

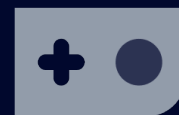
**Вам не нужно
хранилище
в мобильном
приложении**



Сегодня



Базовые принципы Offline-first мобильных приложения



Классификация приложений



Способы реализации «правильного» offline-first



Выводы

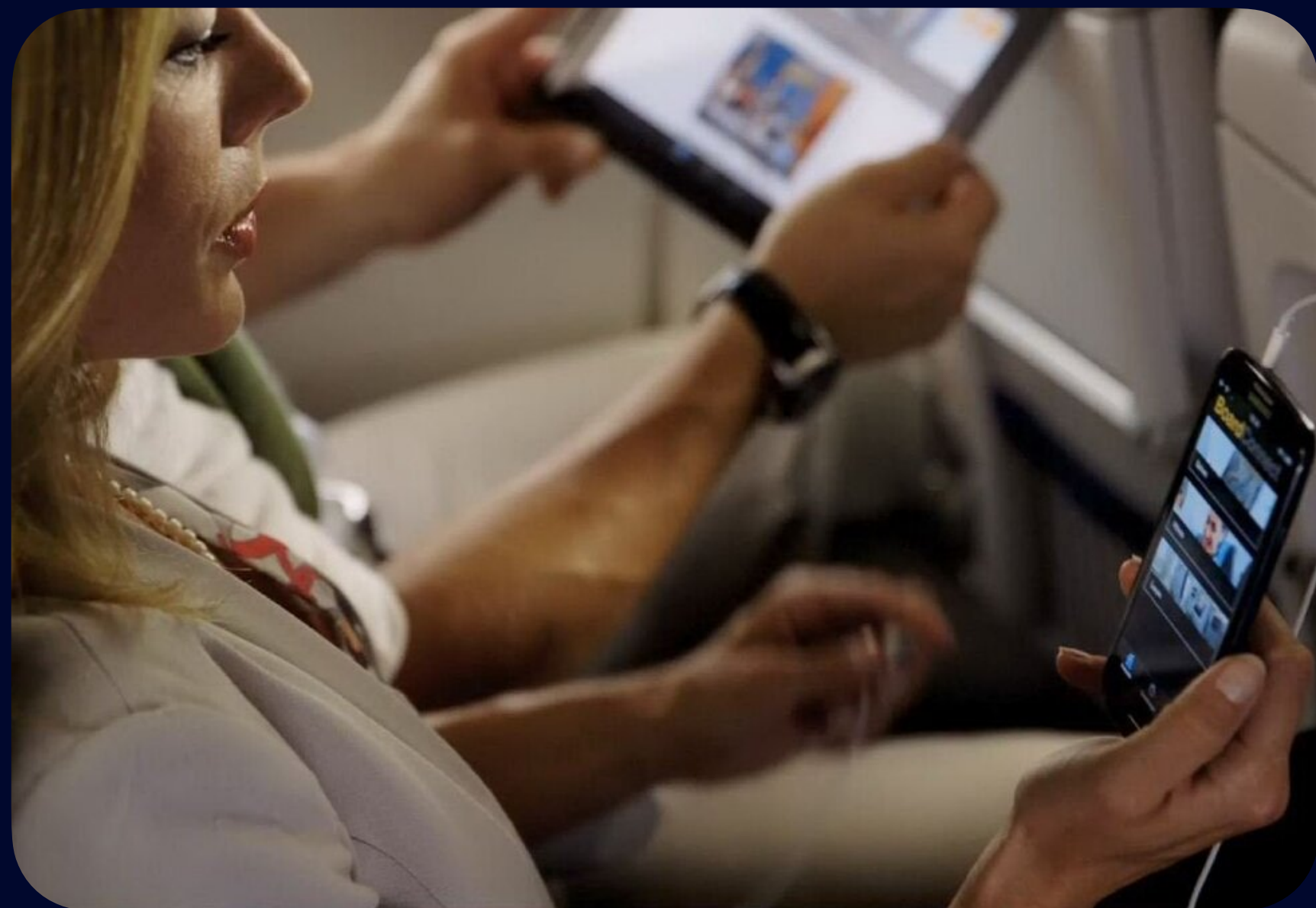
С чего все начиналось

→ Слабый процессор

→ Медленный 3g

→ Экономия трафика





Когда нет интернета?

- Самолет
- Кассы и пункты выдачи
- Лифт



Когда нет интернета?

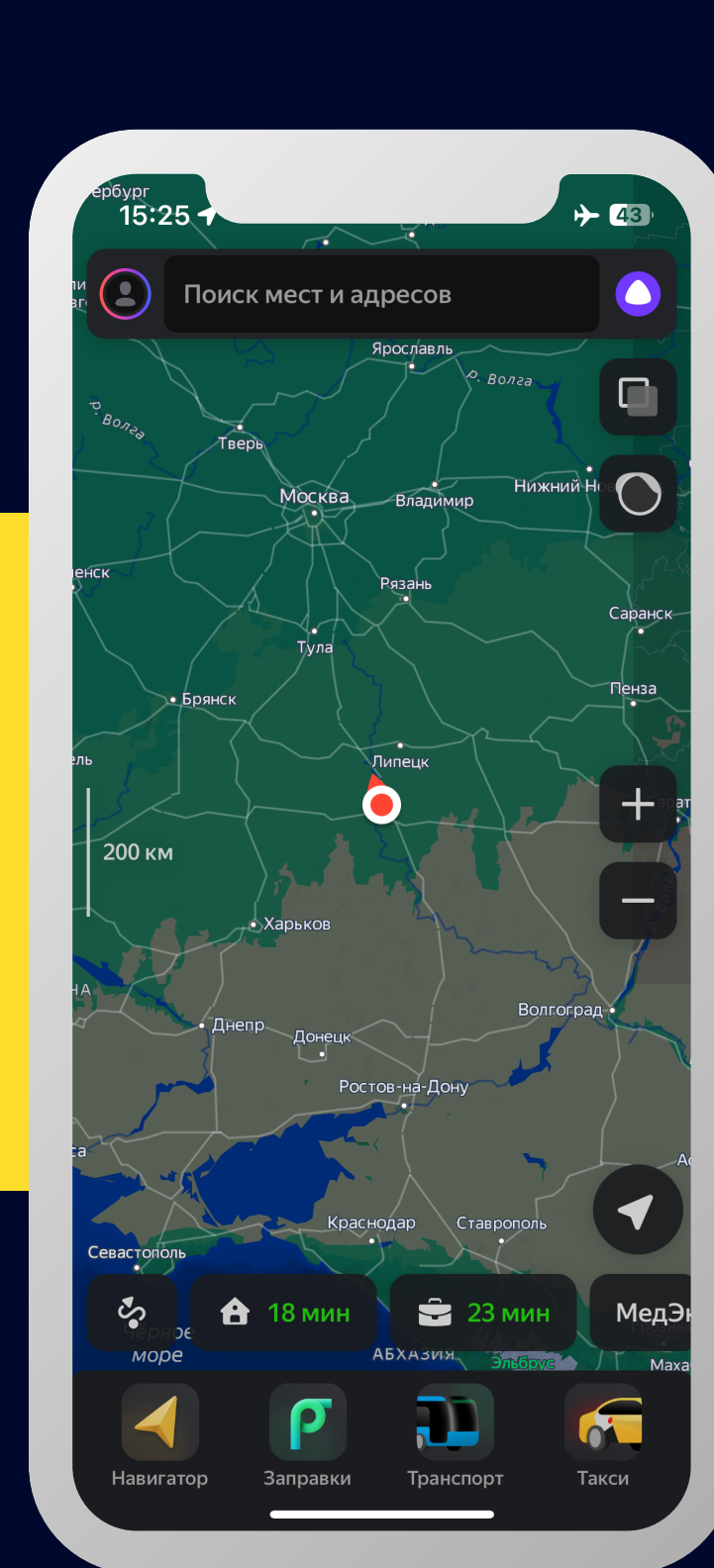
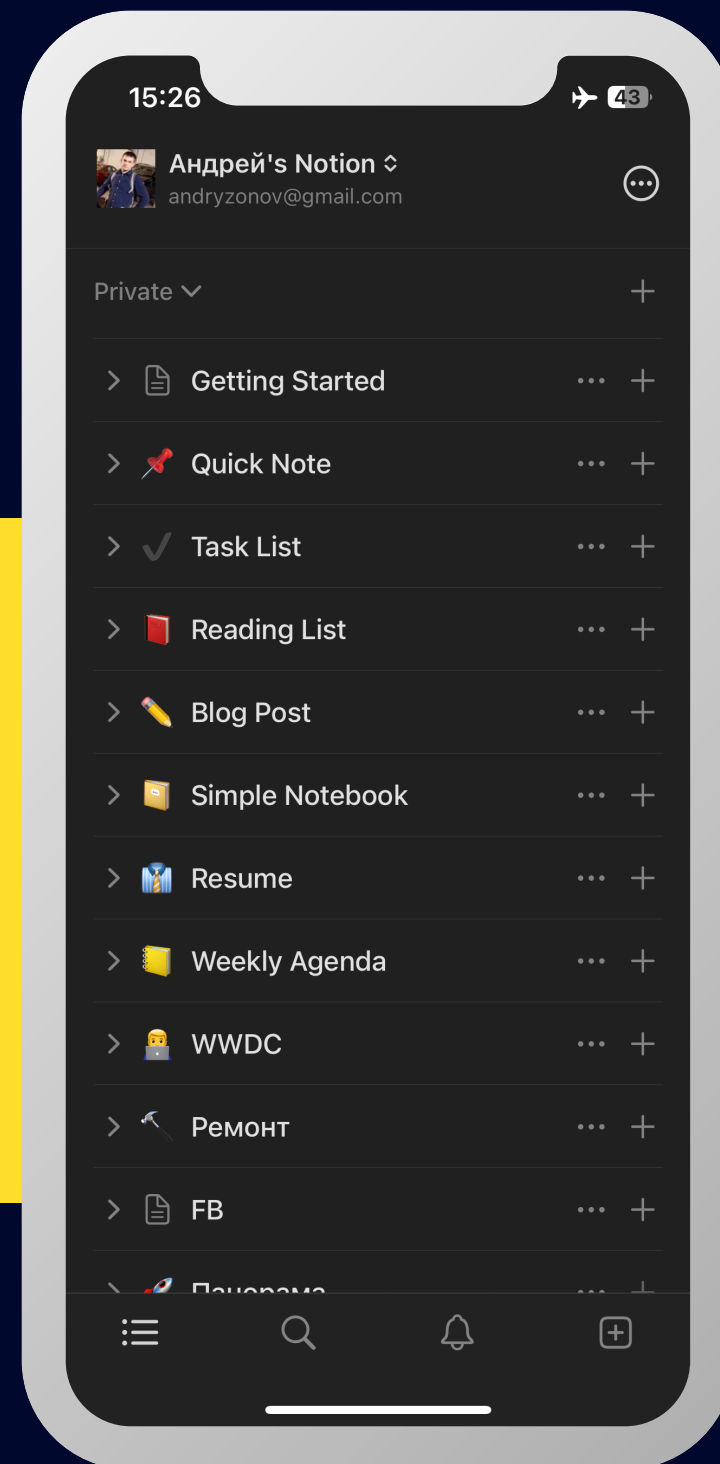
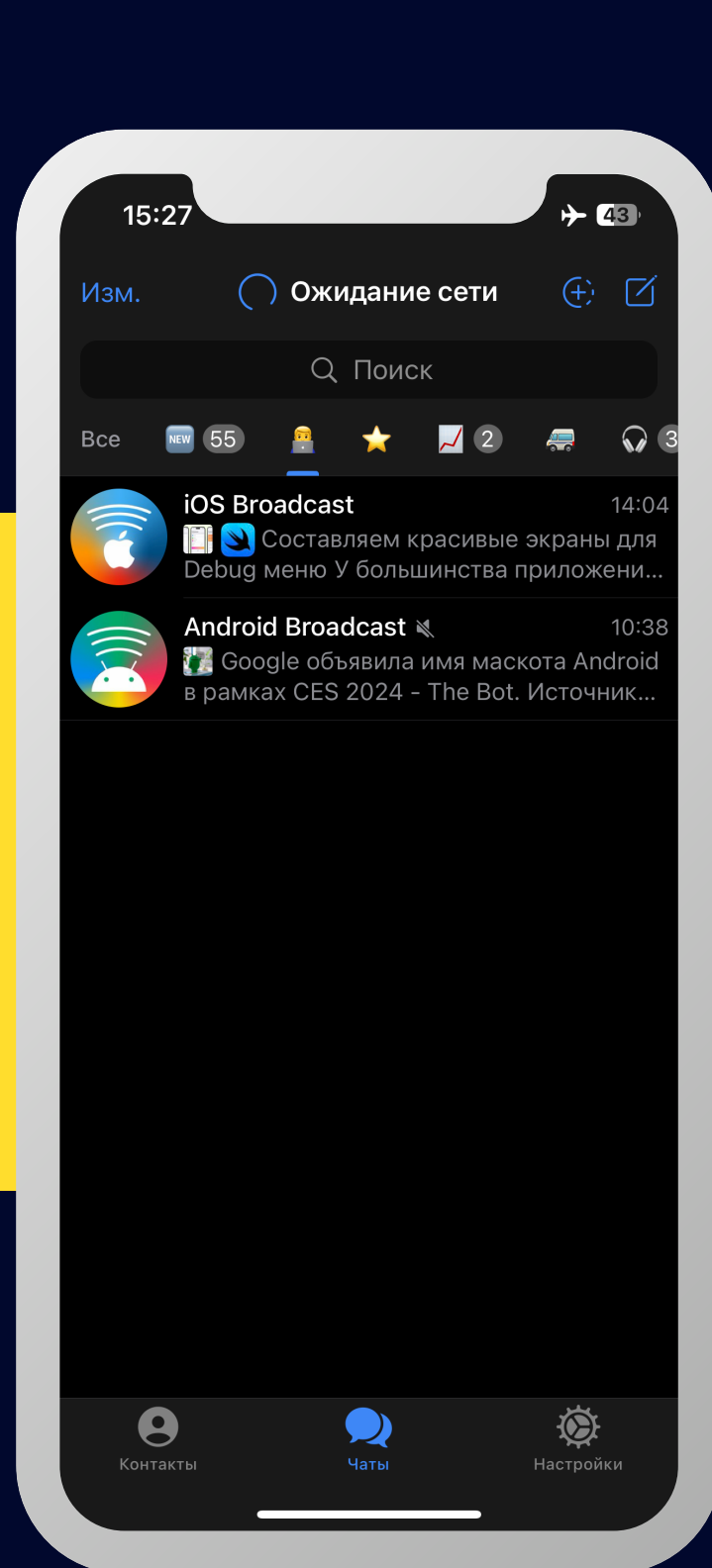
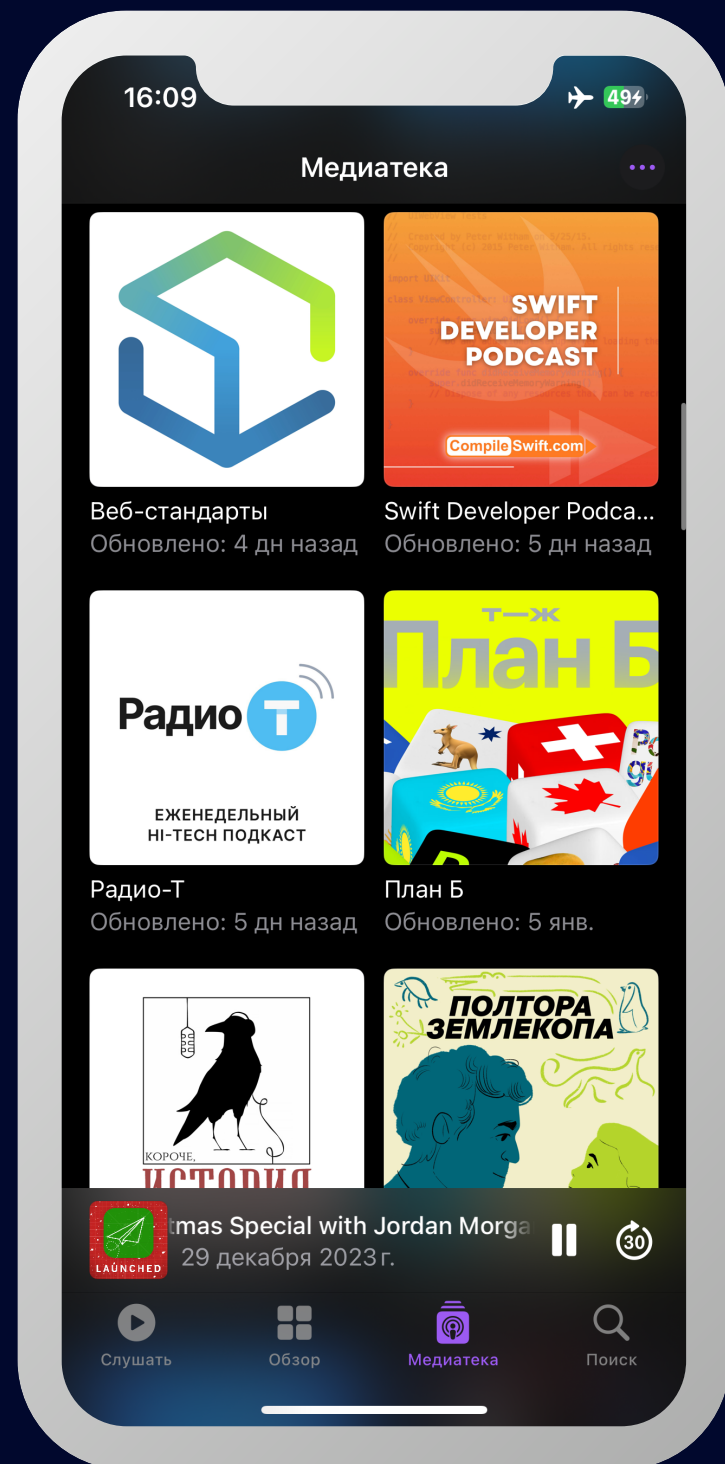
- Самолет
- Кассы и пункты выдачи
- Лифт



