Avito

МЕССЕНДЖЕР НАИЗНАНКУ

Максим Соколов, iOS @ Avito
max_sokolov

ЧТО СЕГОДНЯ УМЕЕТ ЛЮБОЙ МЕССЕНДЖЕР?

ТЕКСТОВЫЕ СООБЩЕНИЯ

ШАБЛОНЫ ОТВЕТОВ

НЕОТПРАВЛЕННЫЕ СООБЩЕНИЯ

ЧЕРНЫЙ СПИСОК

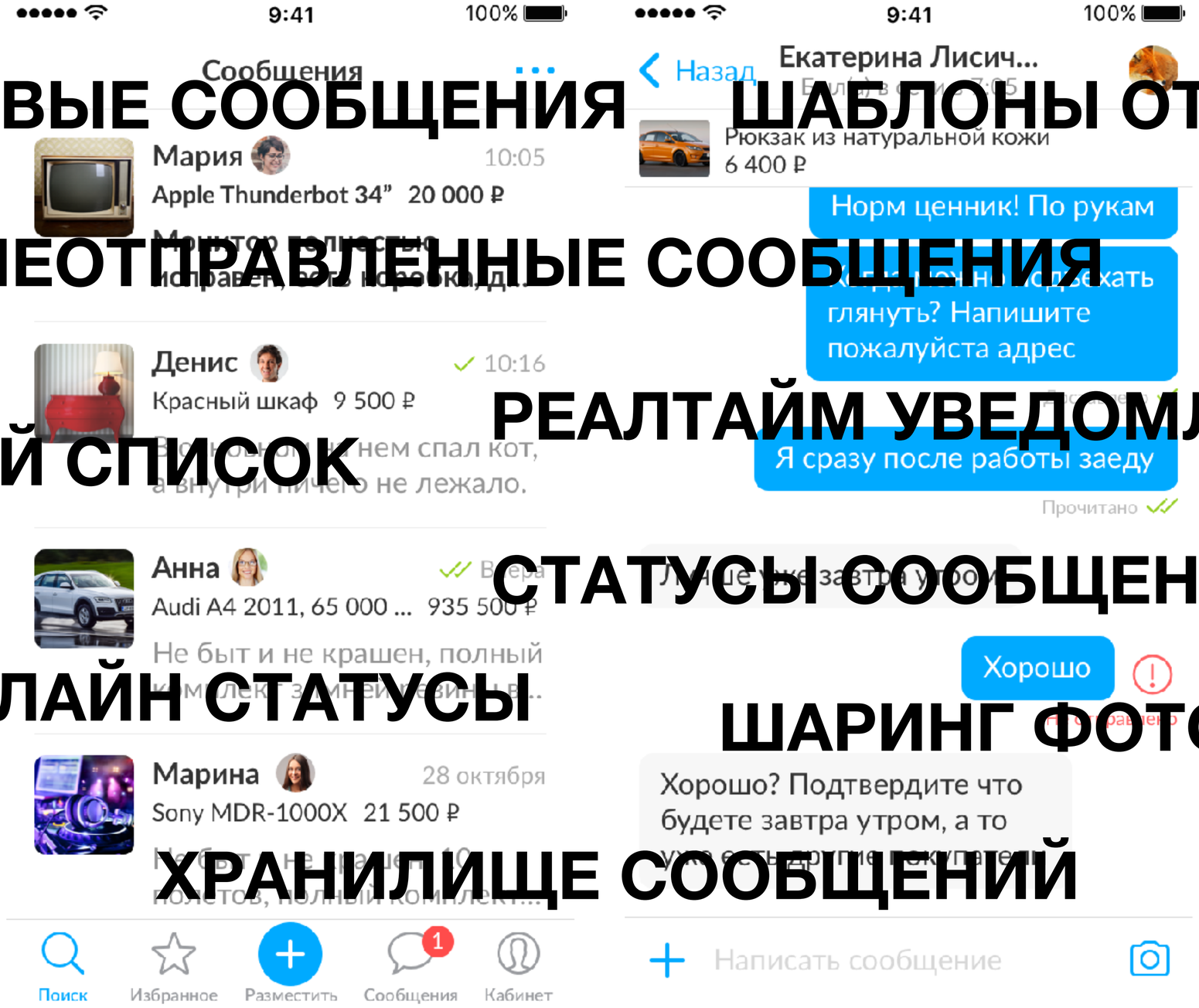
РЕАЛТАЙМ УВЕДОМЛЕНИЯ

СТАТУСЫ СООБЩЕНИЙ

ОНЛАЙН СТАТУСЫ

ШАРИНГ ФОТО

ХРАНИЛИЩЕ СООБЩЕНИЙ



