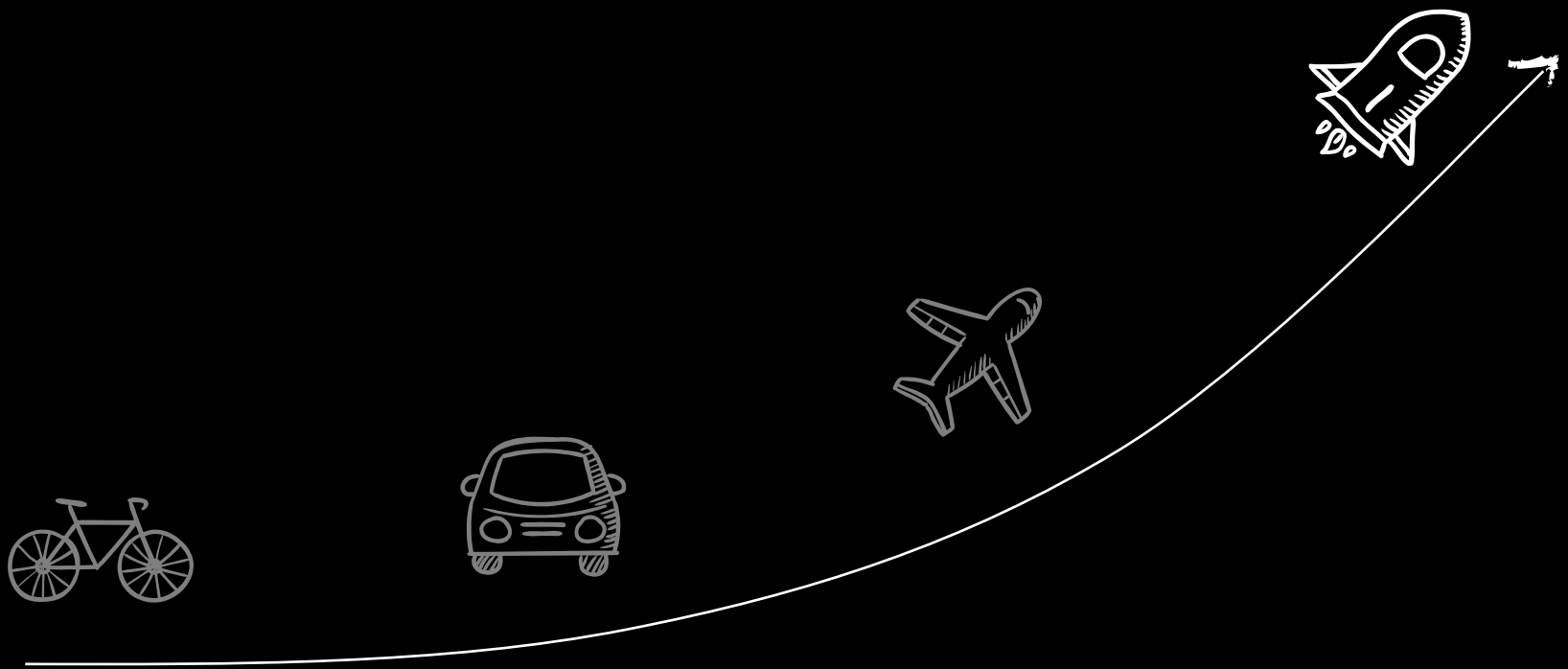
5 Метрики, на которые смотрят руководители платформы

YfD */

Отслеживаем эти метрики, чтобы понимать, как работа платформенных команд и изменения процессов влияют на

- Производительность разработчиков (MR, LoC)
- Качество продукта (prod bugs, hotfixes and cherry picks)
- Производительность команд (статистика по фичам, throughput/headcount)

6 Пример: split



Подводим итоги

Мы рассмотрели

- Ключевые подходы

Мы рассмотрели

- Ключевые подходы
- Алгоритм формирования структуры

Мы рассмотрели

- Ключевые подходы
- Алгоритм формирования структуры
- Применение алгоритма на примерах из Тинькофф

Спасибо

Александр Поломодов
Технический директор
Тинькофф

t.me/book_cube/XXXX