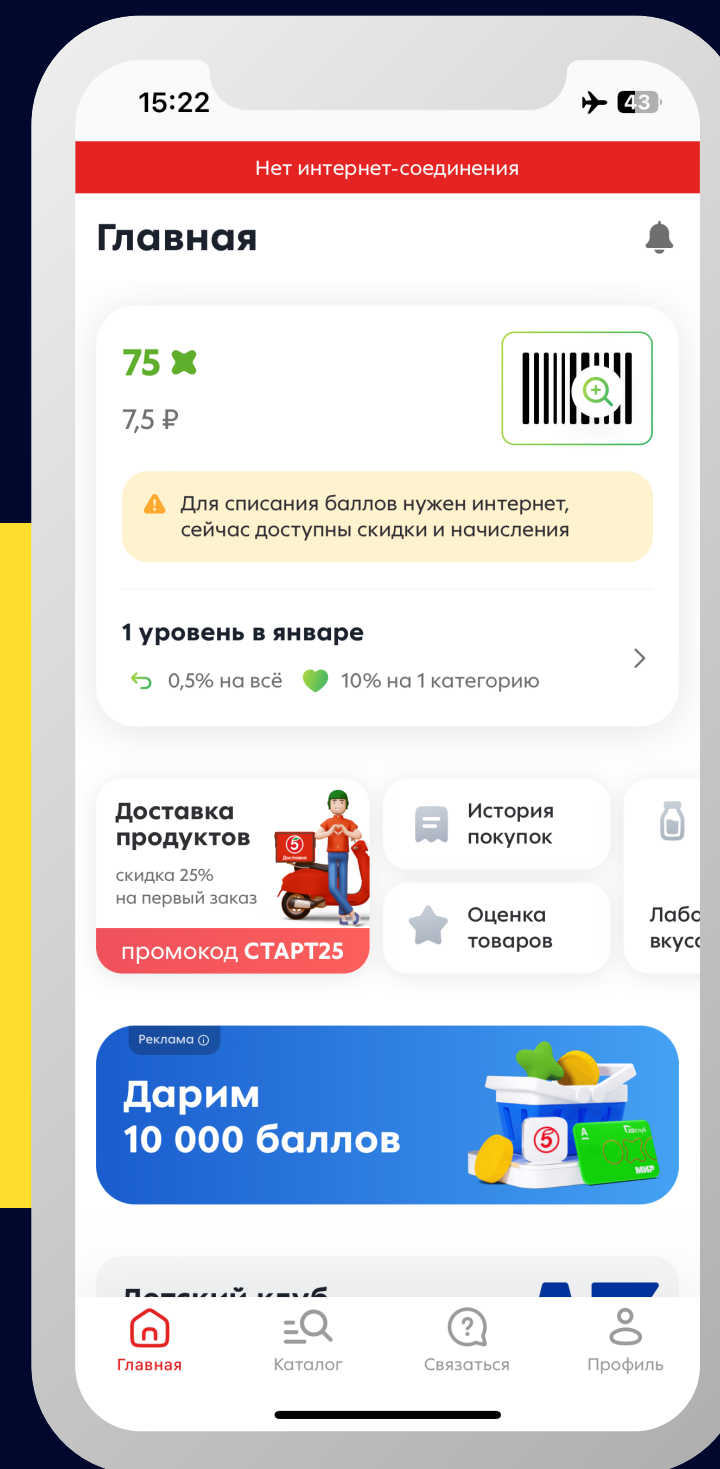
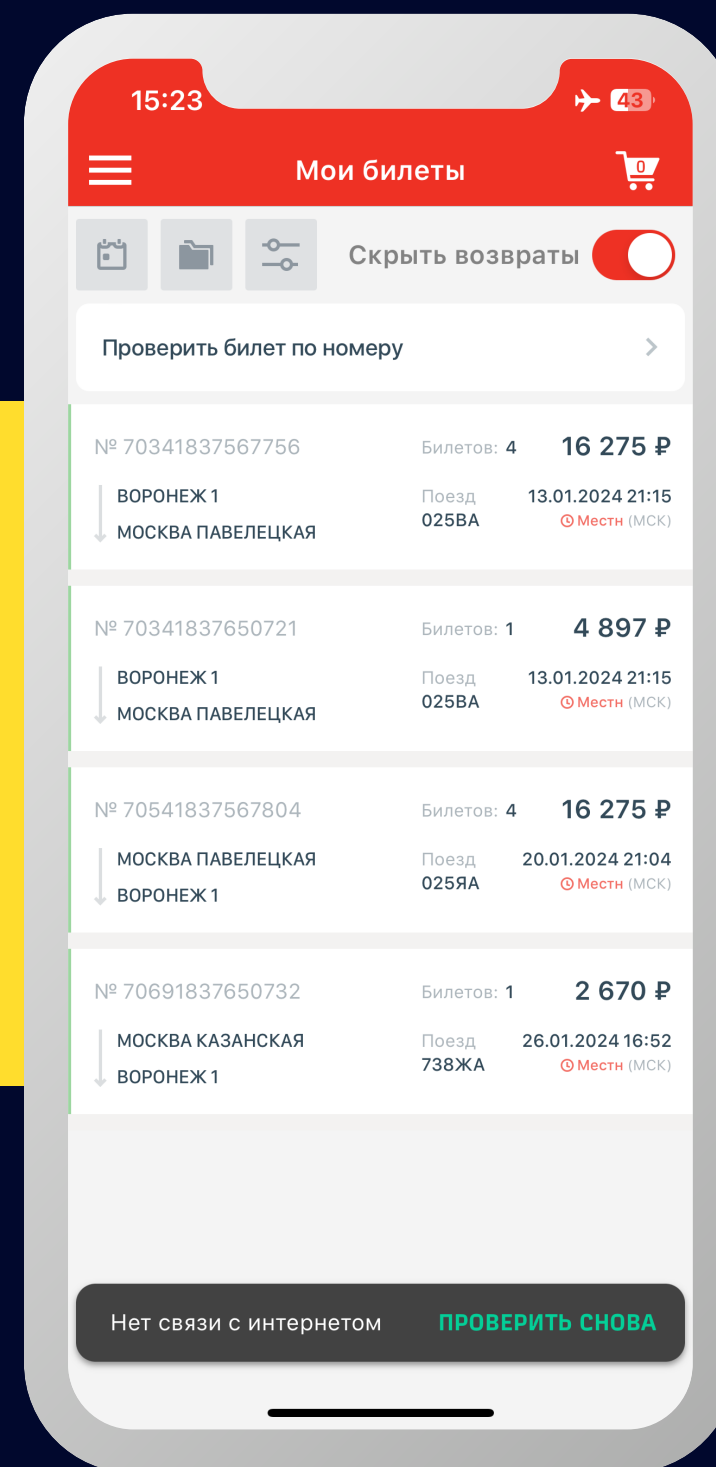
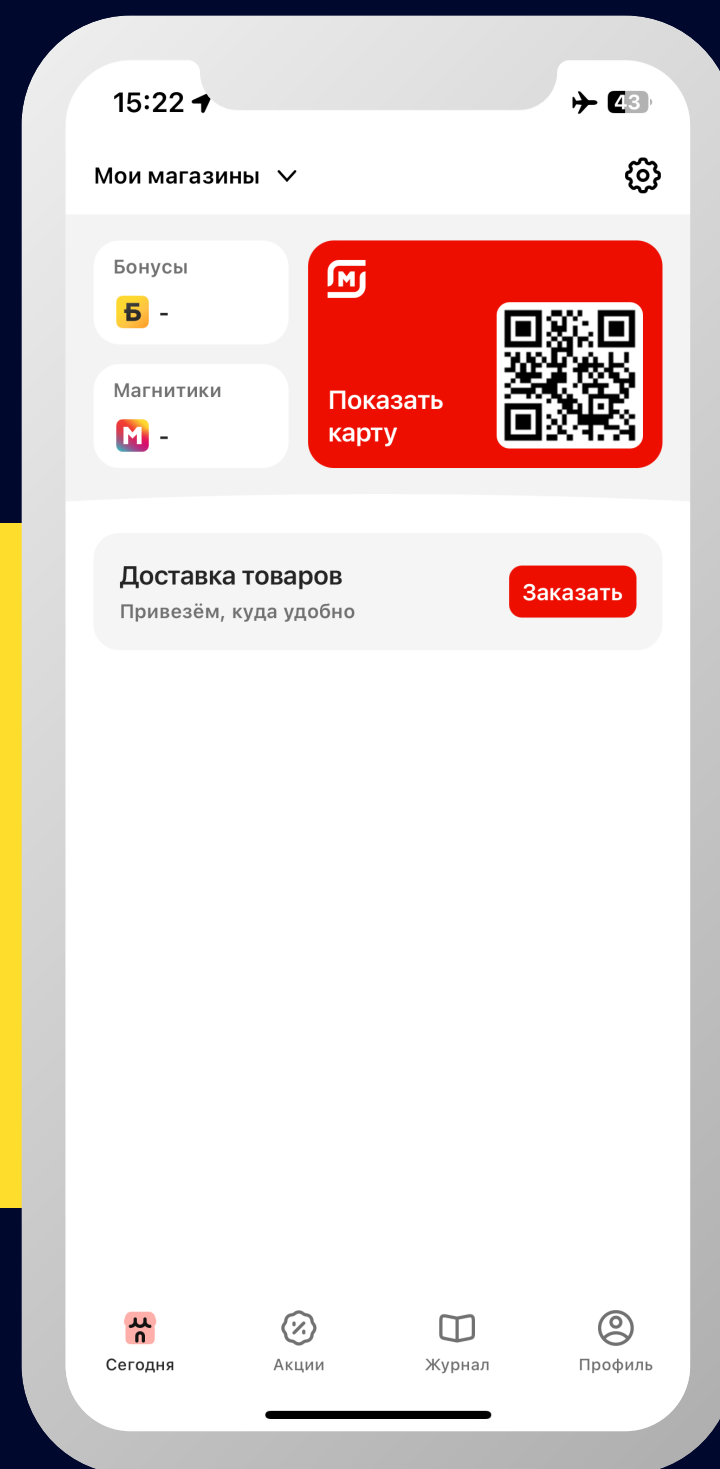
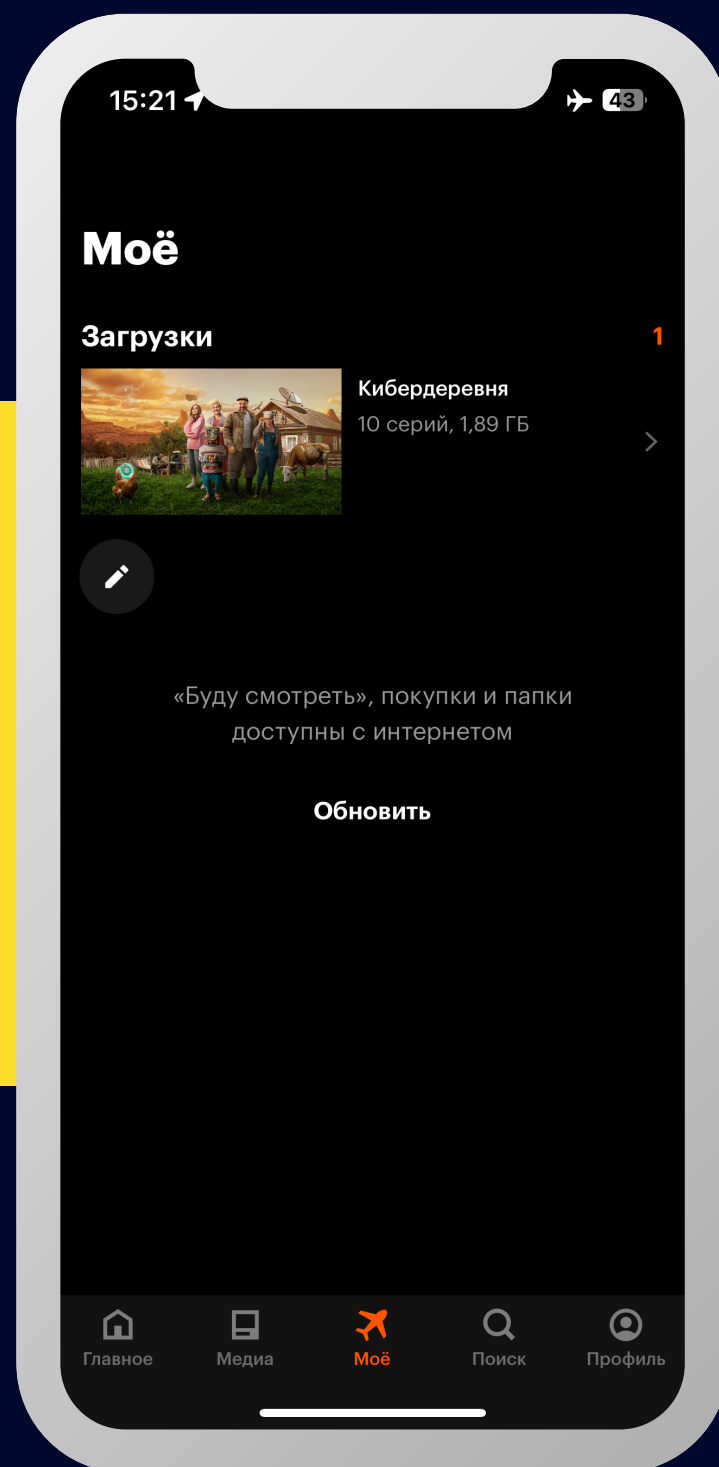
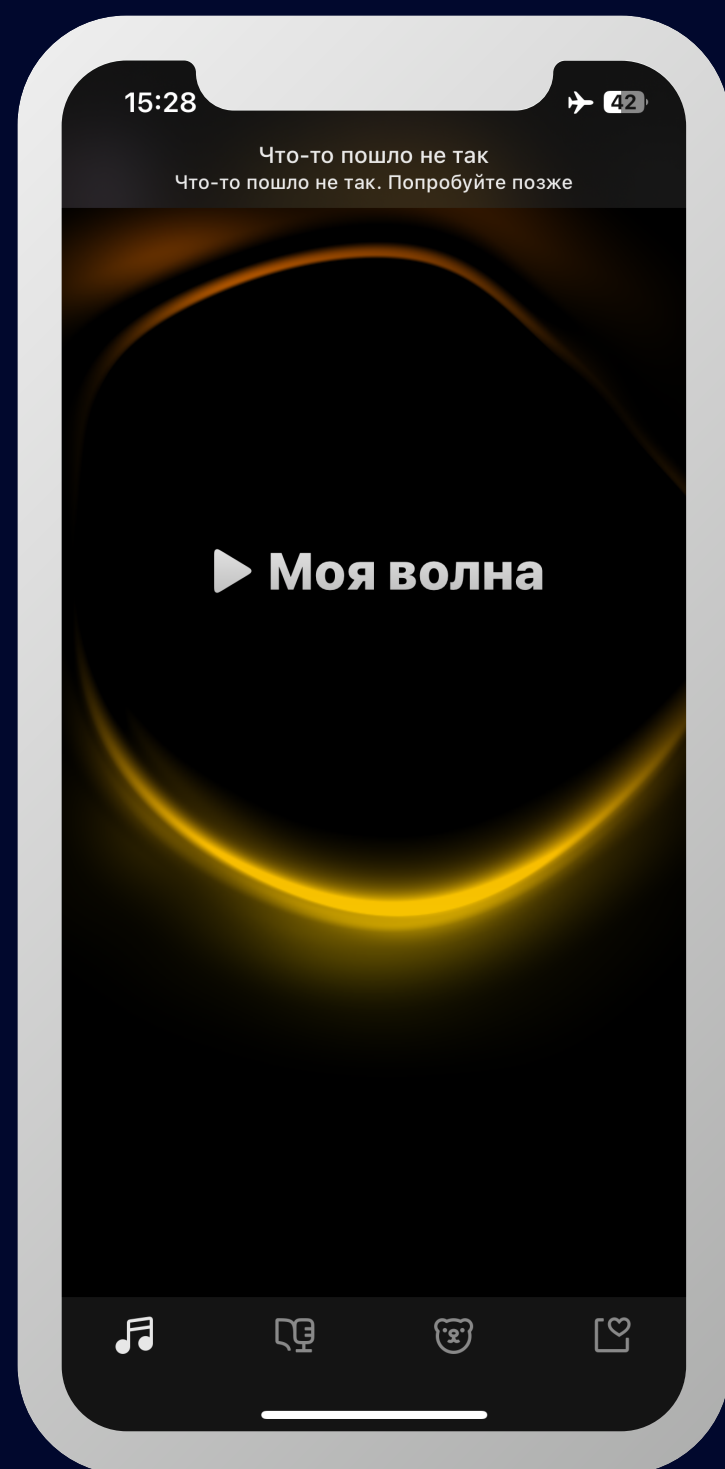
Когда нет интернета?

- Самолет
- Кассы и пункты выдачи
- Лифт

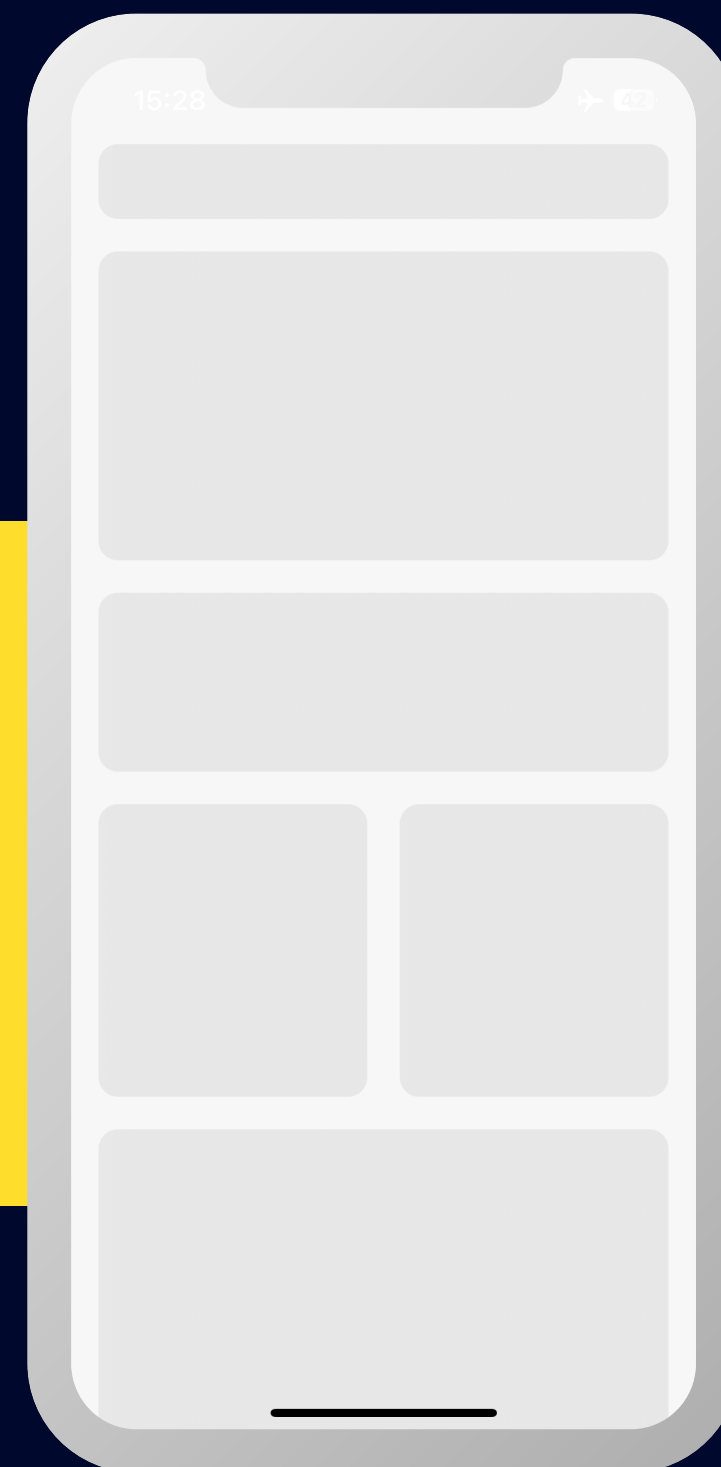
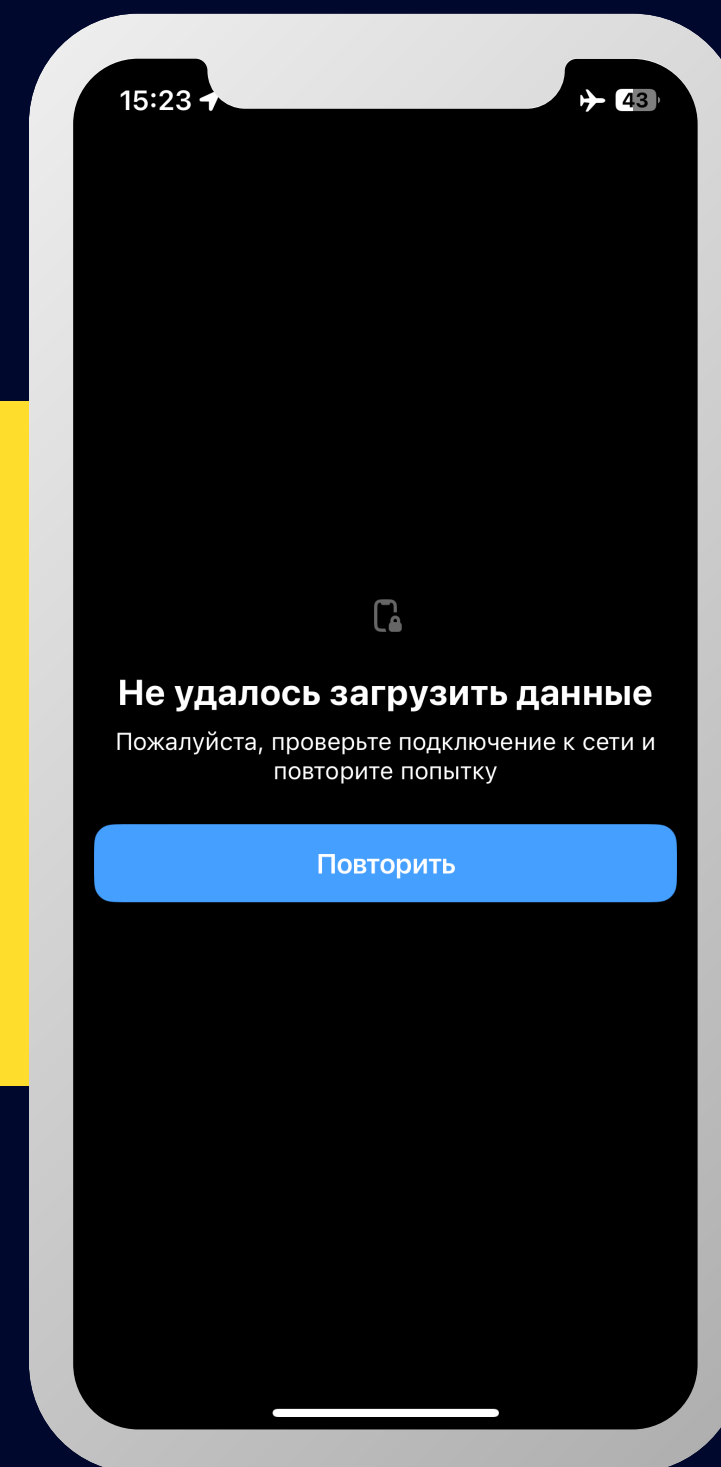
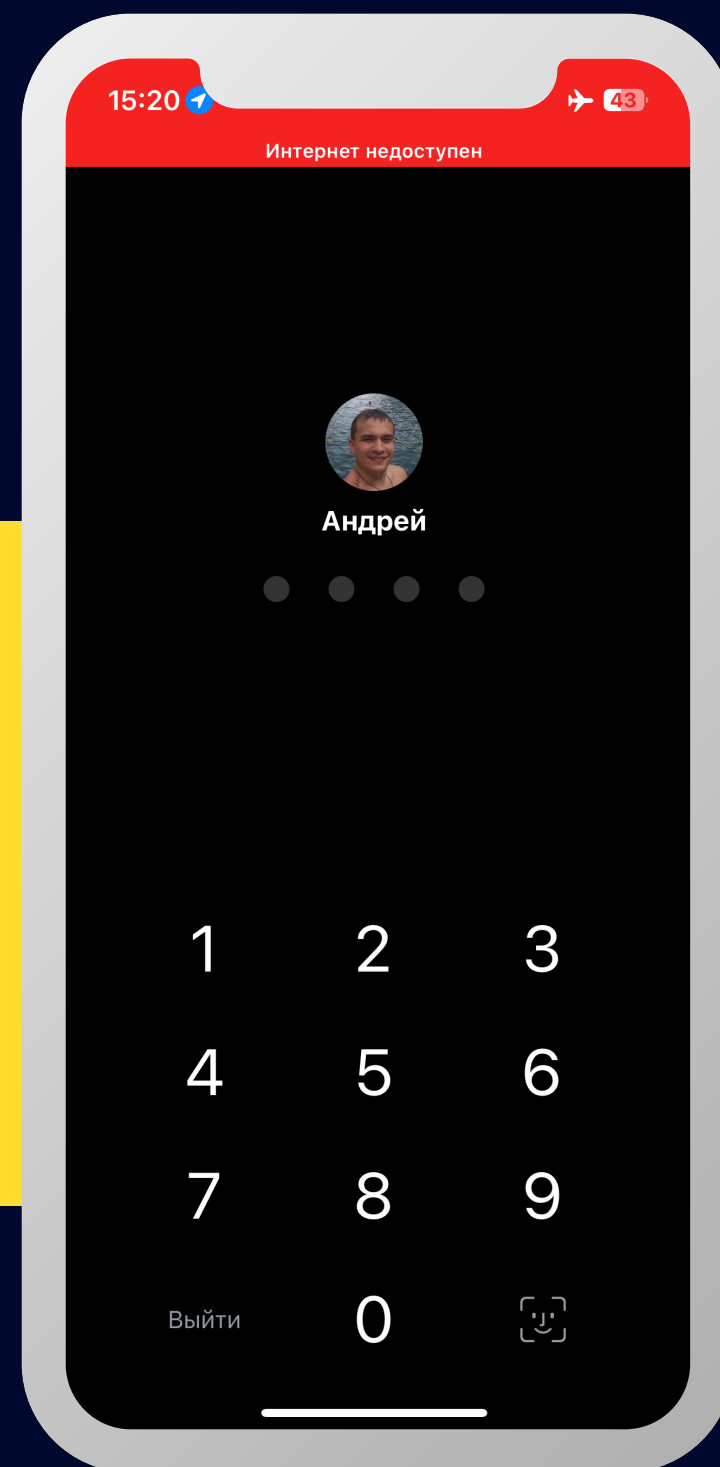
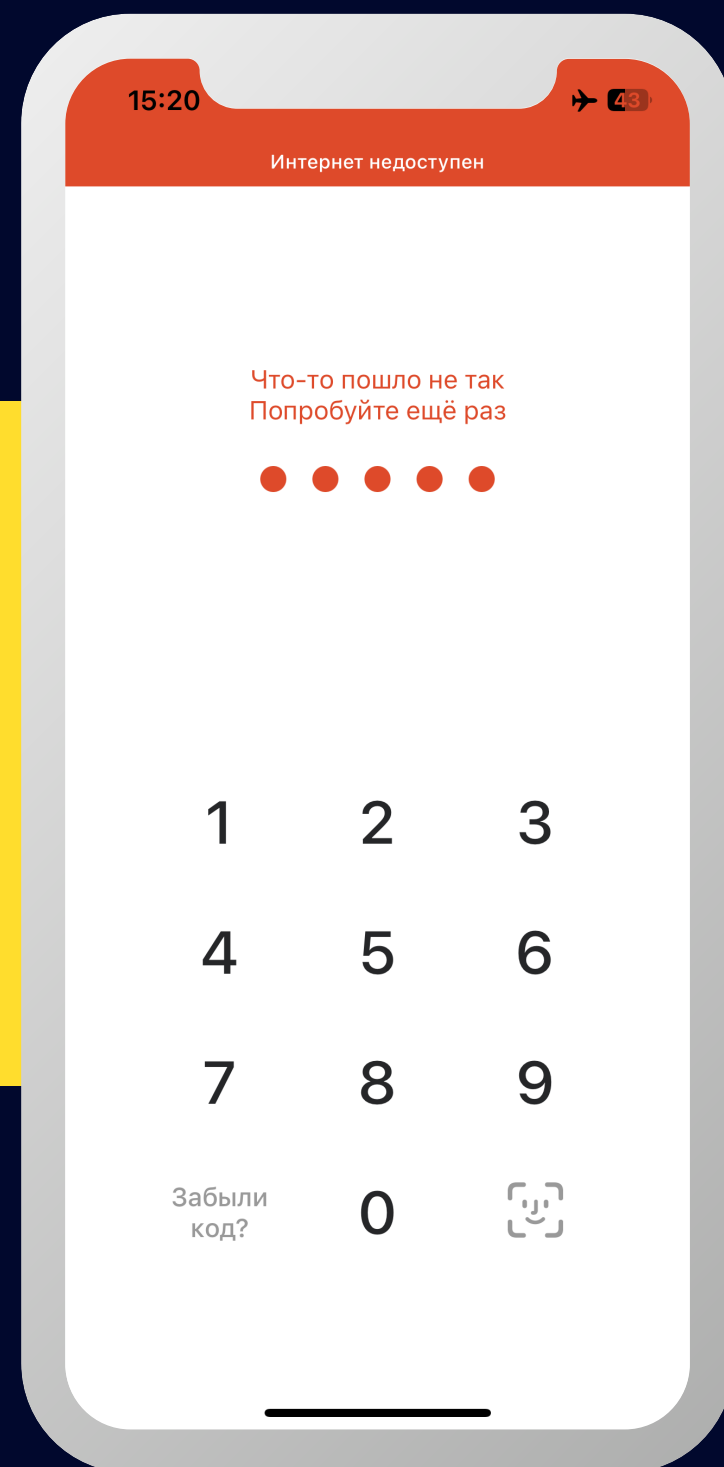
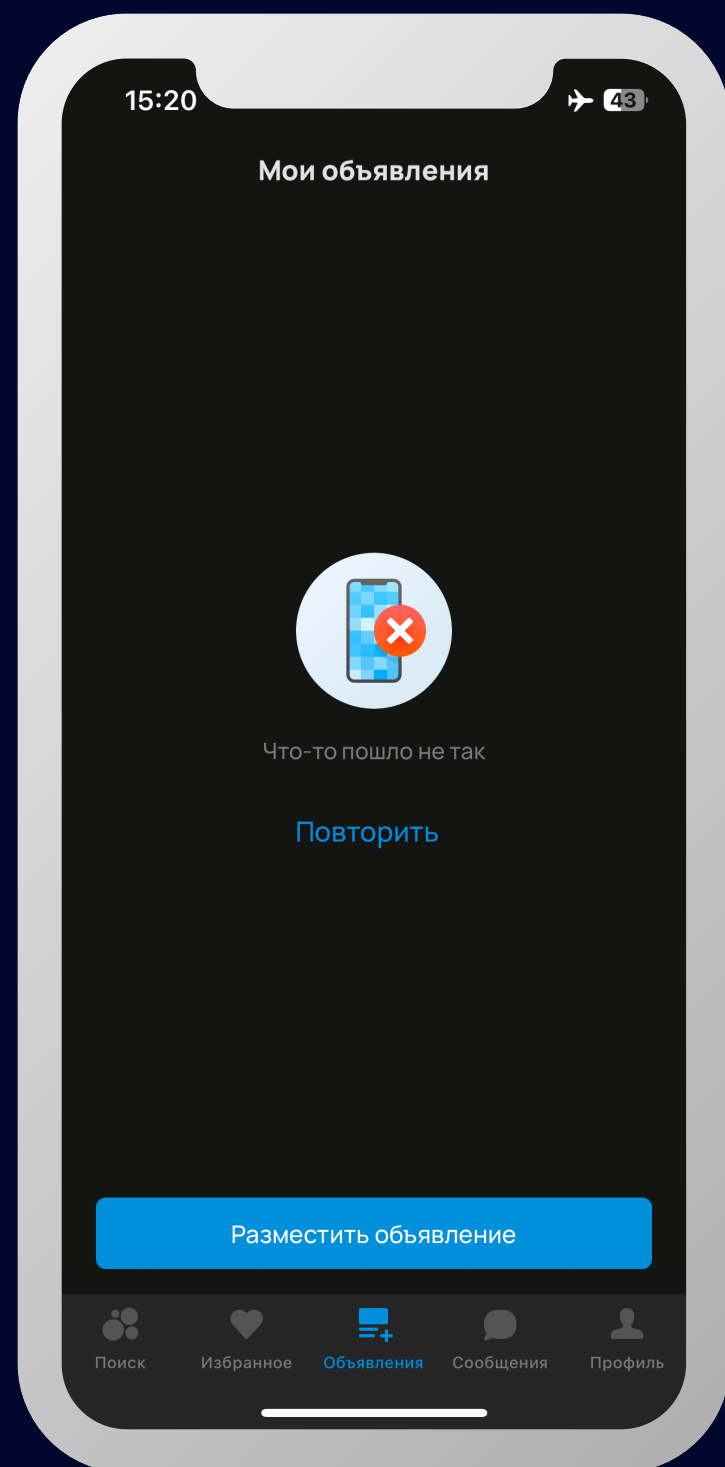
Доступные без сети



Ограниченно доступные



Не доступные



Зачем сеть?

Зачем сеть?

Получение
и обновления
данных



Зачем сеть?

Получение
и обновления
данных



Централизованное
место вычислений
и логики



Зачем сеть?

Получение
и обновления
данных



Централизованное
место вычислений
и логики



Хранение
данных



Зачем кеш?

Загрузка экрана

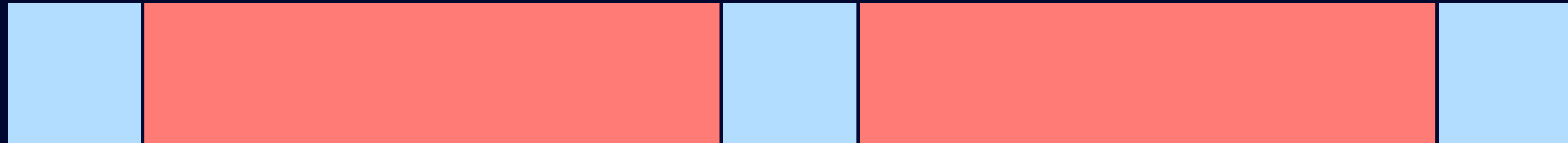
CPU

Сеть

CPU

Сеть

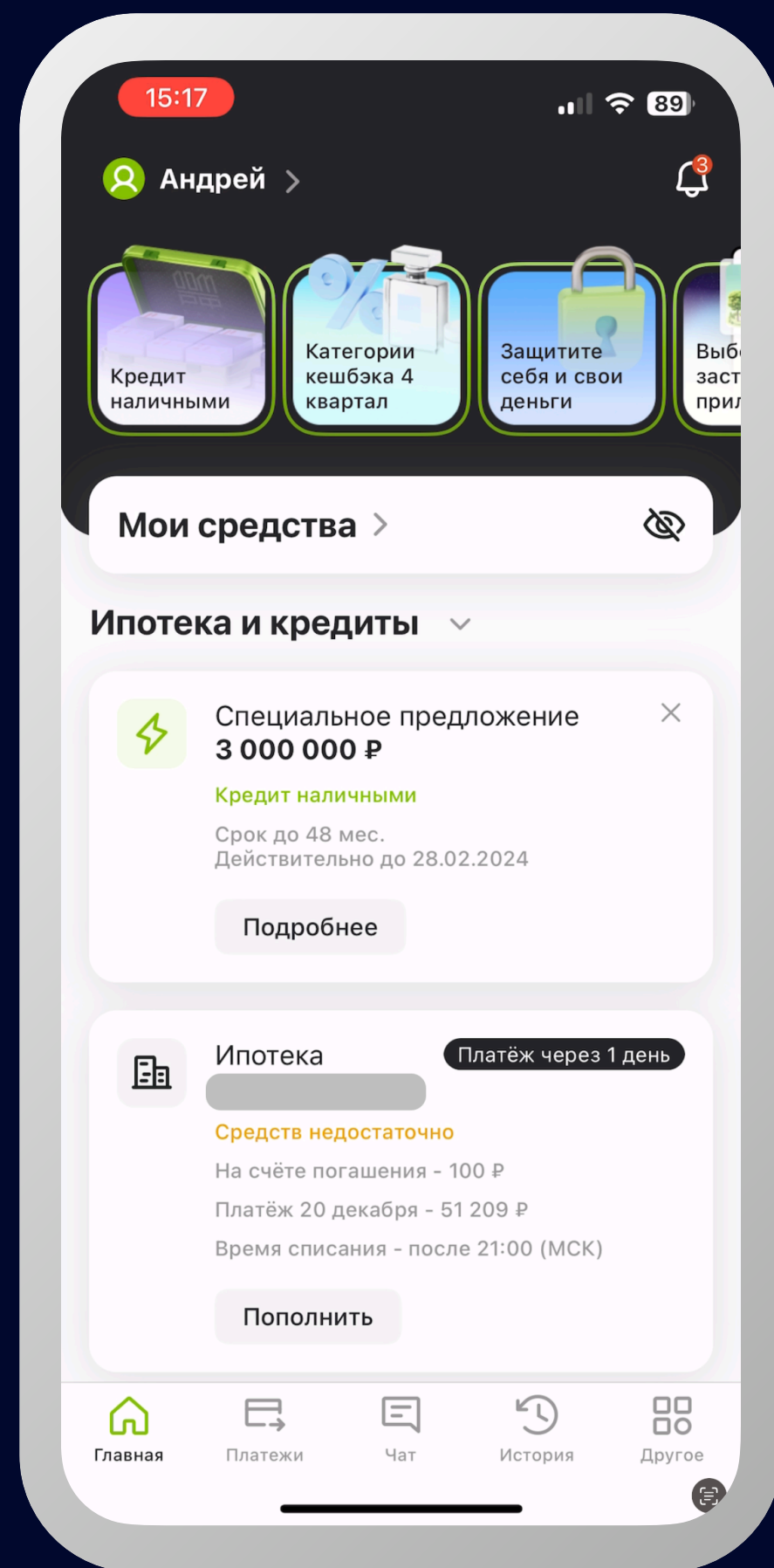
CPU



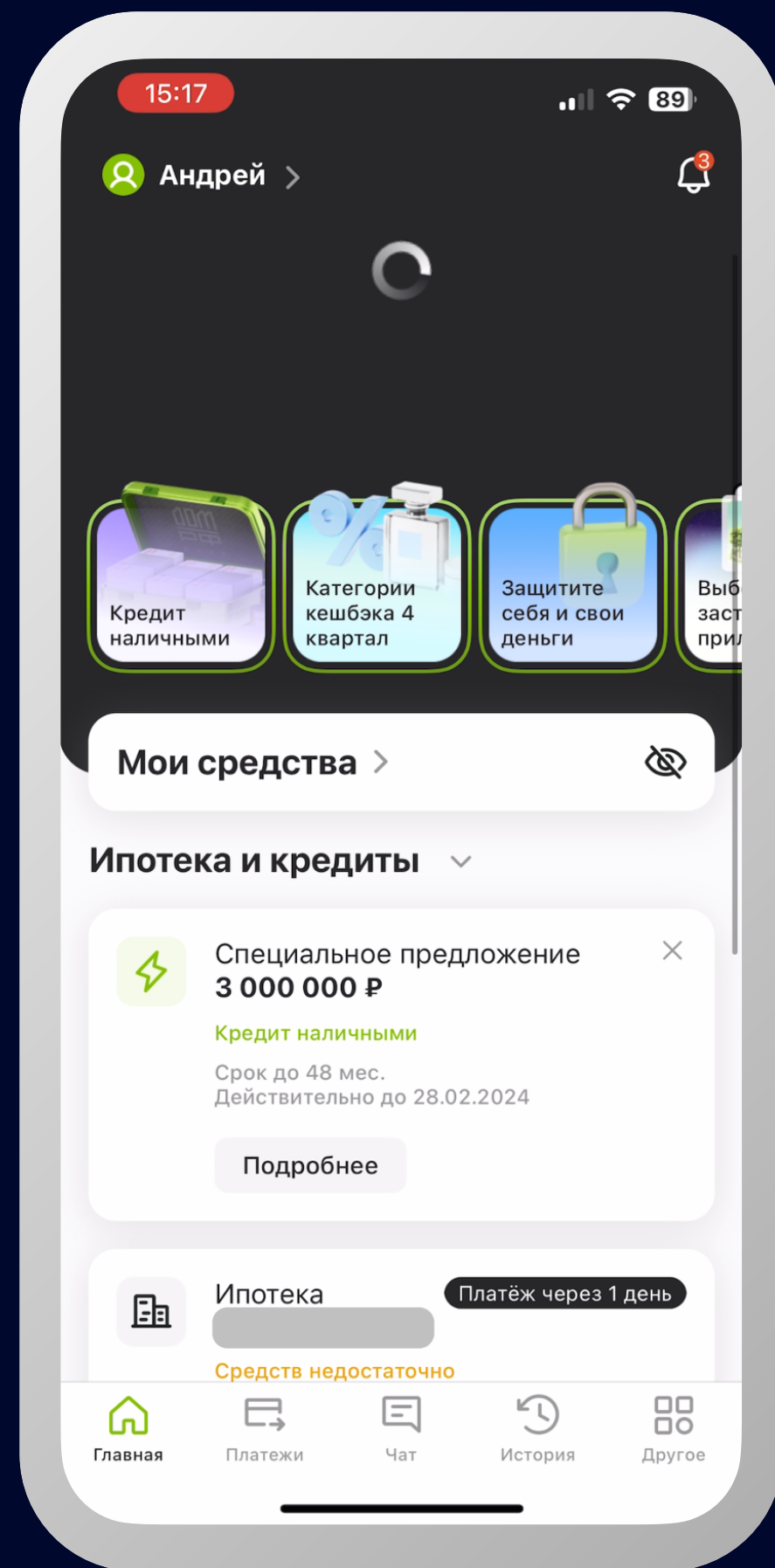
Загрузка экрана

CPU

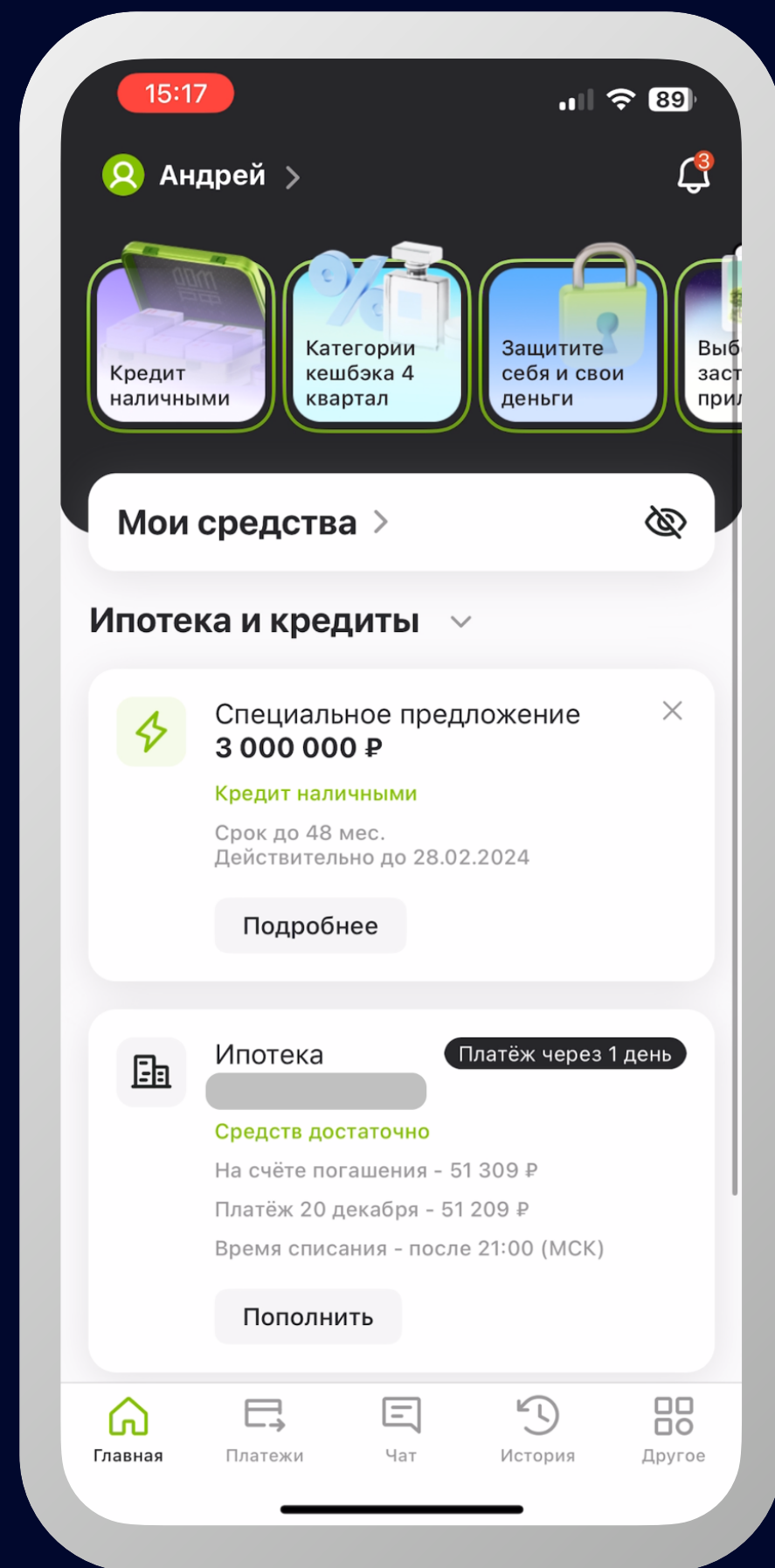




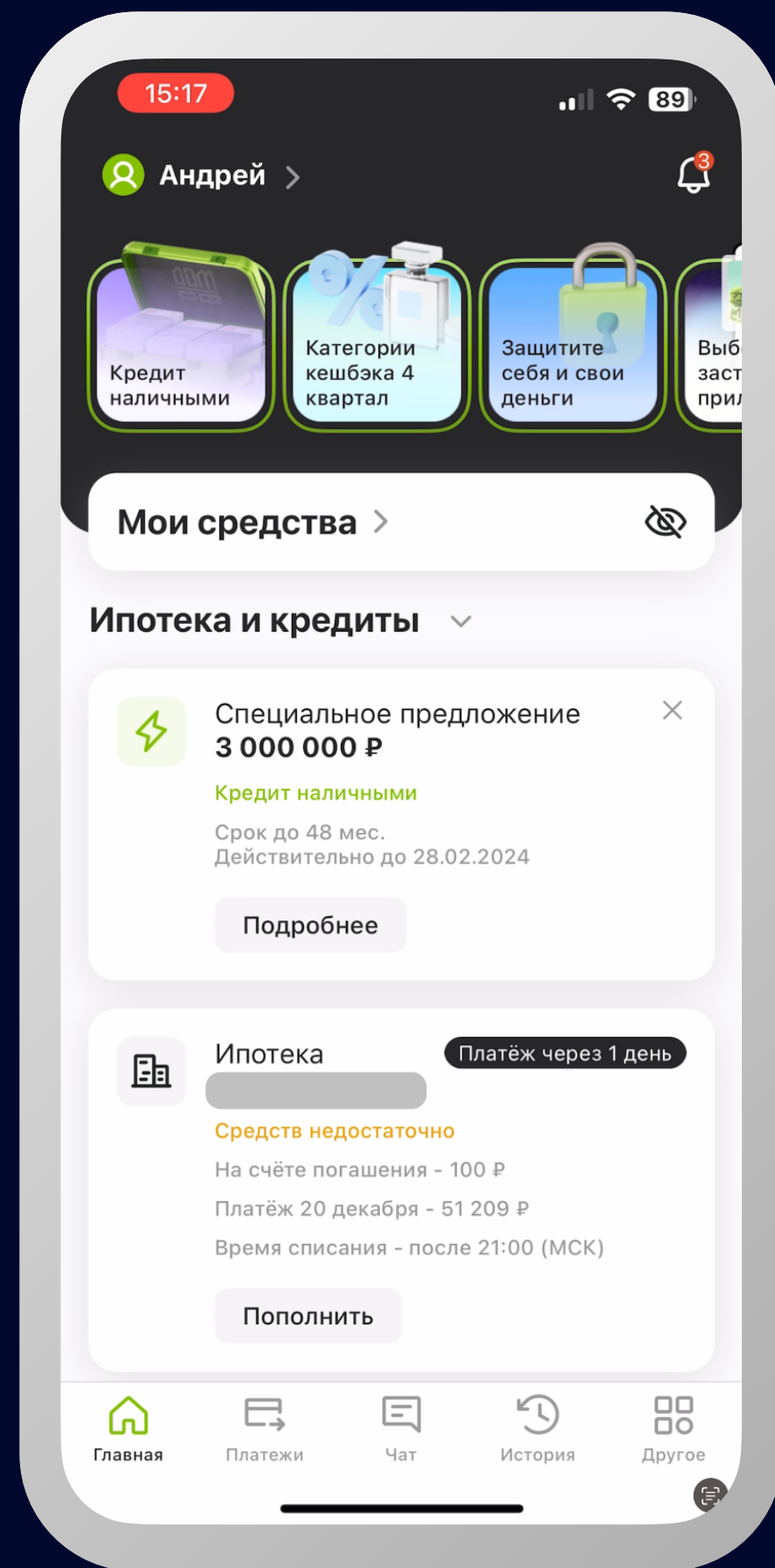
UX ПОЛЬЗОВАТЕЛЯ С СЕТЬЮ



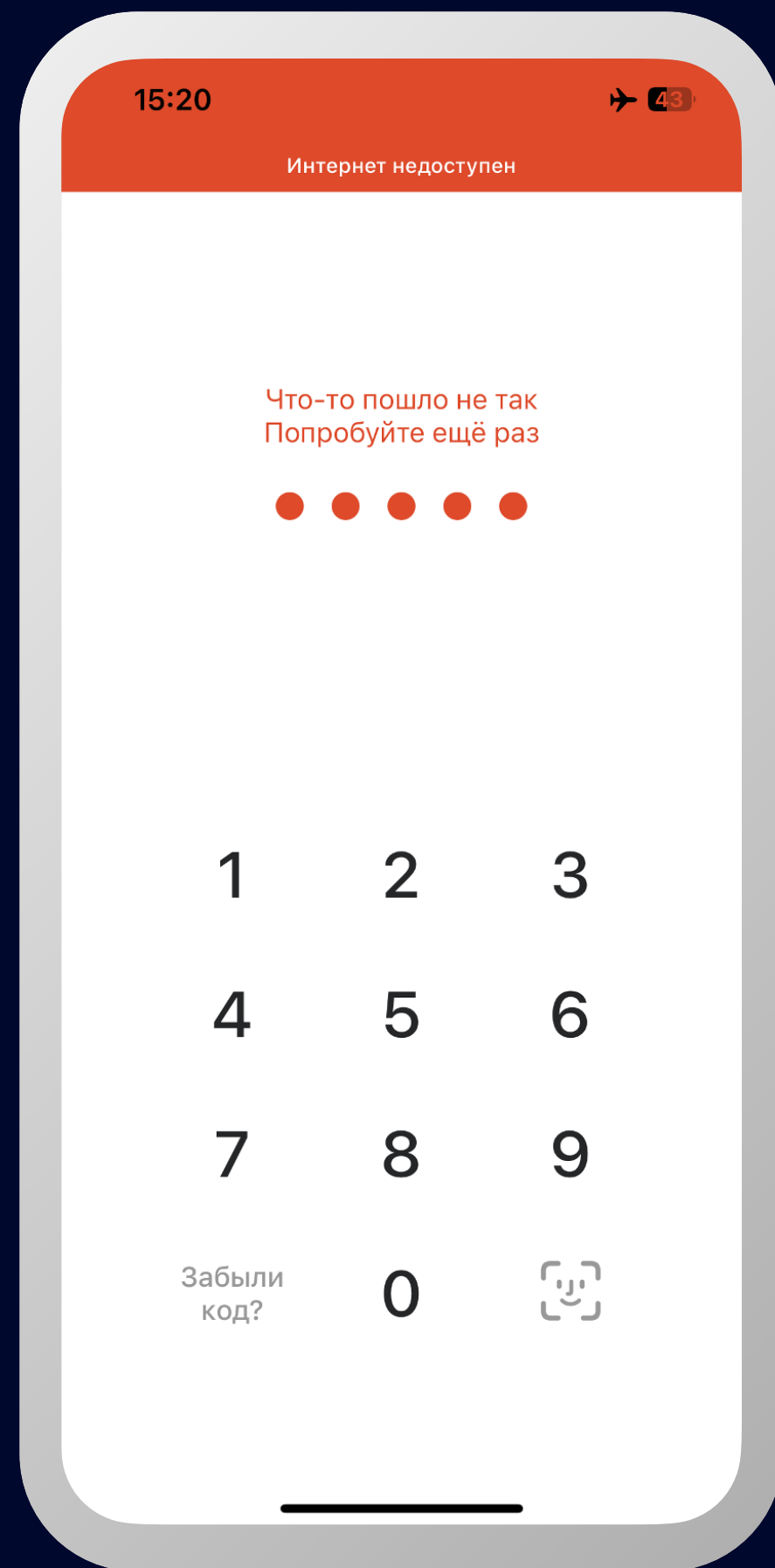
UX ПОЛЬЗОВАТЕЛЯ С СЕТЬЮ



UX ПОЛЬЗОВАТЕЛЯ С СЕТЬЮ



UX ПОЛЬЗОВАТЕЛЯ С СЕТЬЮ



Без сети

Загрузка экрана

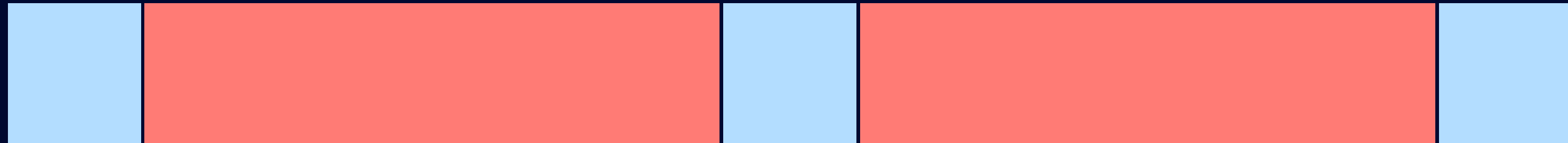
CPU

Сеть

CPU

Сеть

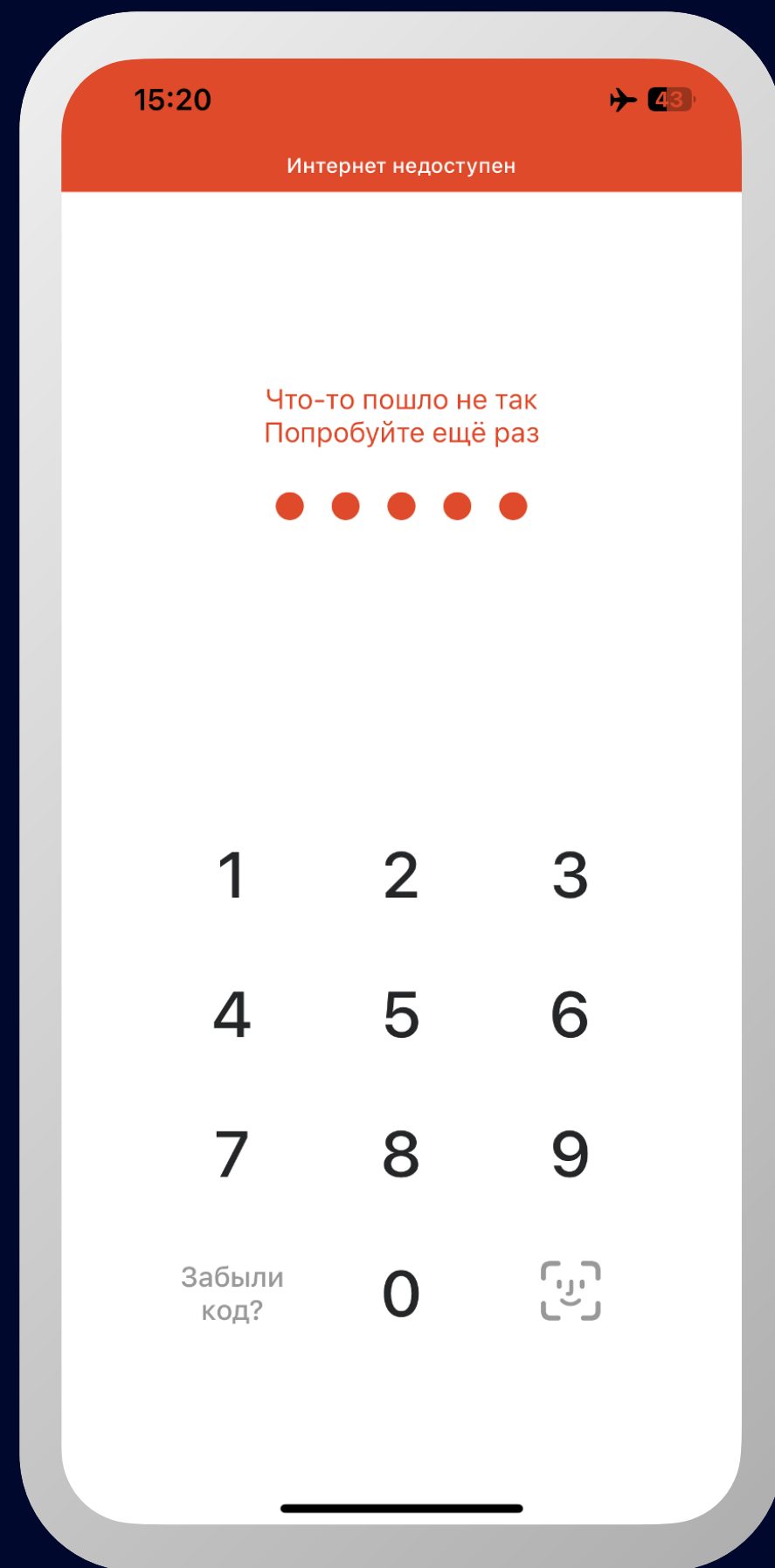
CPU



Загрузка экрана

CPU





Без сети

Кеш

Персистентное хранилище

Экономить время загрузки экрана

Структурирование хранилище

Снижает нагрузку сервера

Возможность делать запросы

Временное Хранилище

Поддерживает транзакционность и ACID



Кеш

Персистентное хранилище

NSURLCache
URLRequest.CachePolicy

CoreData

Codable+Filesystem

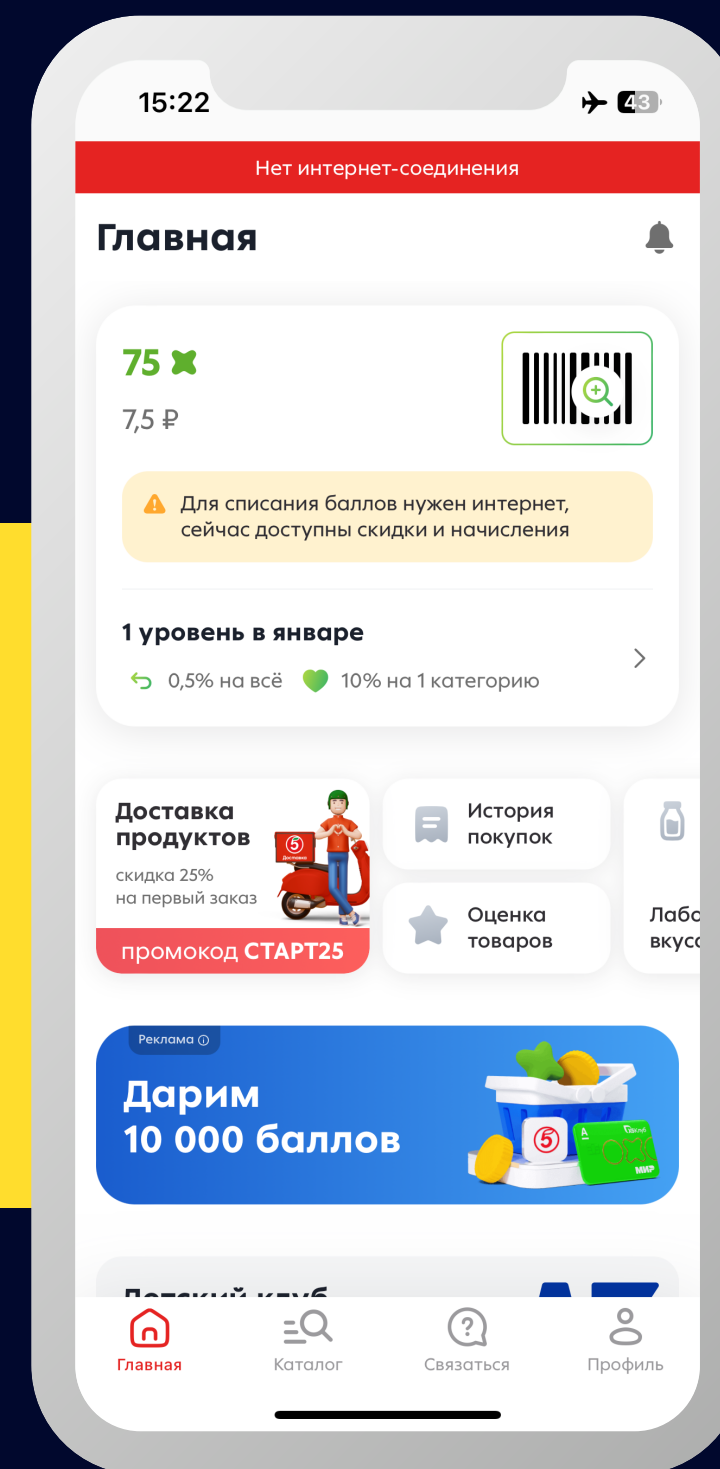
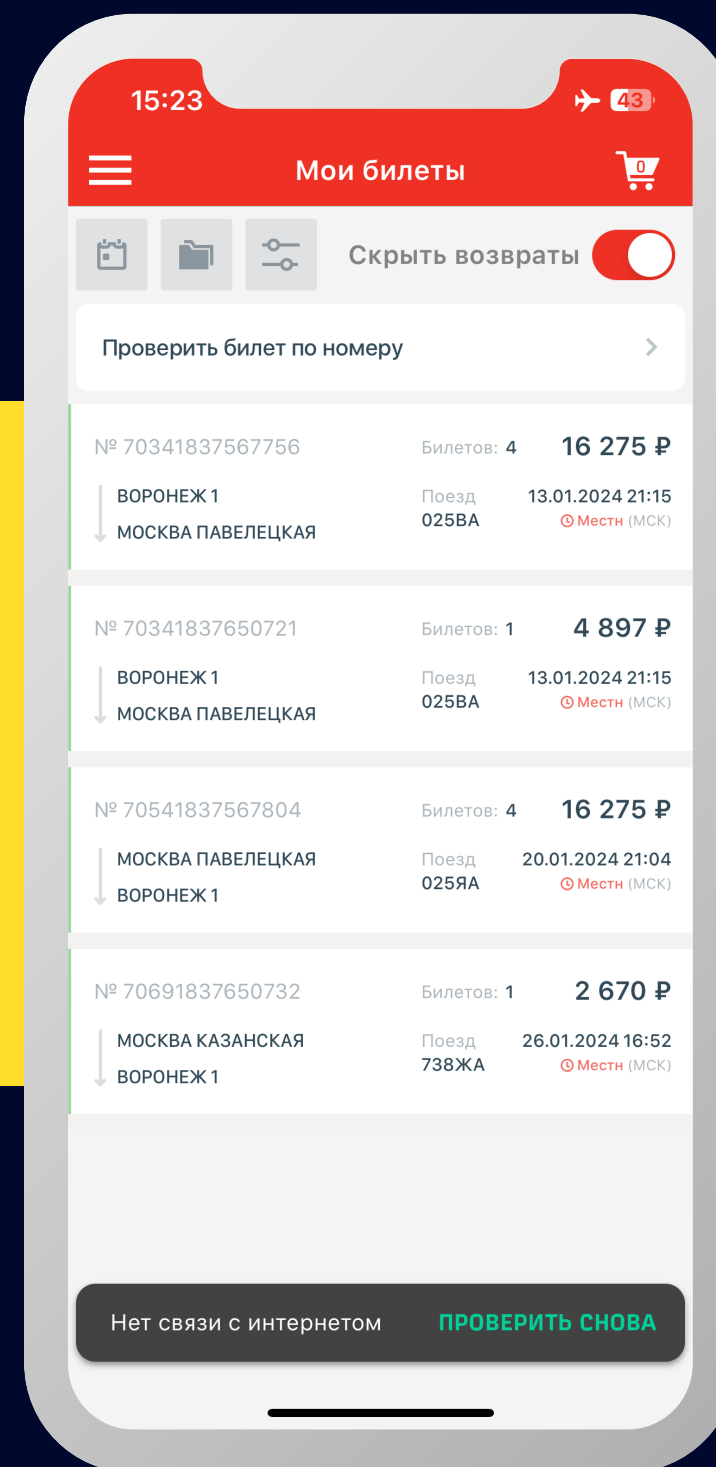
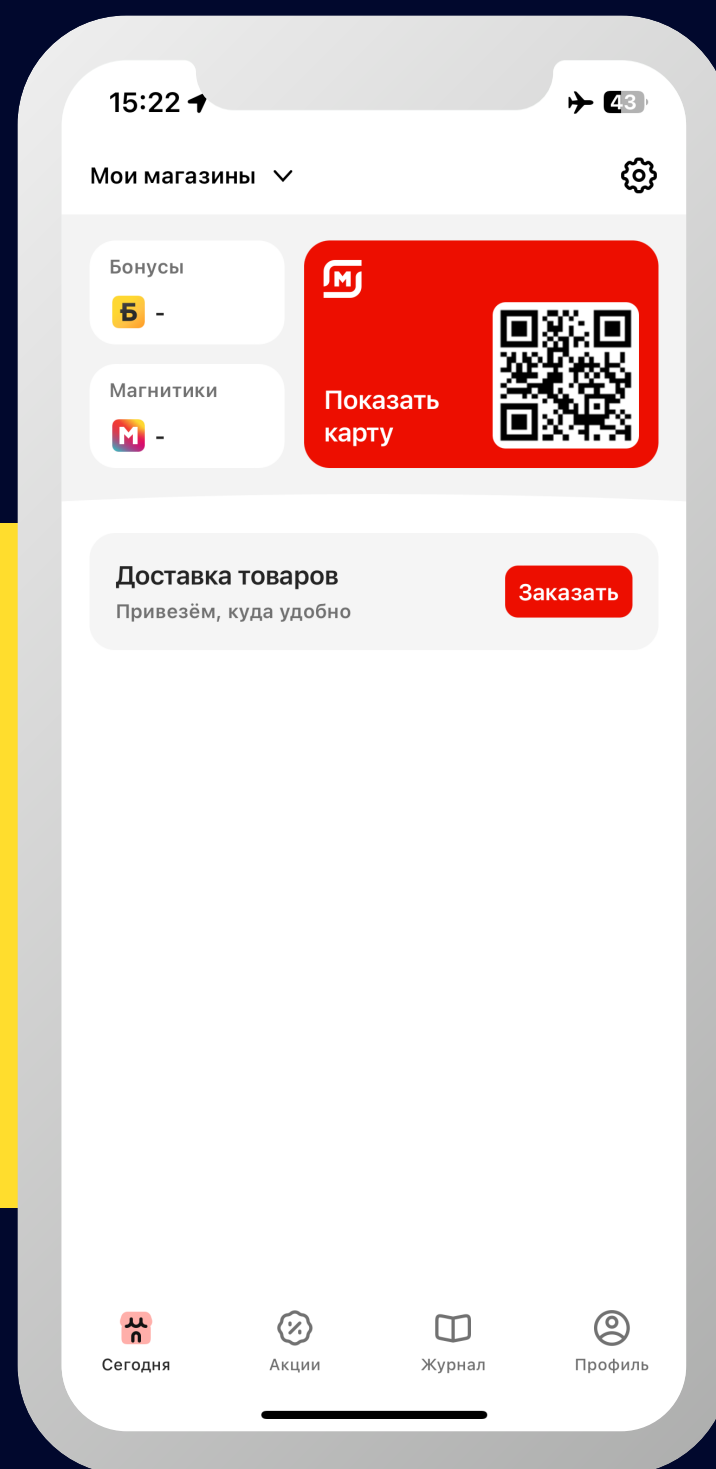
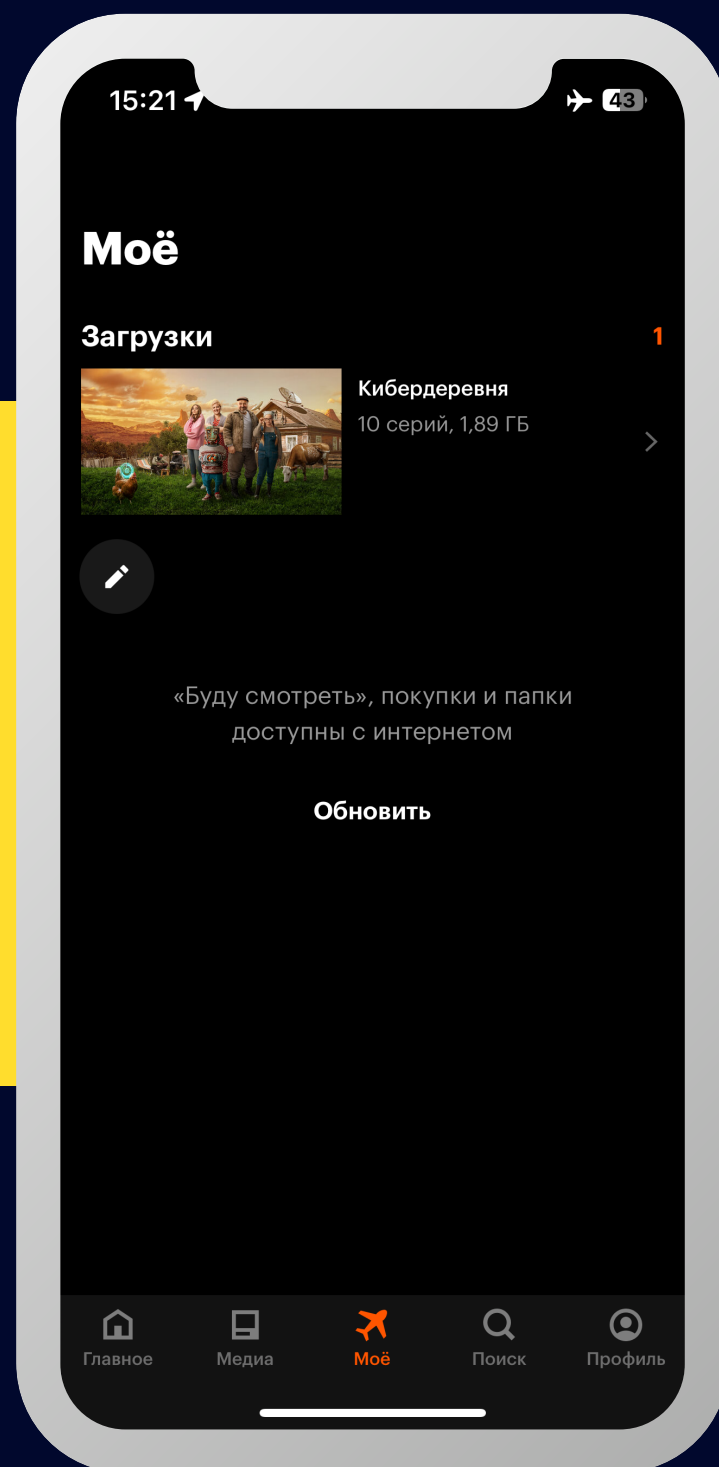
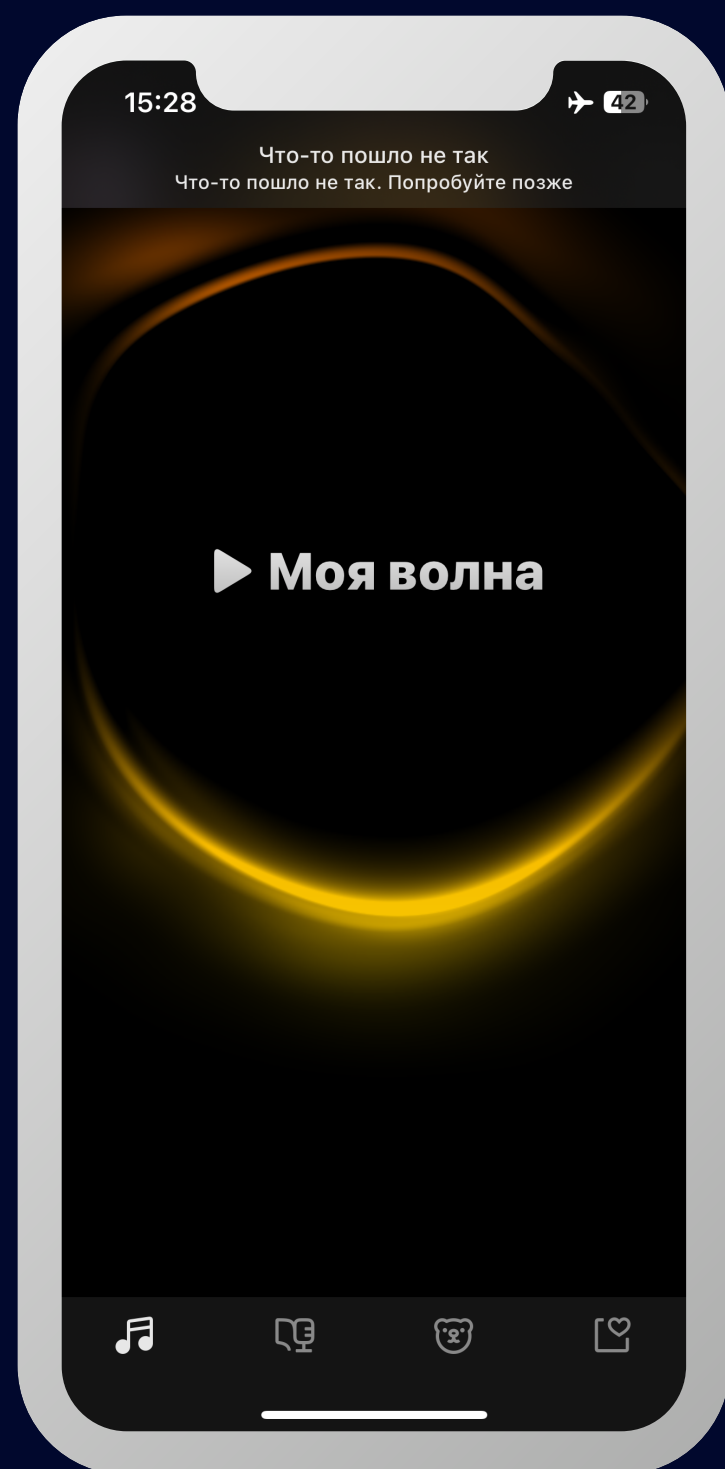
SwiftData

Codable+UserDefaults

SQLite



Ограниченно доступные



Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults



Сервер

Клиент

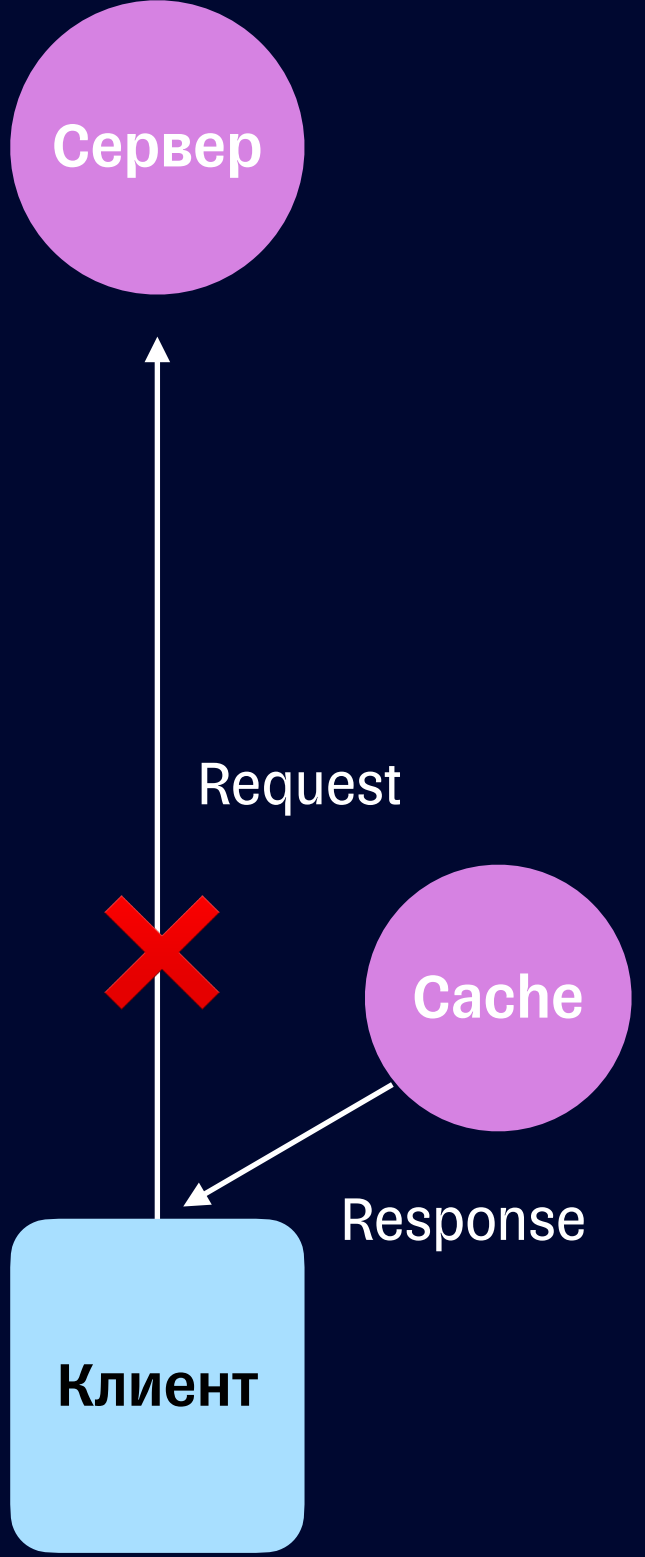
Request

Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults

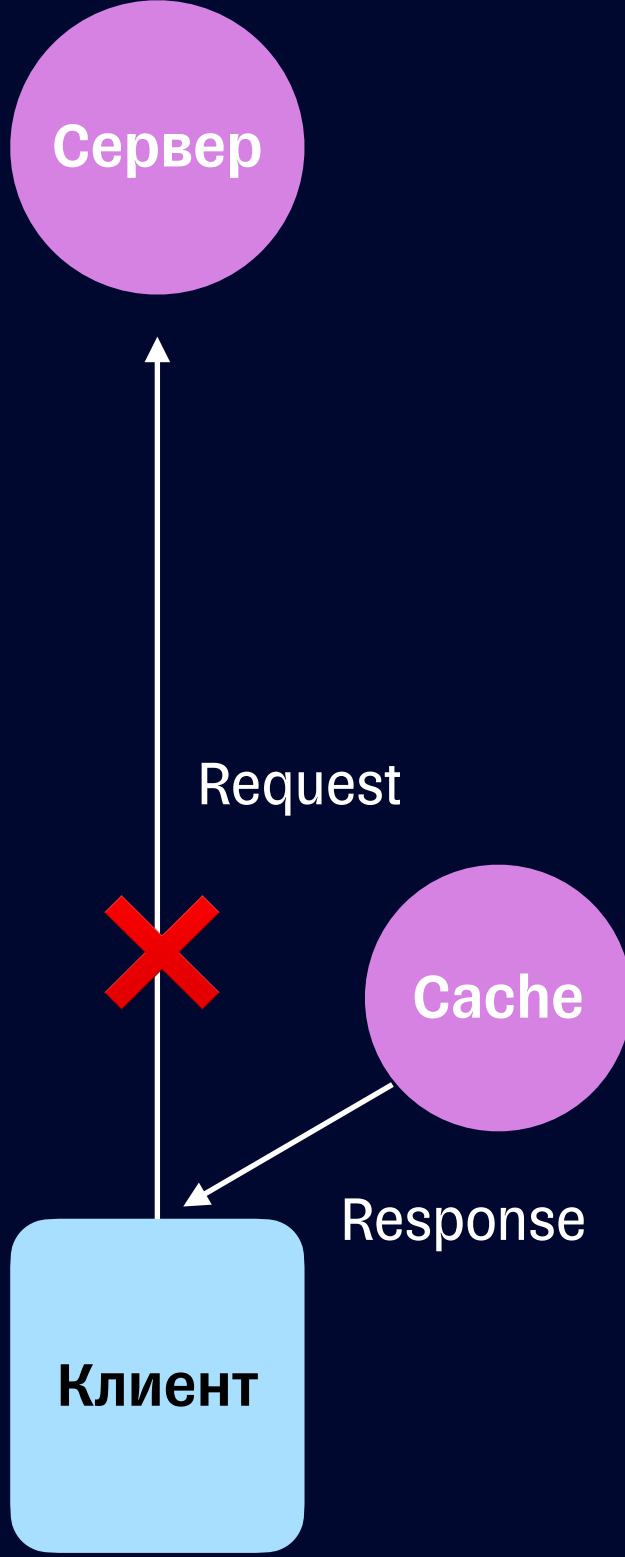


Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults



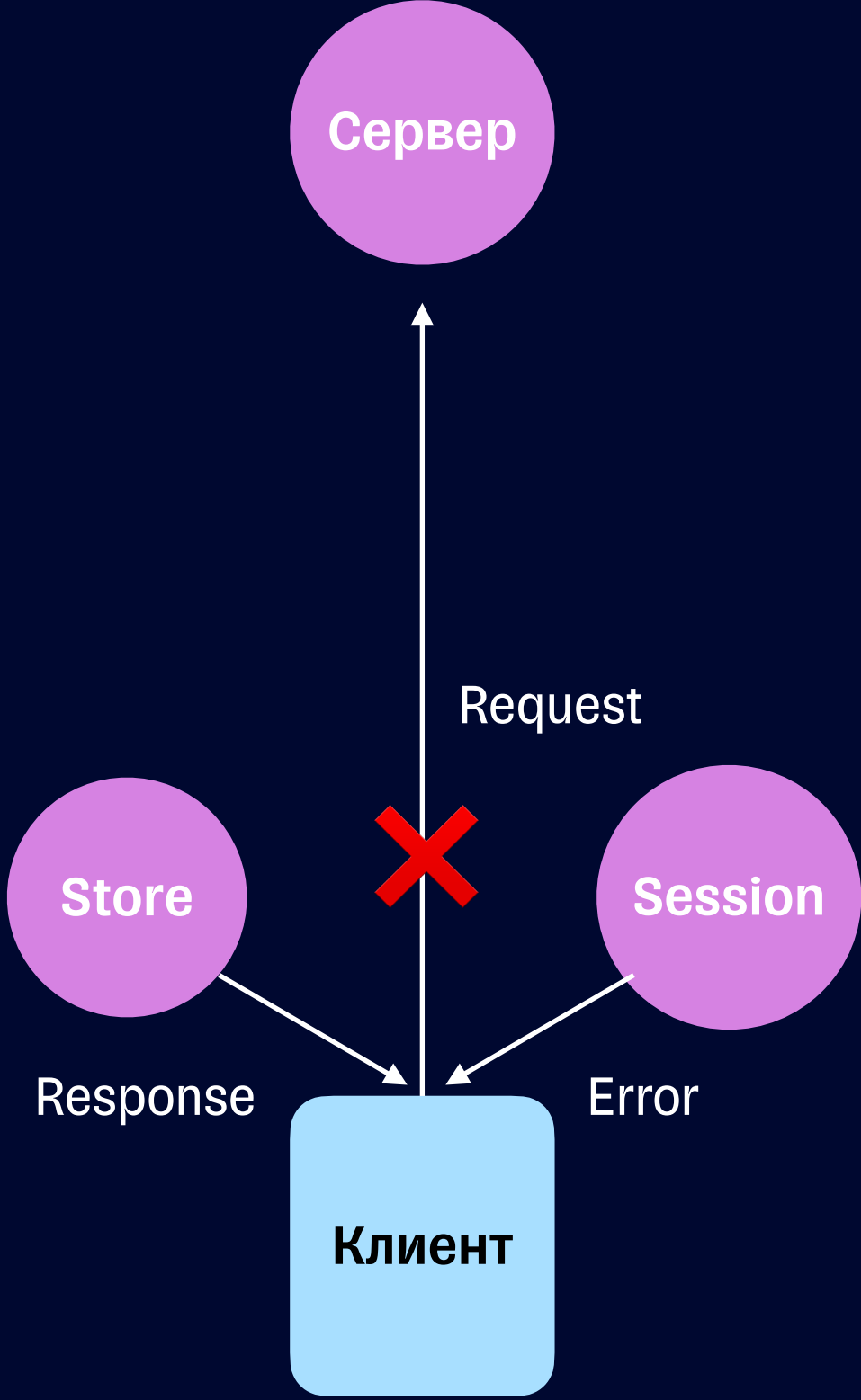
В чем тут проблема?

Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults



В чем тут проблема?

Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults



Сервер

Клиент

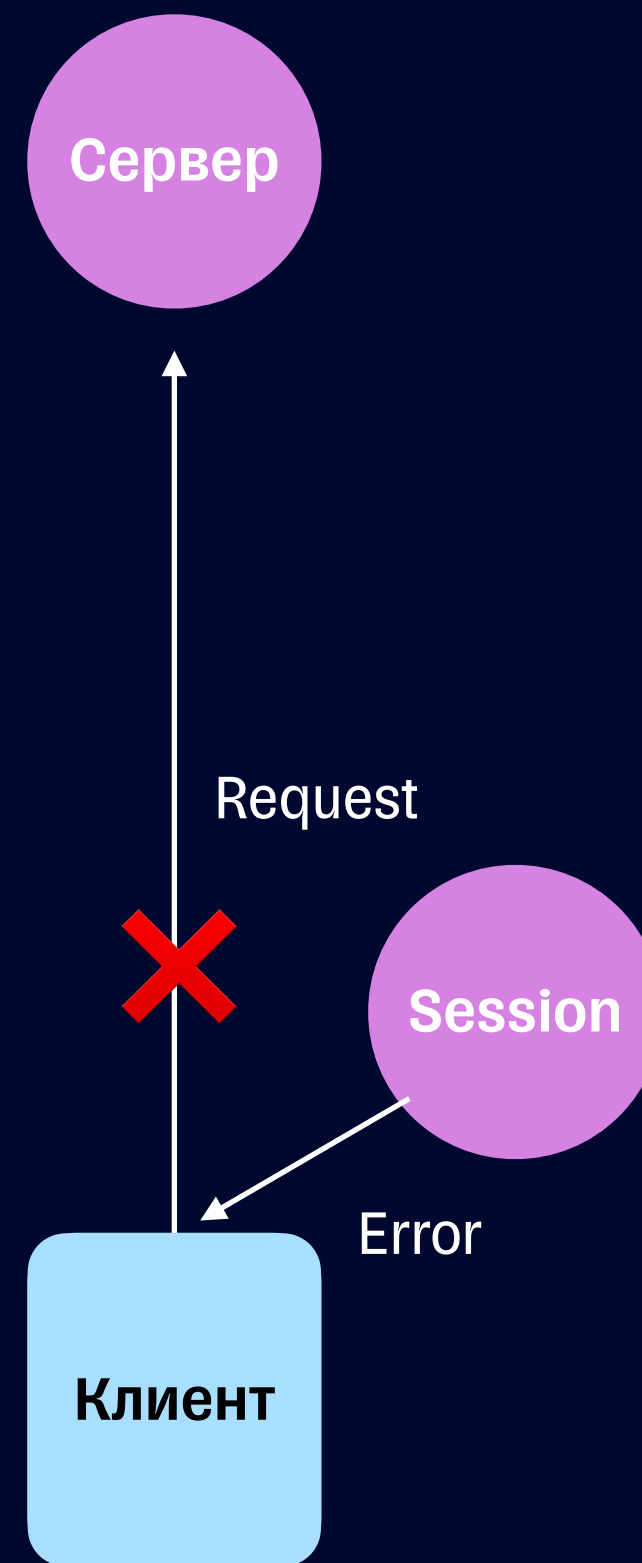


Кеш

NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

Codable+UserDefaults

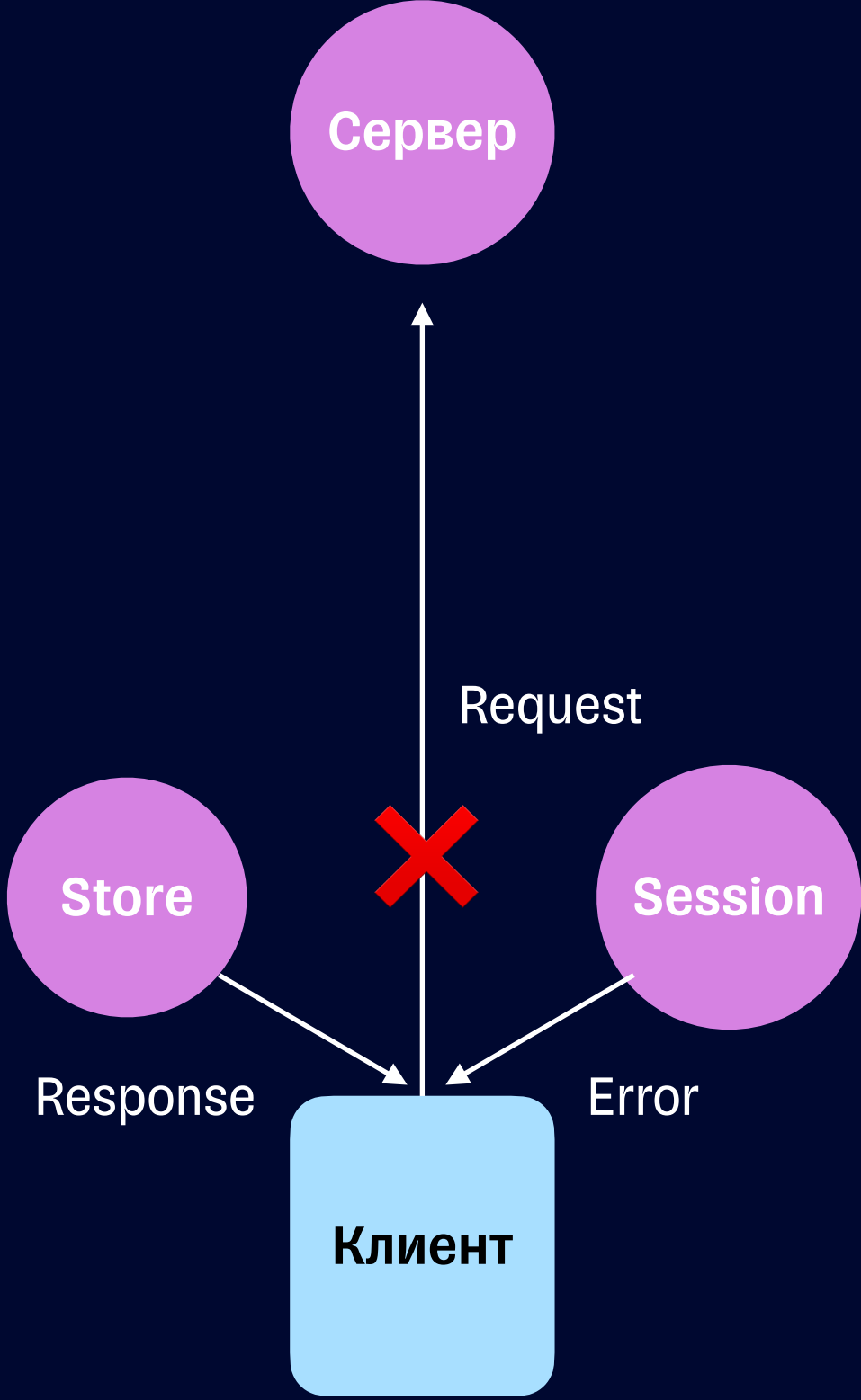


Кеш

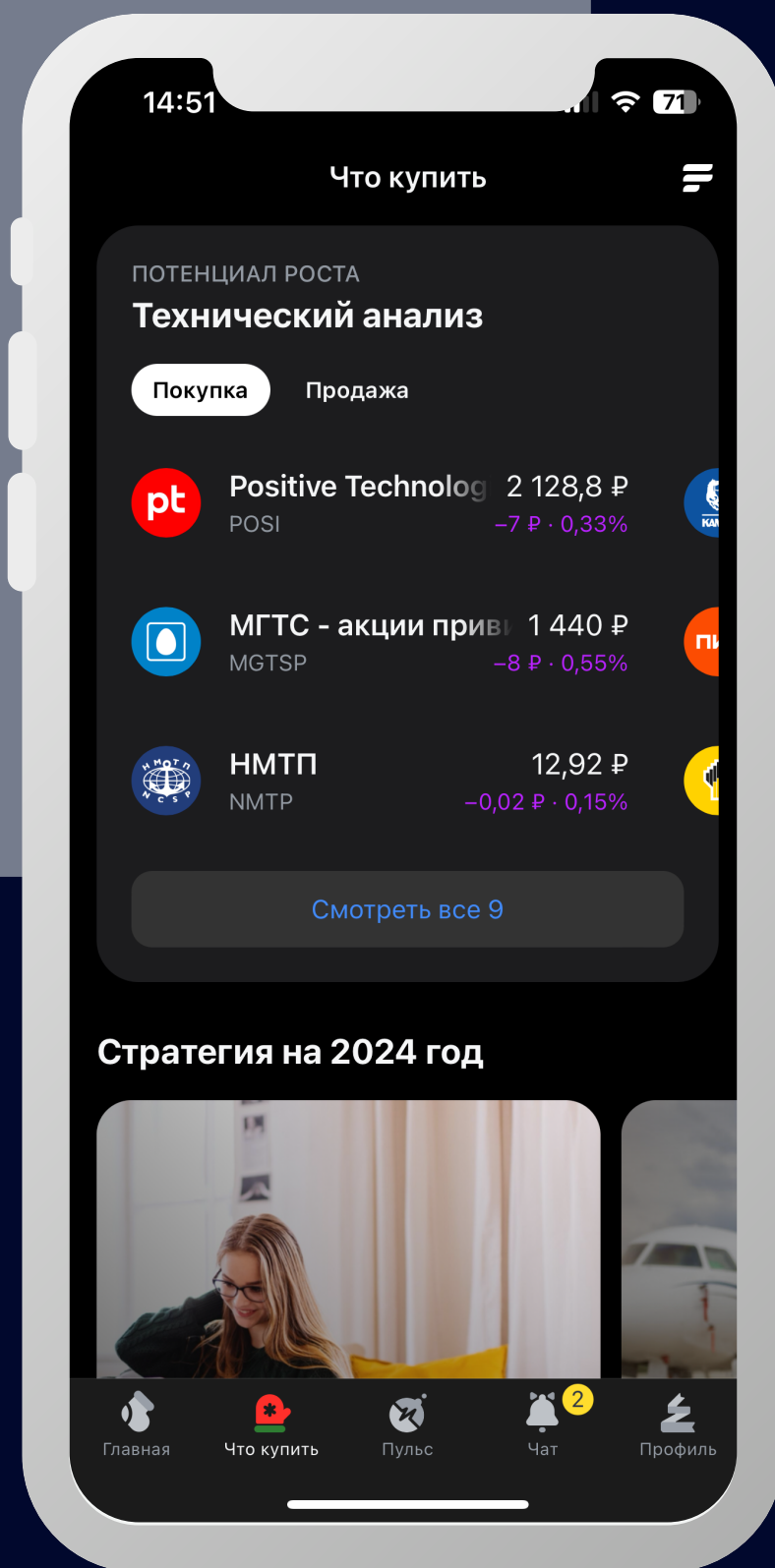
NSURLCache
URLRequest.CachePolicy

Codable+Filesystem

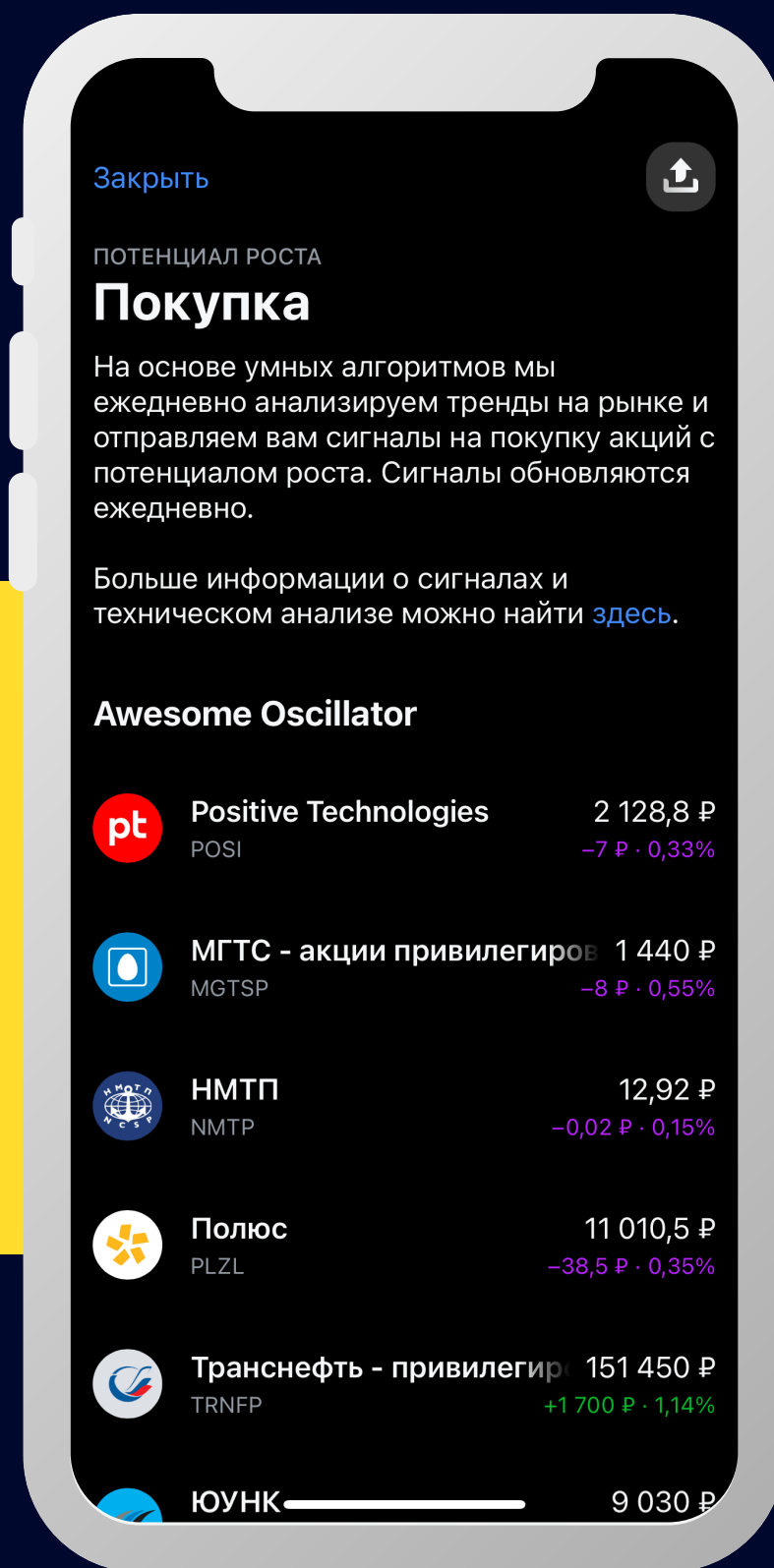
Codable+UserDefaults



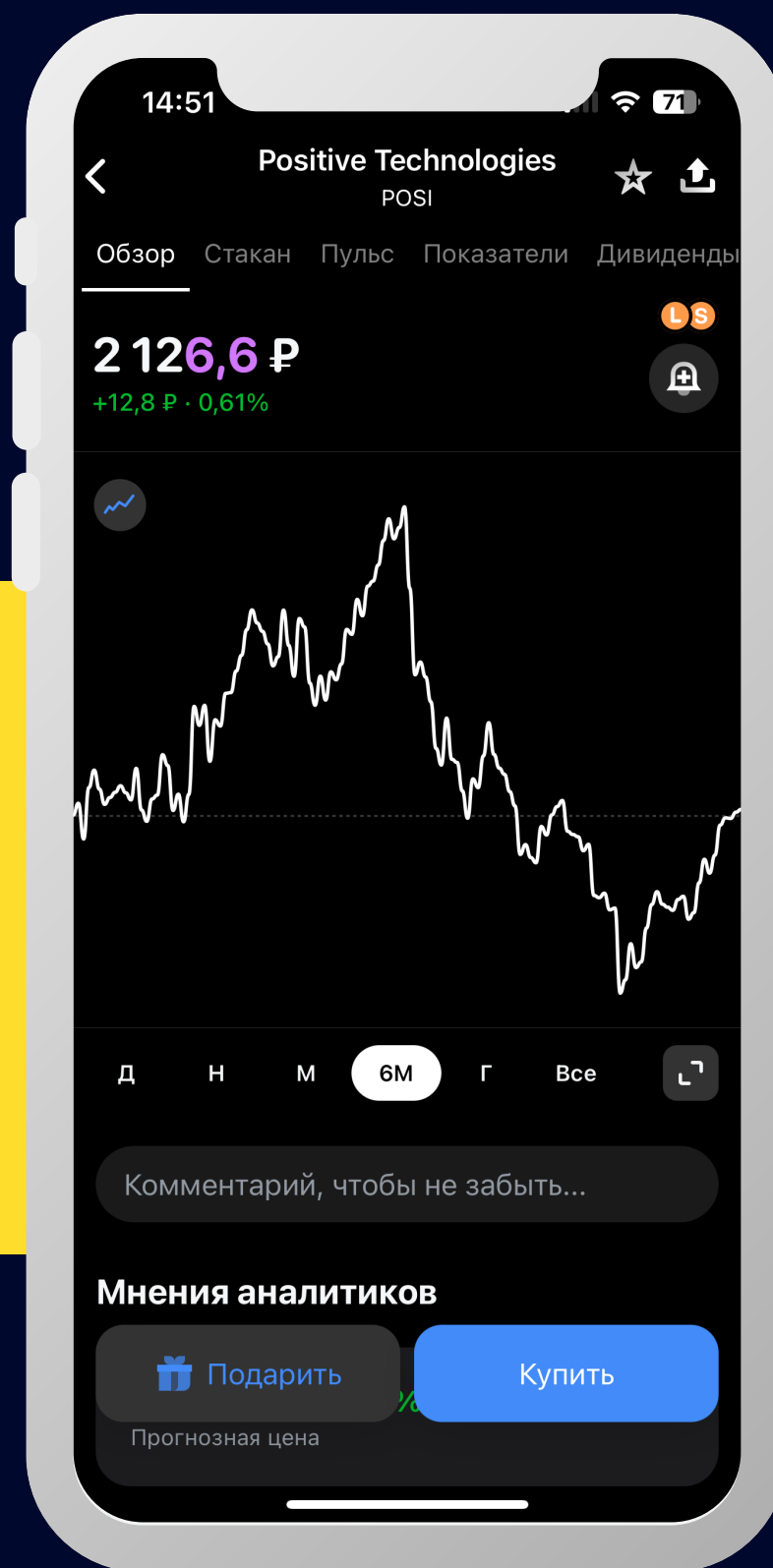
В чем тут проблема?



1

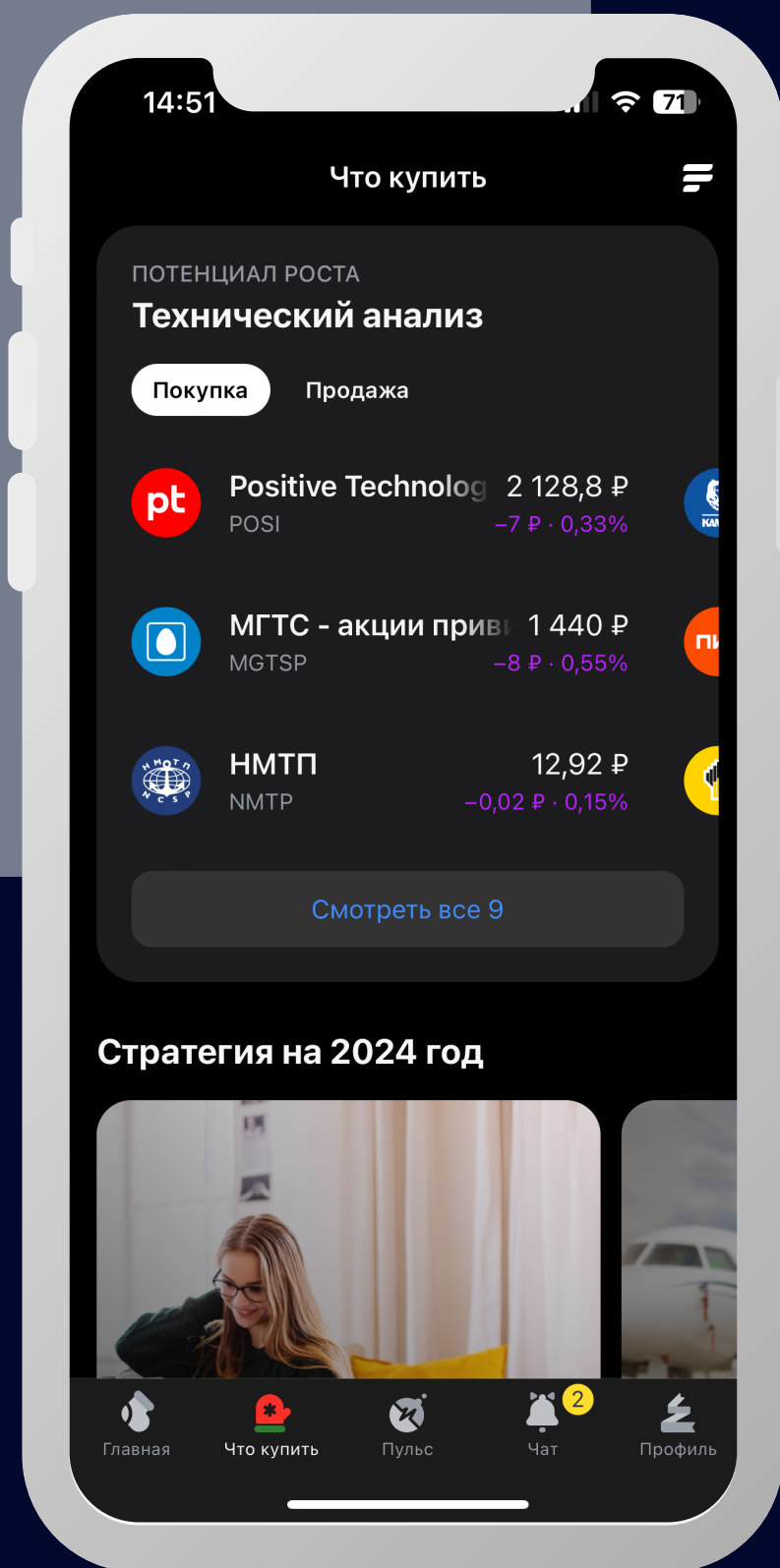


2



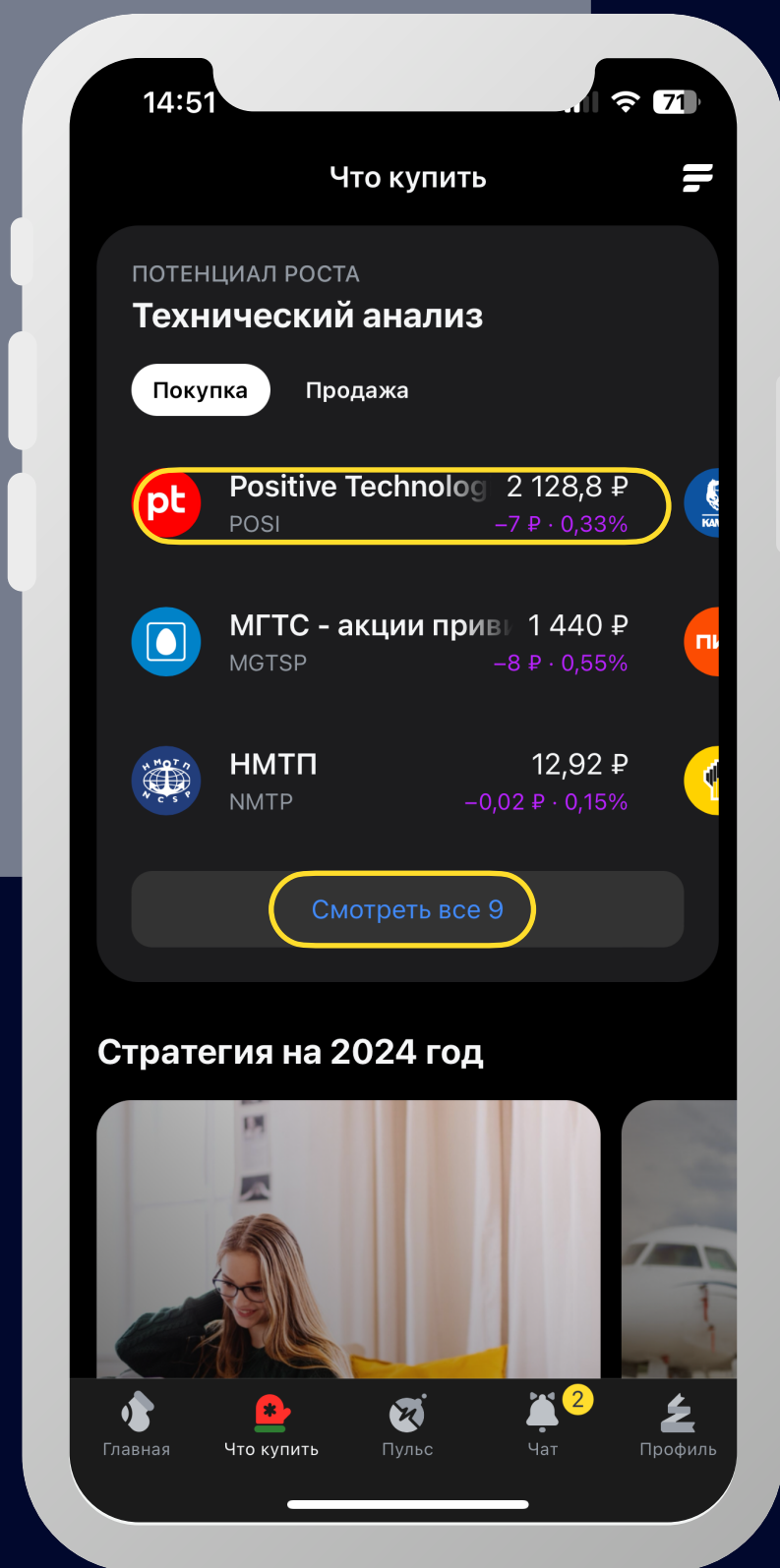
3

Консистентность экранов



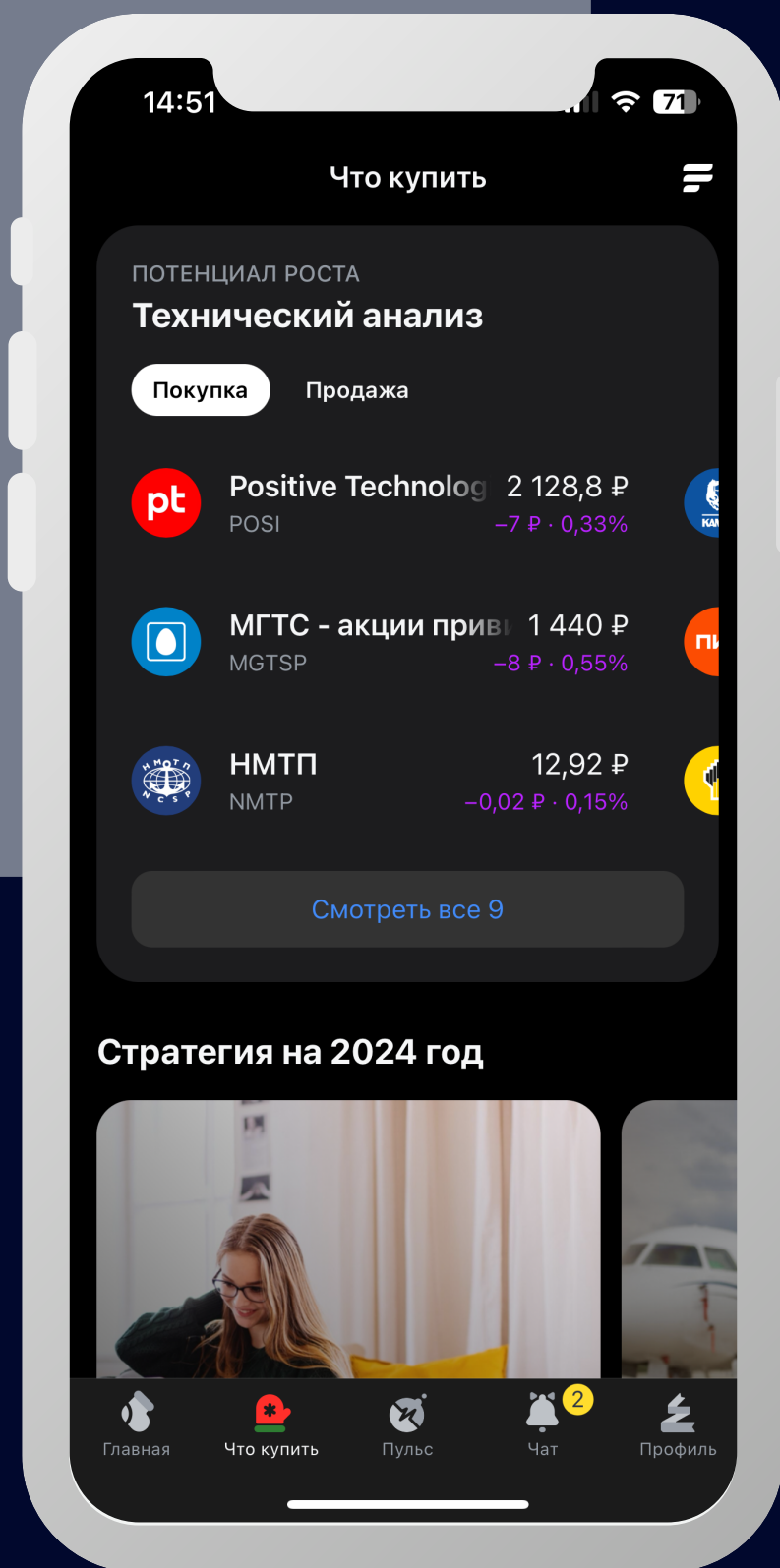
/widgets

ЭНДПОИНТЫ

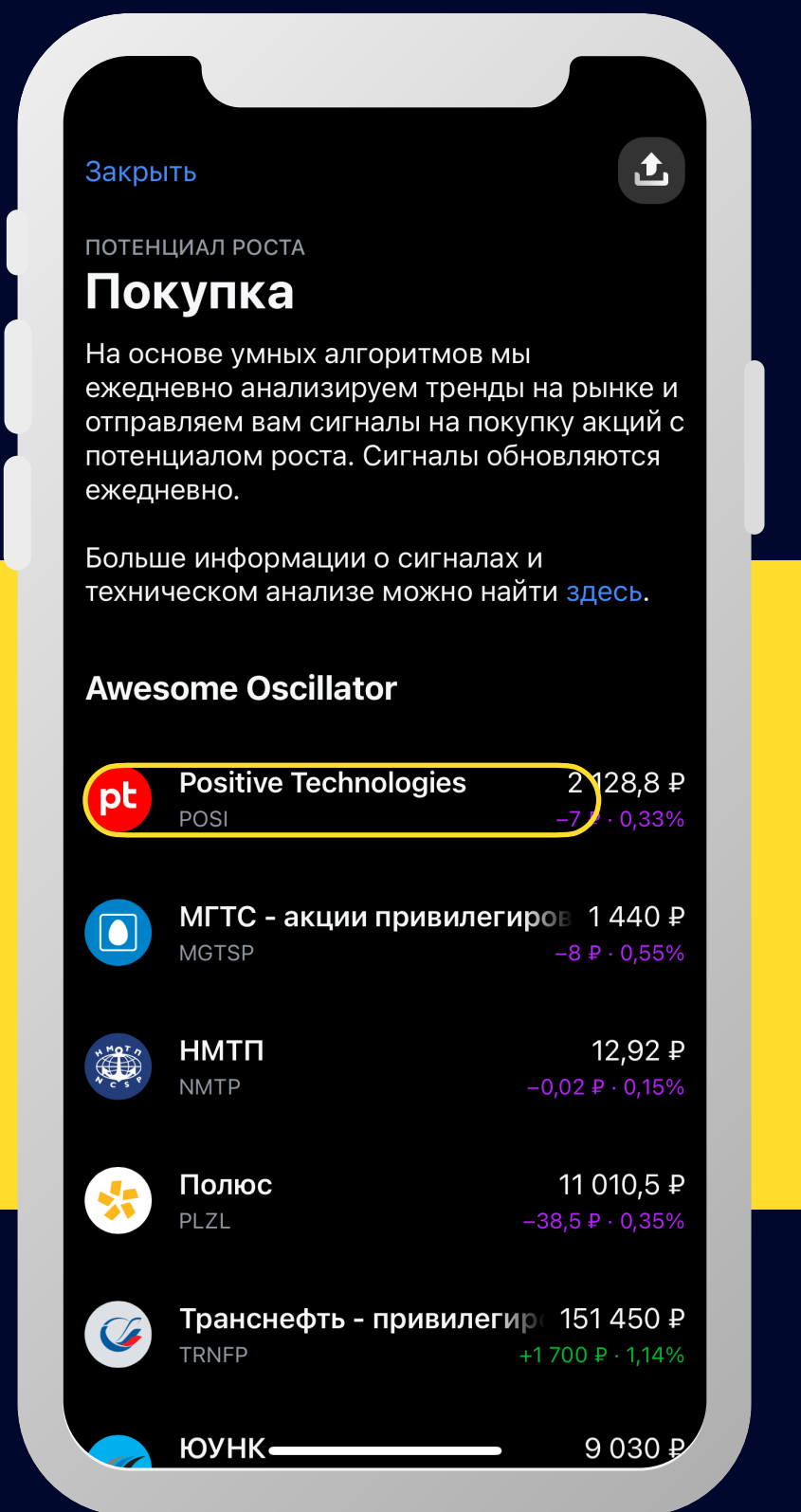


ЭНДПОИНТЫ

/widgets

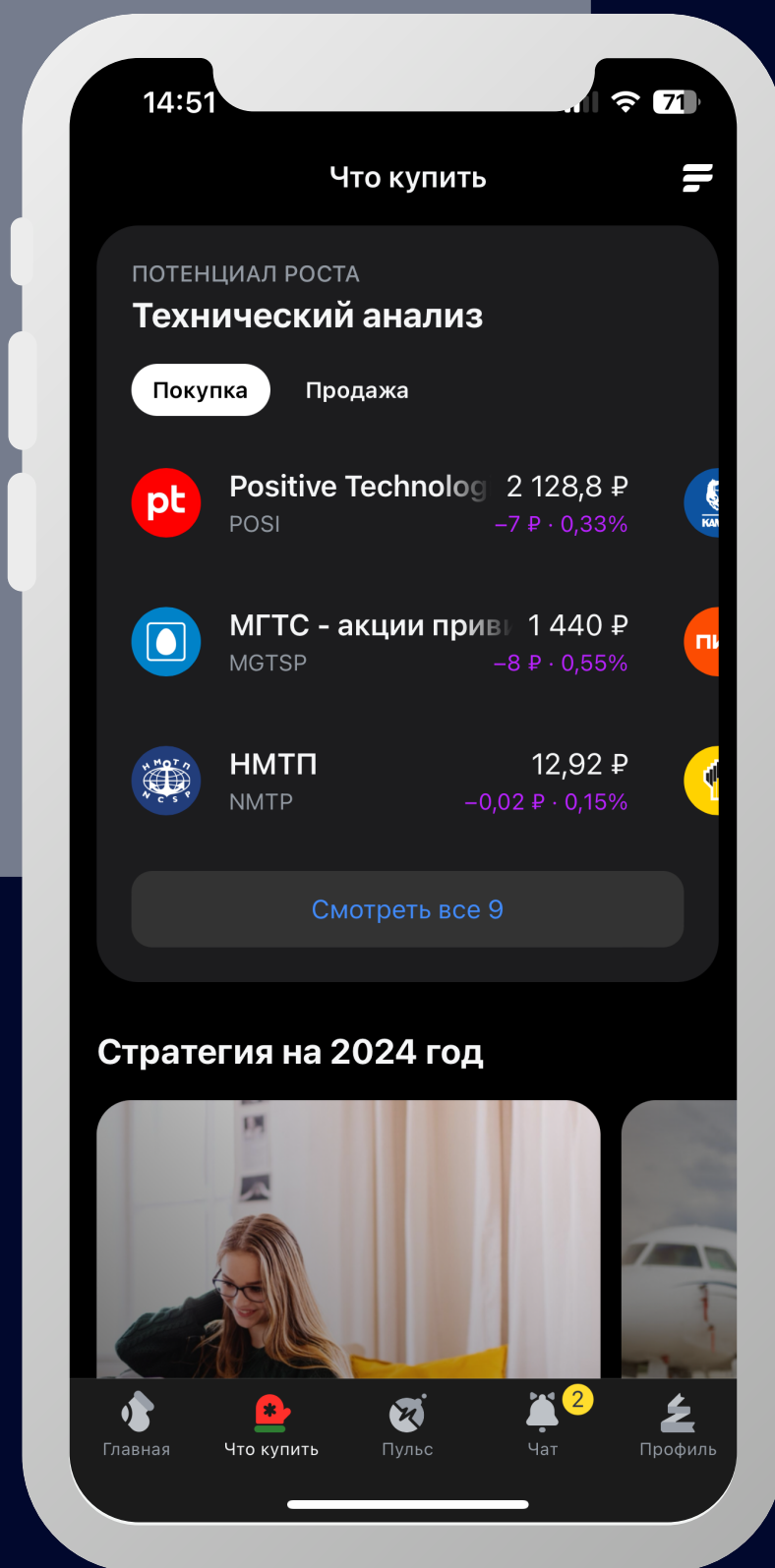


/widgets

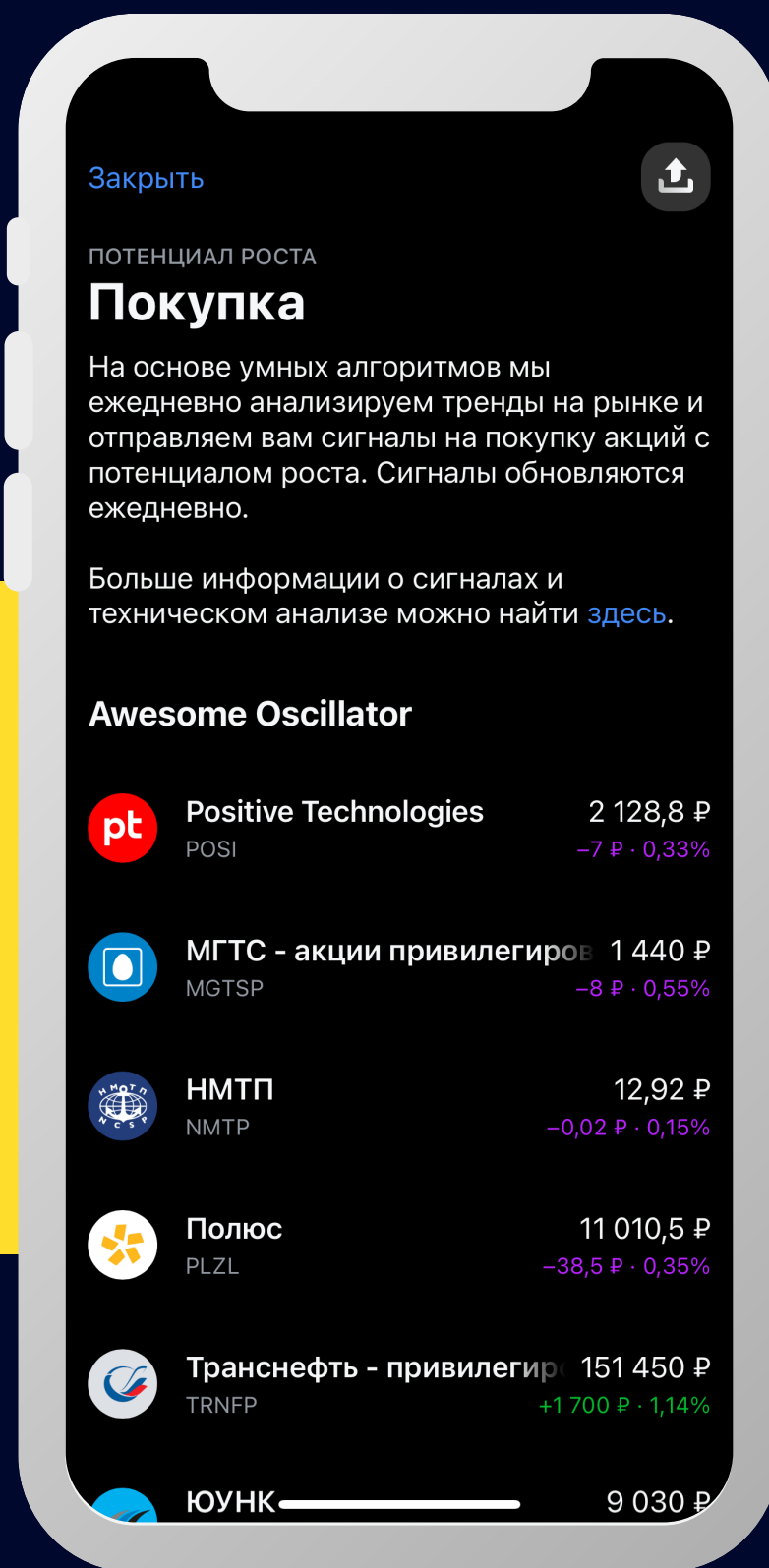


widgets/
growtpotential

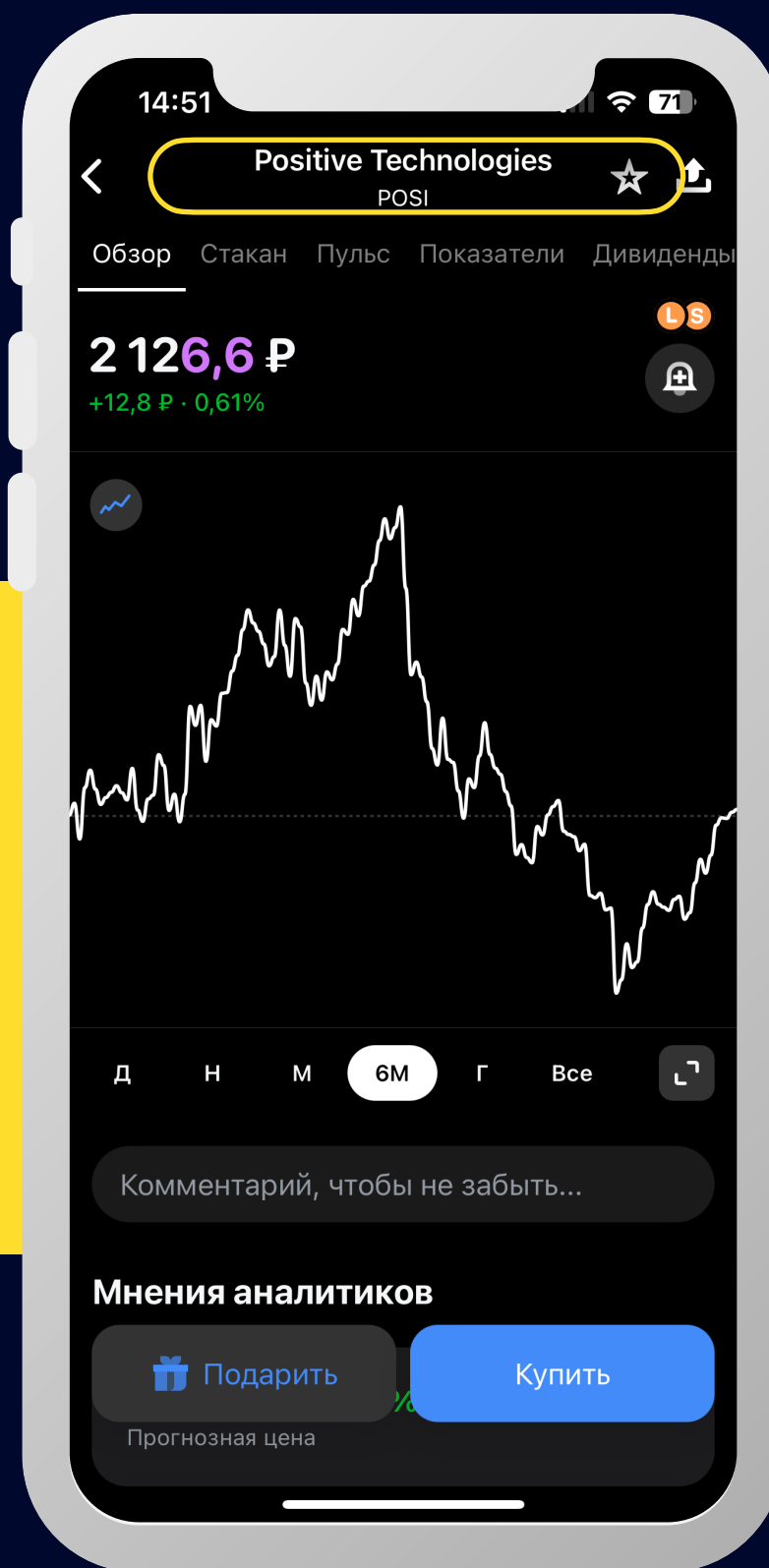
ЭНДПОИНТЫ



/widgets



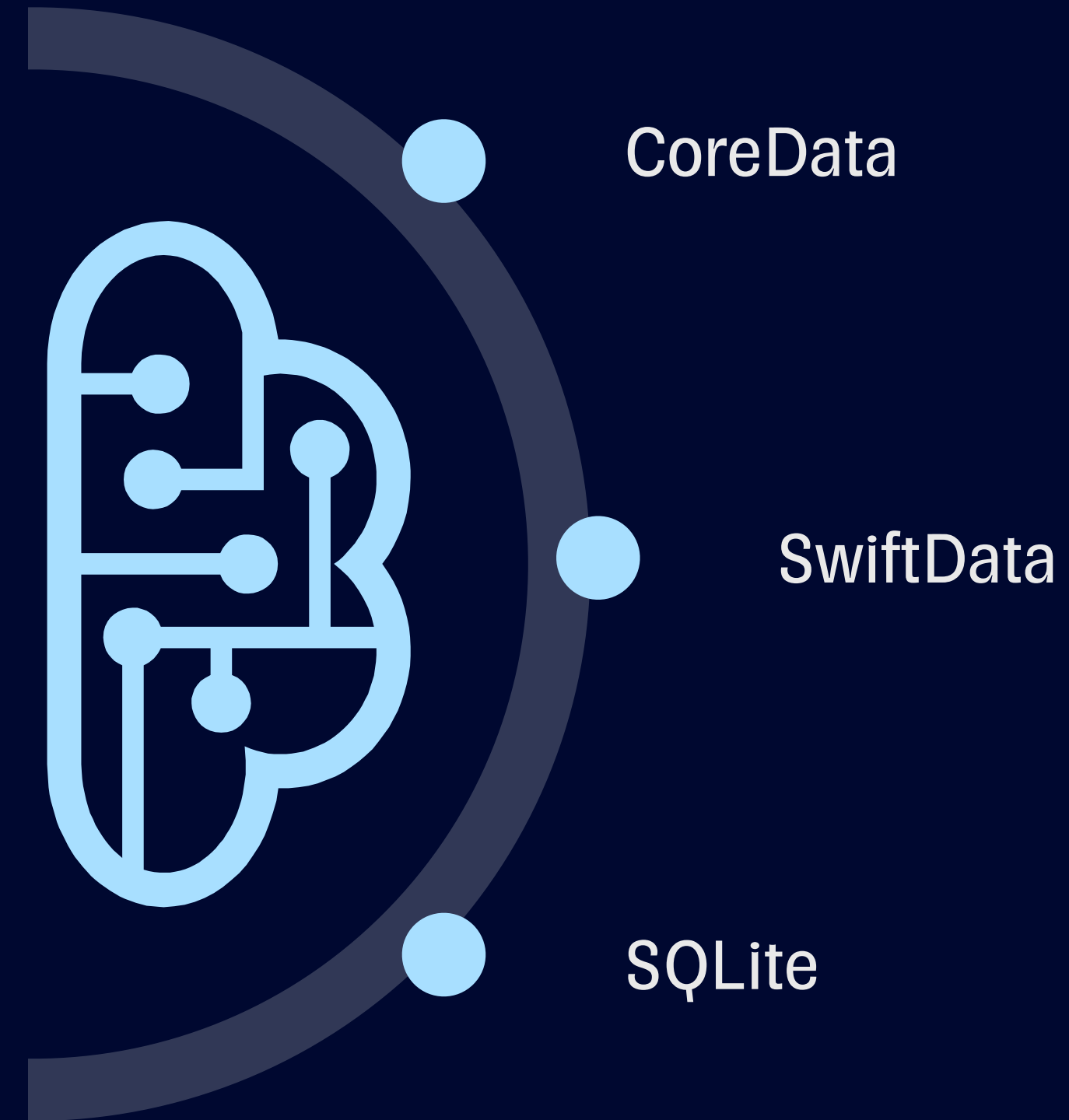
widgets/
growtpotential

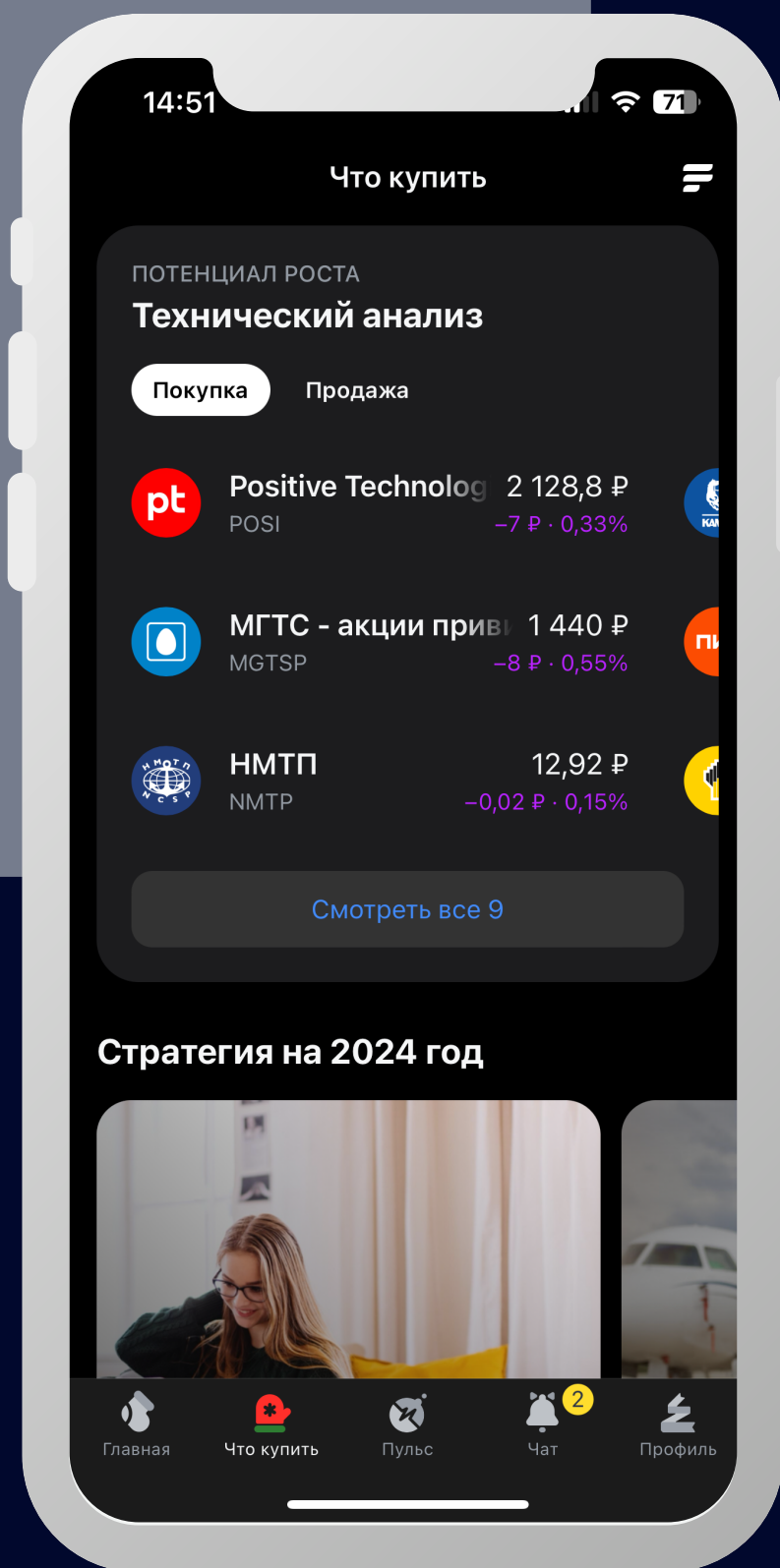


/details/POSI

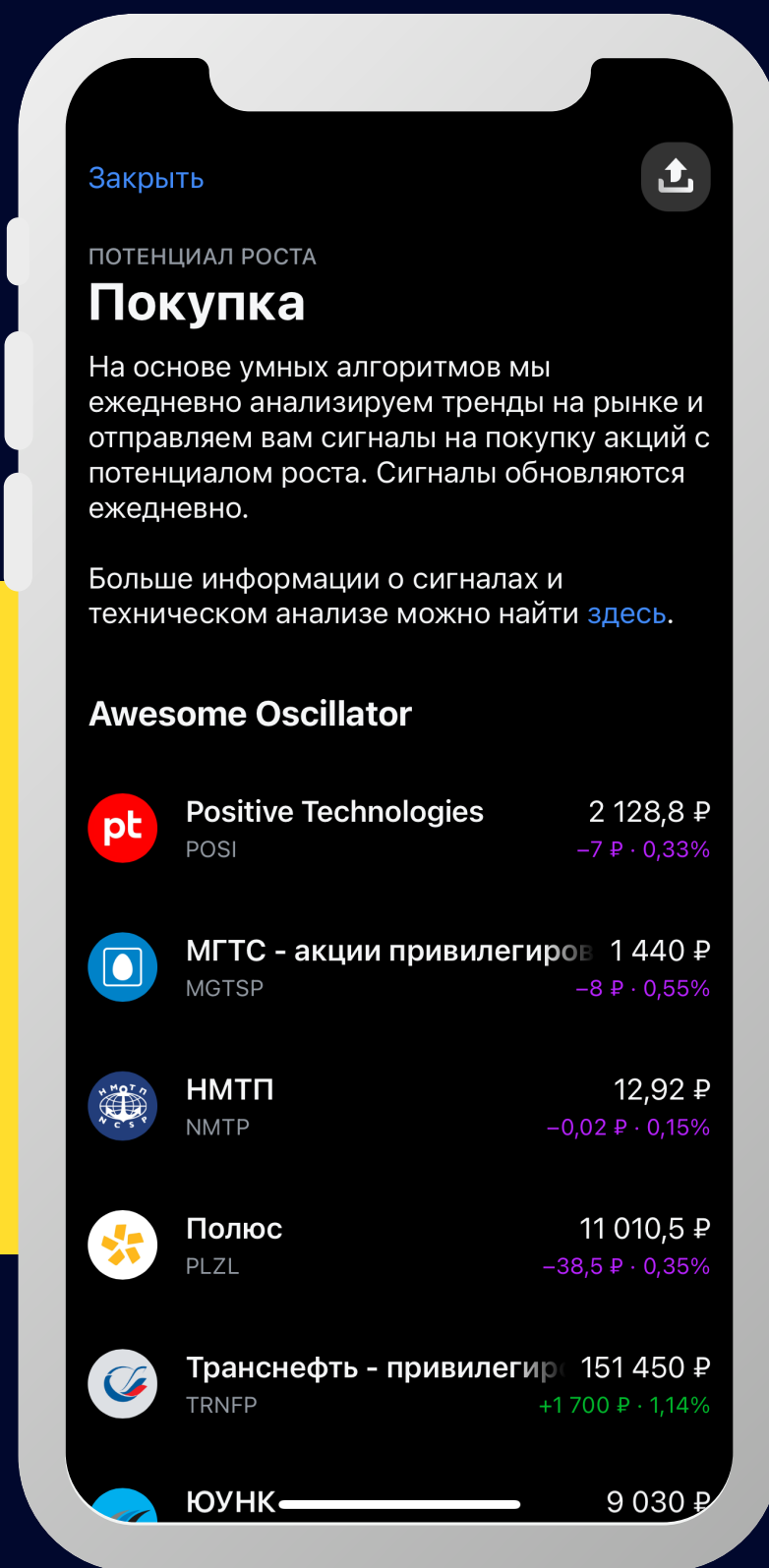
ЭНДПОИНТЫ

Персистентное хранилище

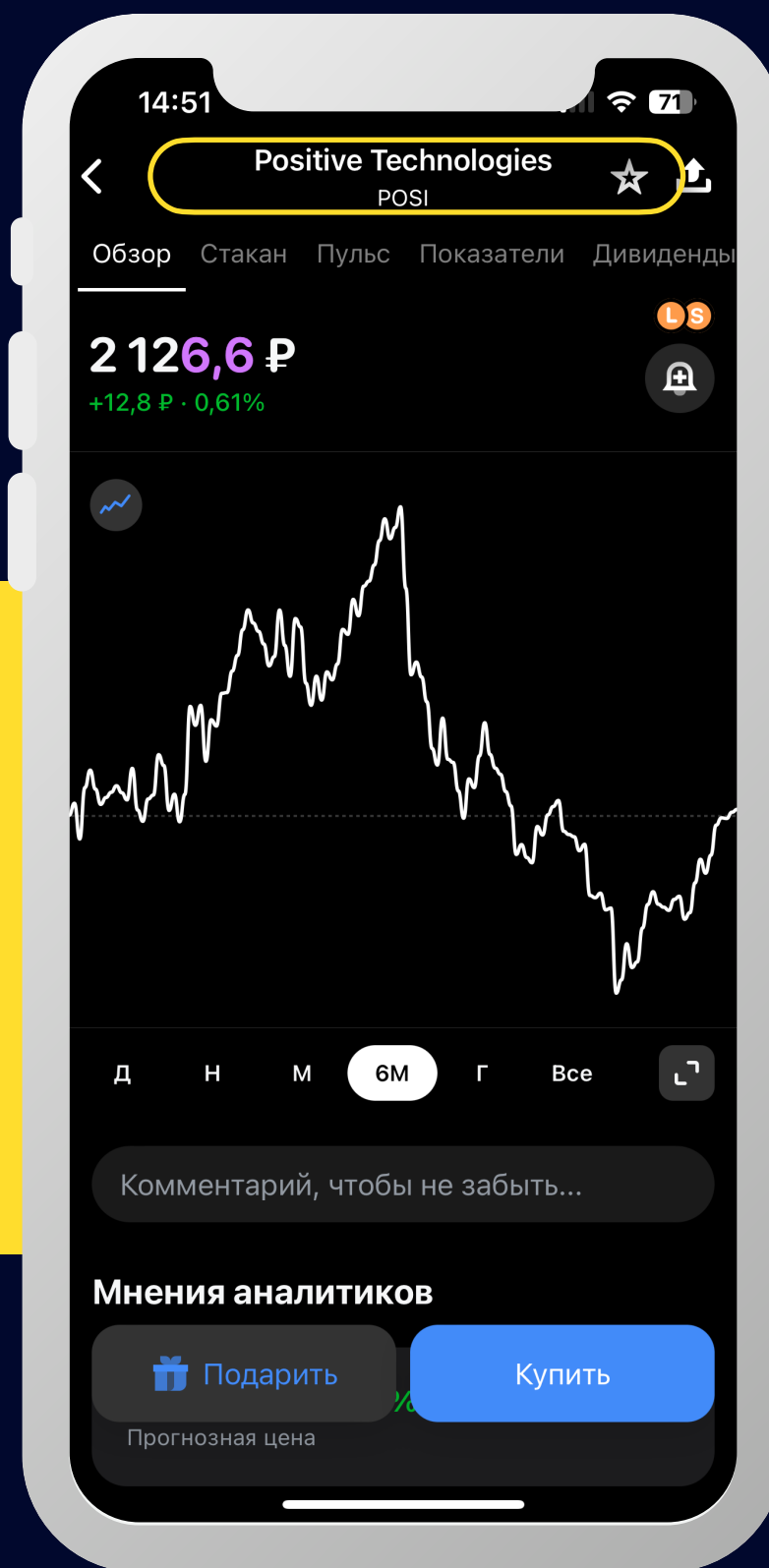




/sync



/sync



/sync

ЭНДПОИНТЫ

Уровни синхронизации

01

Многопользовательское взаимодействие

02

Синхронизация в реальном времени

03

Полный доступ к записи

04

Ограниченный доступ к записи

05

Оффлайн чтение

Обновление дельты



Обновление дельты



Оптимистичное обновление

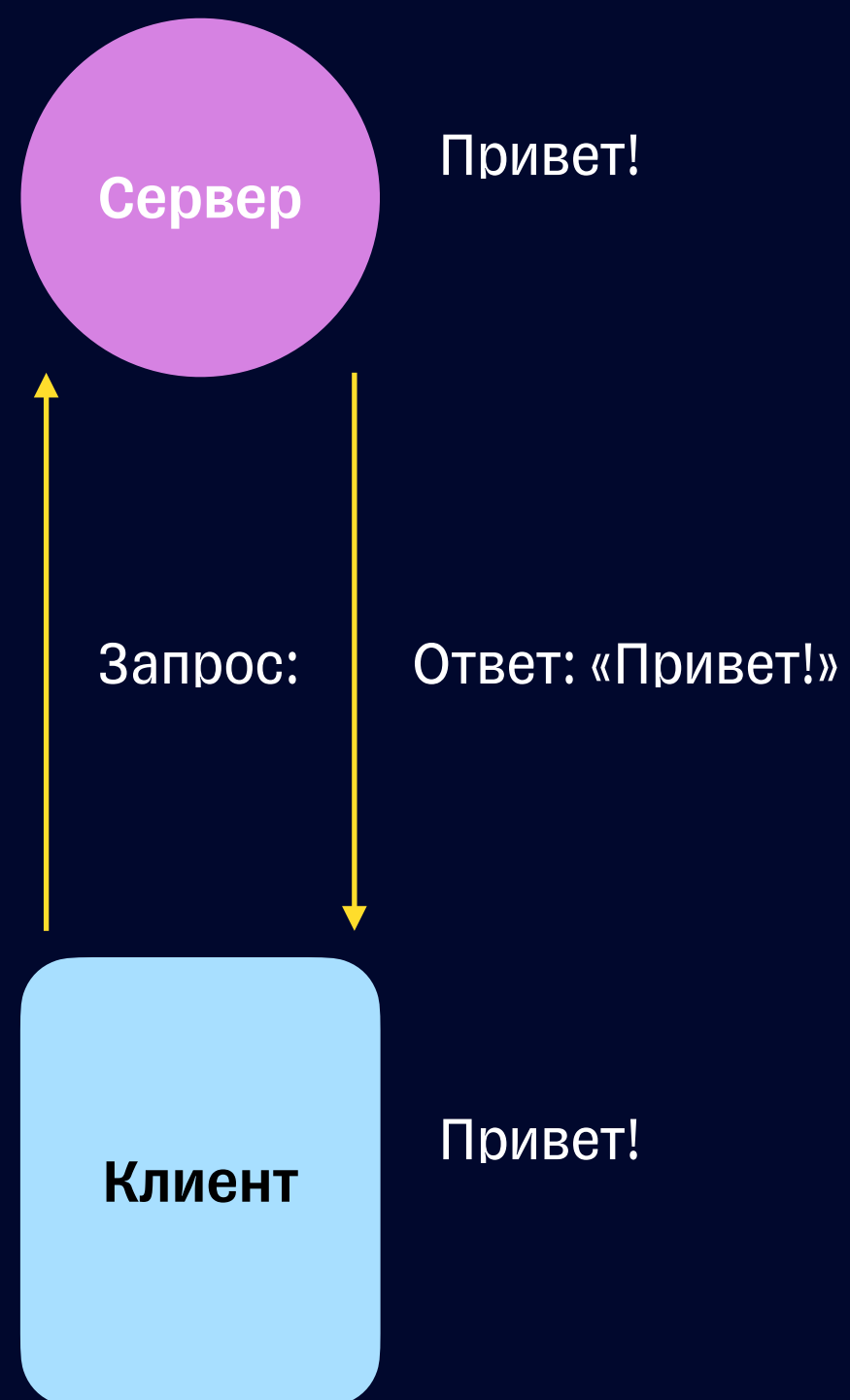


Создание контента оффлайн



Конфликты

Обновление дельты



Обновление
дельты



Оптимистичное
обновление

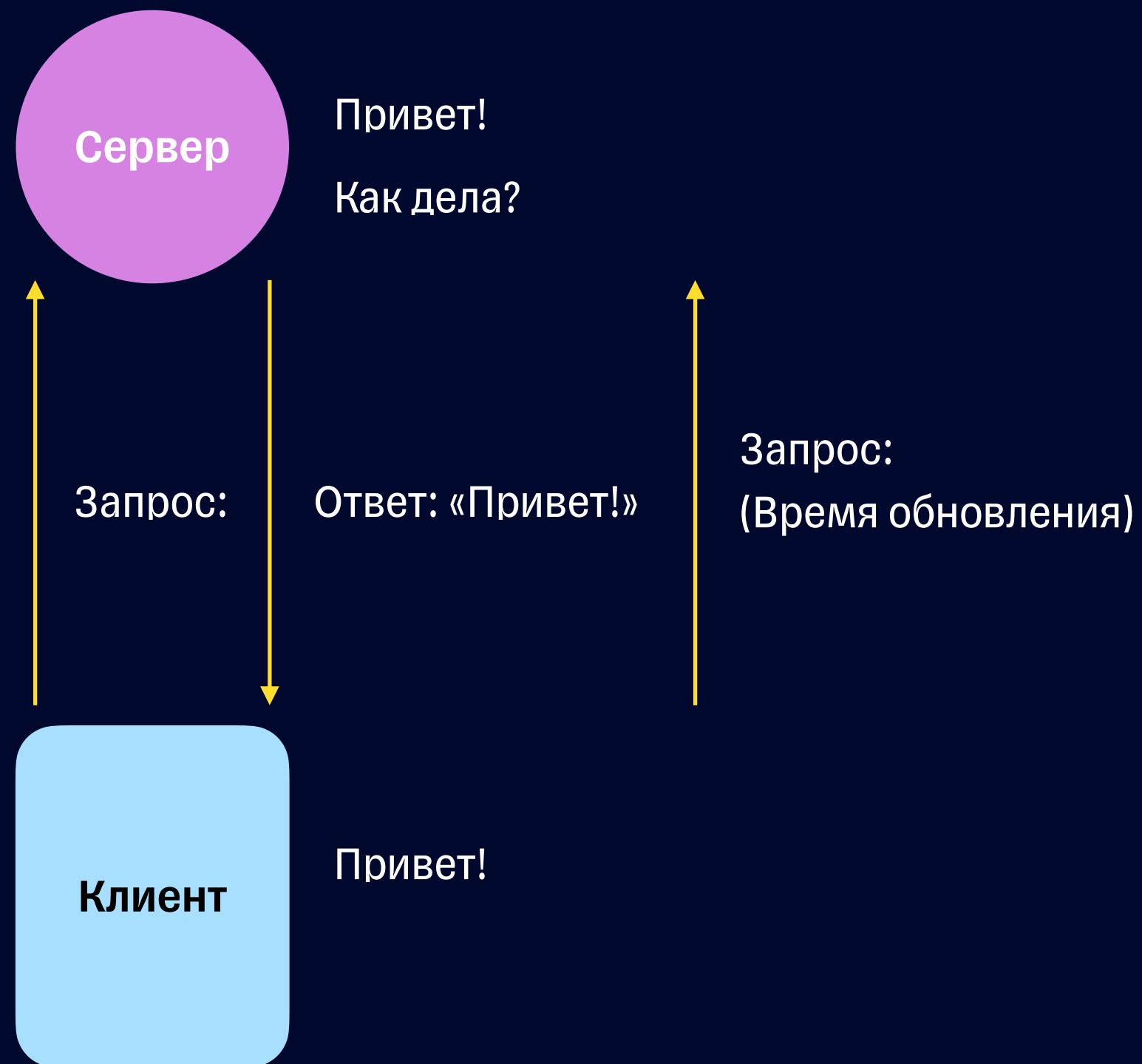


Создание контента
оффлайн



Конфликты

Обновление дельты



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Обновление дельты



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Обновление дельты

- 01** Бекенд всегда должен хранить историю изменений
- 02** Первая загрузка требует большего времени — обычно решается пагинацией
- 03** Обновление дельты всей базы практически не реализуемо и требует использования оптимизации по слайсам



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Оптимистичное обновление

- Значительное увеличение независимости наших приложений.
- Внесение изменений без ожидания подтверждения со стороны серверной части.
- Предполагается, что серверная часть одобрит изменения.
- Реализация части бизнес-логики внутри клиента.



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Создание оффлайн контента

- Создавать сущности оффлайн
- Генерировать уникальные ID
- Валидировать данные на клиенте



Обновление
дельты



Оптимистичное
обновление

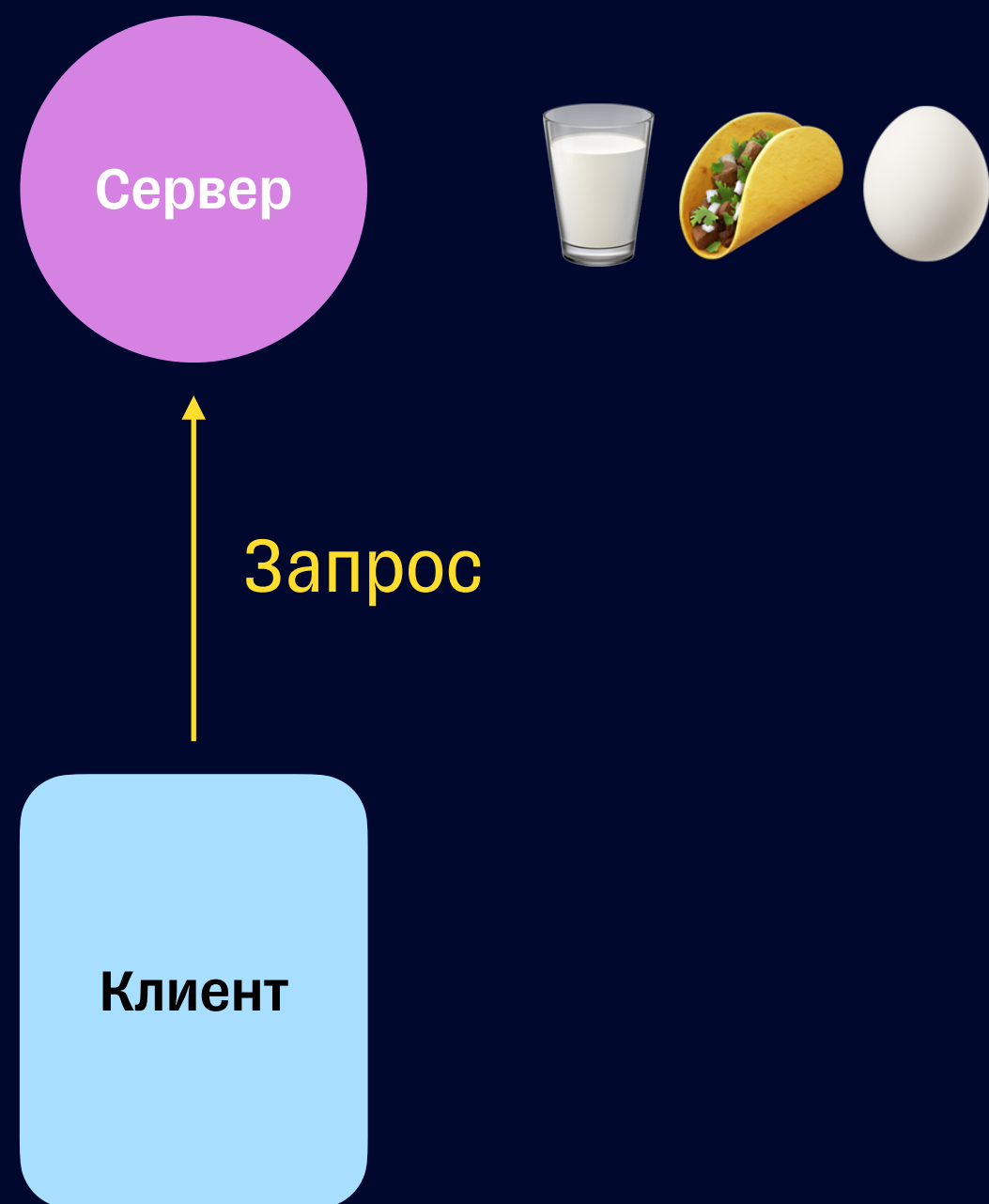


**Создание контента
оффлайн**



Конфликты

Конфликты



Обновление дельты



Оптимистичное обновление

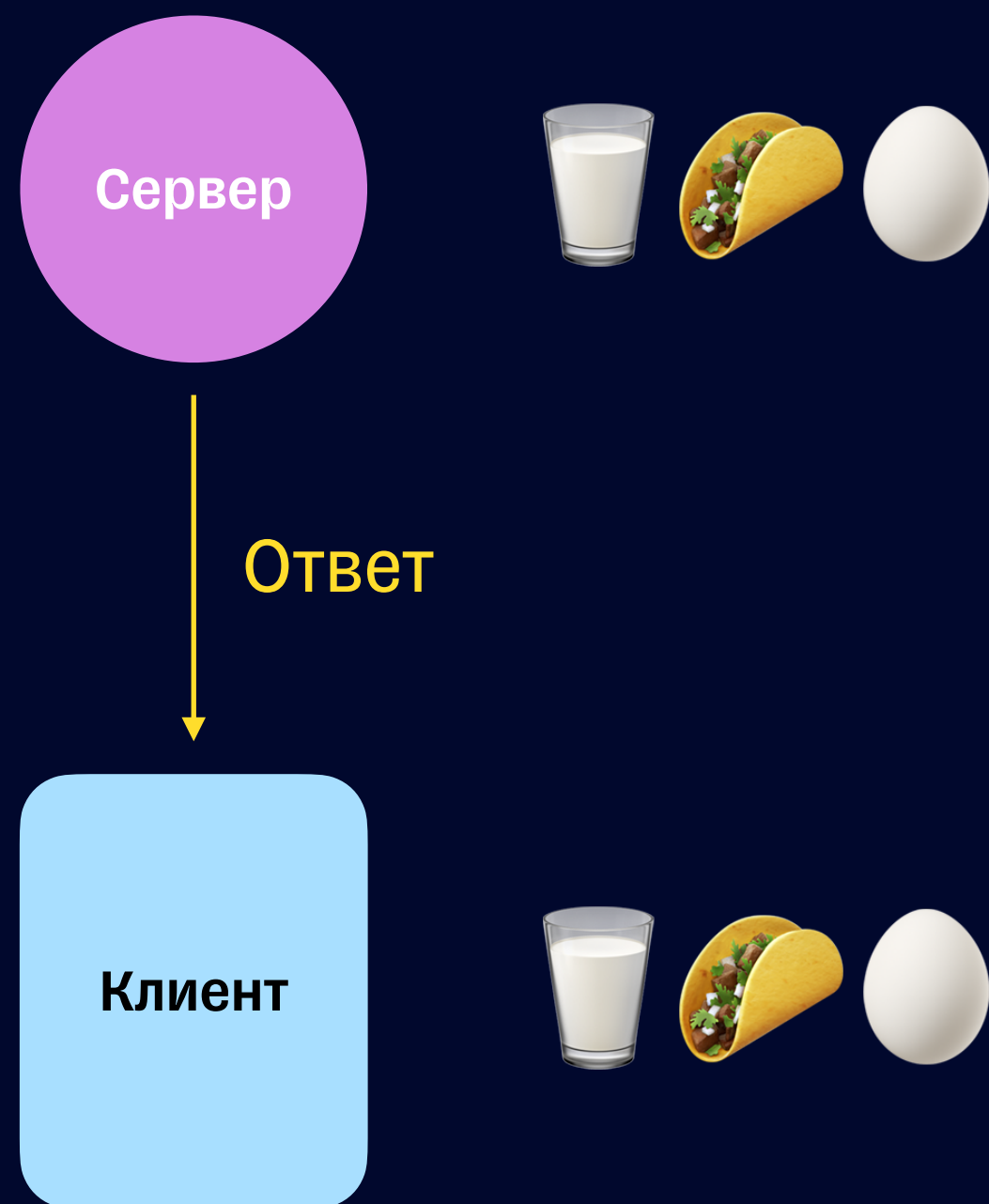


Создание контента оффлайн



Конфликты

Конфликты



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты



Сервер



Клиент



Обновление
дельты



Оптимистичное
обновление



Создание контента
оффлайн



Конфликты

Конфликты

Сервер



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

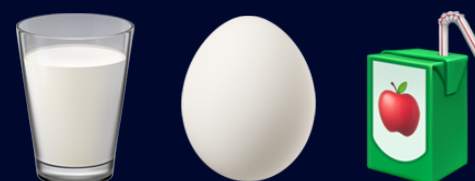
Конфликты



Сервер



Клиент



Обновление дельты



Оптимистичное обновление

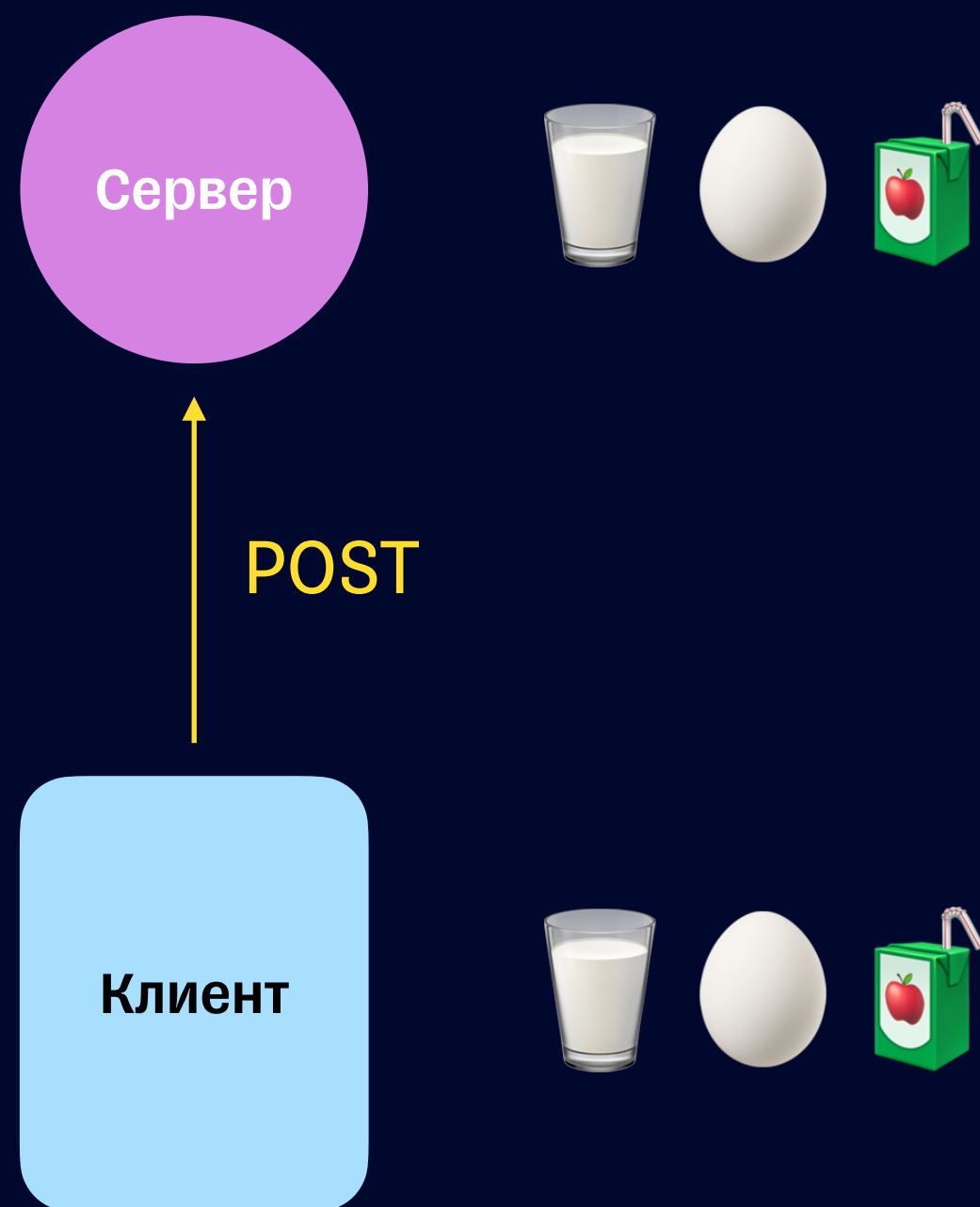


Создание контента оффлайн



Конфликты

Конфликты



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты



Сервер



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты



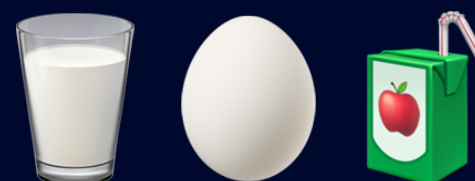
Сервер



FW =



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты

Сервер



LW =



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты

Сервер



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

Конфликты

Сервер



Клиент



Обновление дельты



Оптимистичное обновление



Создание контента оффлайн



Конфликты

CRDT

- C** Conflict-free
- R** Replicated
- D** Data
- T** Type



САР теорема

Согласованность данных (consistency), доступность (availability), устойчивость к разделению (partition tolerance)



Различные структуры CRDT

2 Счетчика, 2 Регистра, 5 Сетов, 3 графа, Словарь



Разные задачи решаются
разными структурами

CRDT сообщения

Сервер



Клиент



CRDT сообщения

Сервер



Клиент



CRDT сообщения

Сервер



Клиент



CRDT сообщения

Сервер



Клиент



CRDT сообщения

Сервер

  $D(\text{  })$

$D(\text{  }); I(\text{  }); D(\text{  })$

Клиент

   $D(\text{  }); I(\text{  })$

CRDT сообщения

Сервер



$D(\text{🌮}); I(\text{🥛}); D(\text{🥛})$

Клиент



CRDT

- C** Conflict-free
- R** Replicated
- D** Data
- T** Type



CAP теорема

Согласованность данных (consistency), доступность (availability), устойчивость к разделению (partition tolerance)



Различные структуры CRDT

2 Счетчика, 2 Регистра, 5 Сетов, 3 графа, Словарь

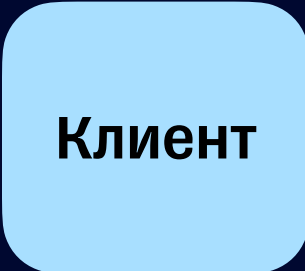


Разные задачи решаются
разными структурами

CRDT сообщения

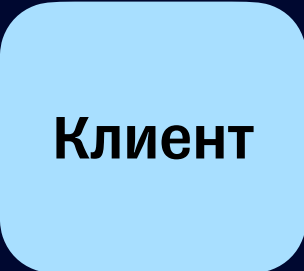


С Э Н Т В Ч
1 2 3 4 5 6



С Э Н _ В Ч
1 2 3 4 5 6

DEL(4)



С Э Н Т В И Ч
1 2 3 4 5 6 7

INS(6, И)

CRDT сообщения



DEL(4)



INS(6, И)

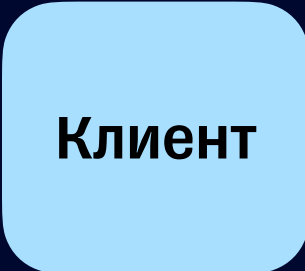
CRDT сообщения



Сервер

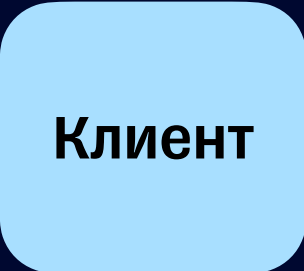
С Э Н В Ч И
1 2 3 5 6 7

DEL(4) INS(6, И)



Клиент

С Э Н _ В Ч
1 2 3 4 5 6



Клиент

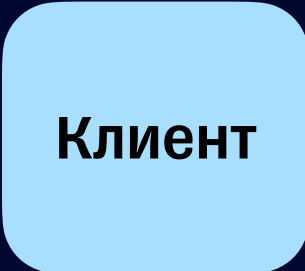
С Э Н Т В И Ч
1 2 3 4 5 6 7

CRDT сообщения



Сервер

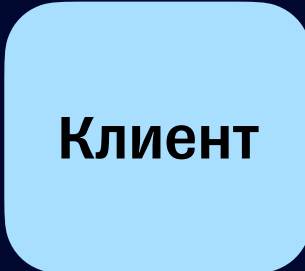
С Э Н Т В Ч
1 2 3 4 5 6



Клиент

С Э Н _ В Ч
1 2 3 4 5 6

DEL(4)



Клиент

С Э Н Т В И Ч
1 2 3 4 5 6 7

INS(6, И)

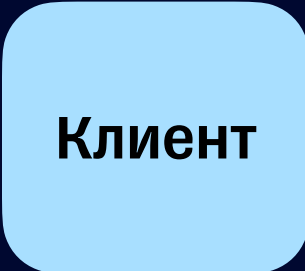
CRDT сообщения



Сервер

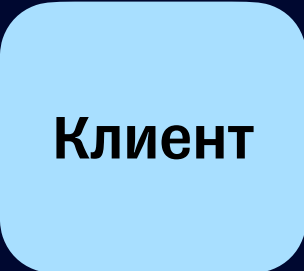
С Э Н _ В Ч
1 2 3 4 5 6

DEL(4)



Клиент

С Э Н _ В Ч
1 2 3 4 5 6



Клиент

С Э Н Т В И Ч
1 2 3 4 5 6 7

INS(6, И)

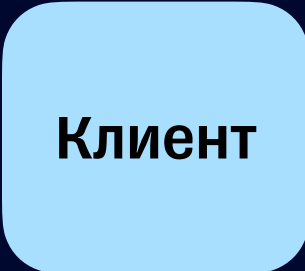
CRDT сообщения



Сервер

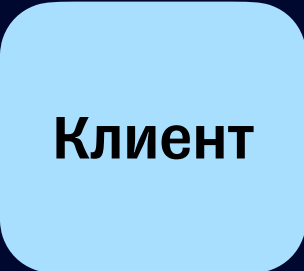
С Э Н _ В И Ч
1 2 3 4 5 6 7

DEL(4) INS(6, И)



Клиент

С Э Н _ В Ч
1 2 3 4 5 6



Клиент

С Э Н Т В И Ч
1 2 3 4 5 6 7

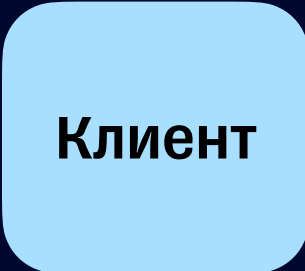
CRDT сообщения



Сервер

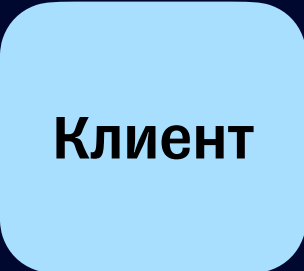
С Э Н В И Ч
1 2 3 4 5 6

DEL(4) INS(6, И)



Клиент

С Э Н _ В Ч
1 2 3 4 5 6



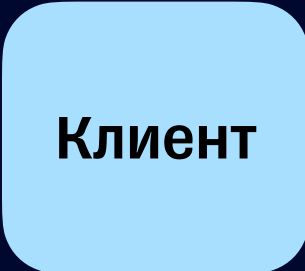
Клиент

С Э Н Т В И Ч
1 2 3 4 5 6 7

CRDT сообщения

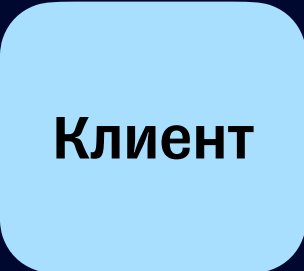


С Э Н Т В Ч
1 2 3 4 5 6



С Э Н Д В Ч
1 2 3 4 5 6

INS(4, Д)



С Э Н Т В И Ч
1 2 3 4 5 6 7

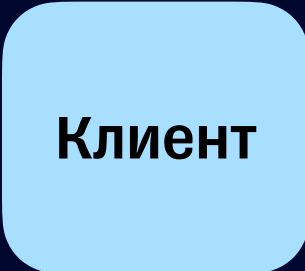
INS(6, И)

CRDT сообщения

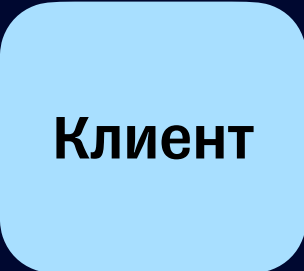


С Э Н Д В И Ч
1 2 3 4 5 6 7

INS(4, Д) INS(6, И)



С Э Н Д В Ч
1 2 3 4 5 6



С Э Н Т В И Ч
1 2 3 4 5 6 7

CRDT сообщения

The image displays a CRDT interface with two sites, Site 1 and Site 2, each showing a text editor and a weave diagram.

Site 1: #258018010

- Selected Atom: (none)
- Total Atoms: 23
- CRDT Size: 0 kb
- Status: Offline 1 2
- Buttons: All Online, All Offline, All Sites, No Sites
- Text: Birdliness
- Revision: [0:0, 1:12, 2:8] (current)
- Buttons: Show Weave, Save to File, Generate Weave, Fork Site

Weave #258018010

The weave diagram for Site 1 shows a sequence of atoms. The top row contains atoms 'e', 'W', 'e', 'r', 'l', 'd', 'l', 'y', 'X', 'l', 'n', 'e', 's'. The bottom row contains atoms 'X', 'X', 'X', 'X', 'X', 'B', 'i', 'r', 'd'. Lines connect the top row atoms to the bottom row atoms, showing the relationships between the two versions of the text.

Site 2: #113994967

- Selected Atom: (none)
- Total Atoms: 17
- CRDT Size: 0 kb
- Status: Online 1 2
- Buttons: All Online, All Offline, All Sites, No Sites
- Text: Birdly
- Revision: [0:0, 1:6, 2:8] (starting)
- Buttons: Show Weave, Save to File, Generate Weave, Fork Site

Weave #113994967

The weave diagram for Site 2 shows a sequence of atoms. The top row contains atoms 'e', 'W', 'e', 'r', 'l', 'd', 'l', 'y'. The bottom row contains atoms 'X', 'X', 'X', 'X', 'X', 'B', 'i', 'r', 'd'. Lines connect the top row atoms to the bottom row atoms, showing the relationships between the two versions of the text.

Работа без сети

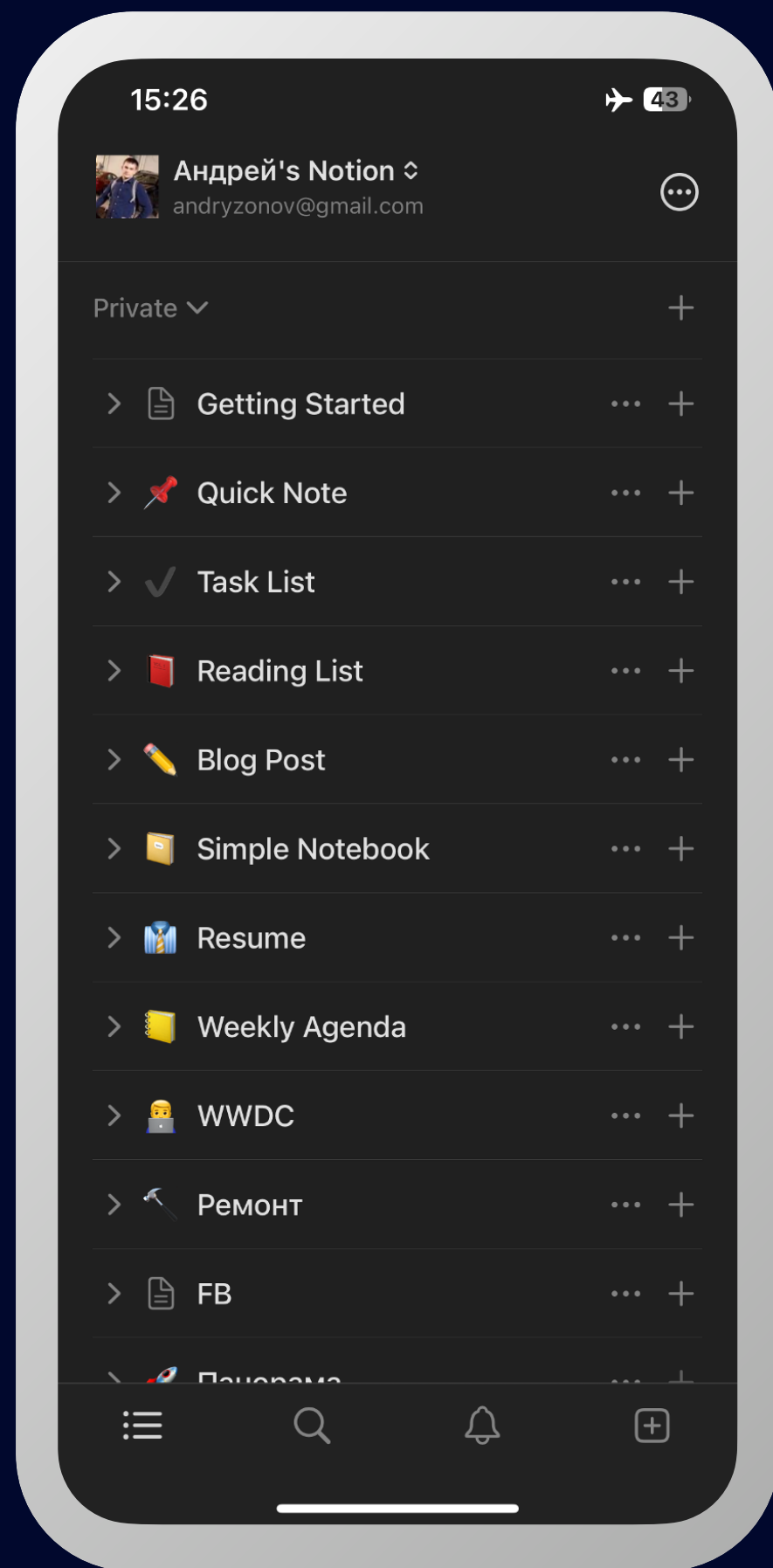


01

Это сложно

02

Это не всегда нужно



Что делать?



URL кеш



Codable + temp



Persistent in RAM



DB with sync

Как выбрать уровень синхронизации?