Как спроектировать все так, чтобы потом не было больно?

Спойлер

Несколько приложений - один мессенджер

План

Проектирование API слоя взаимодействия с сервером

Проектирование сервисного слоя

Трюки и советы

План

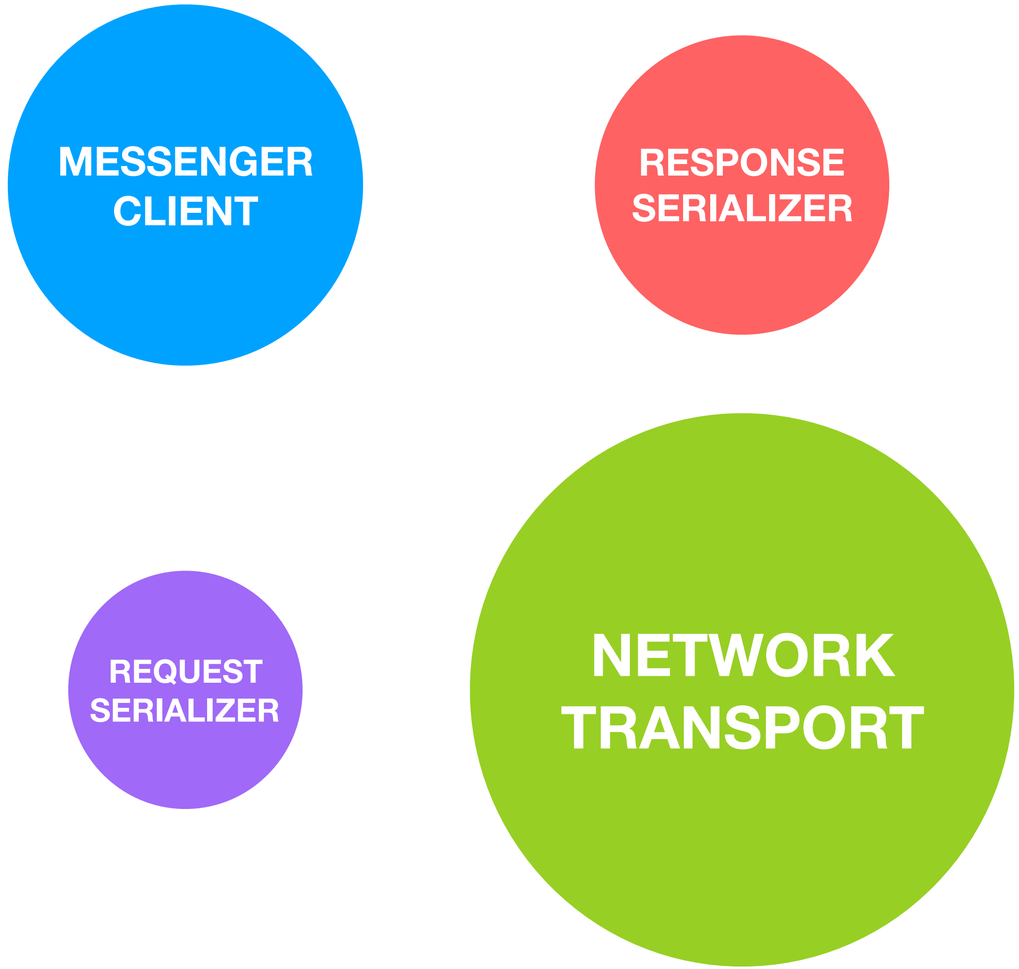
Проектирование API слоя взаимодействия с сервером

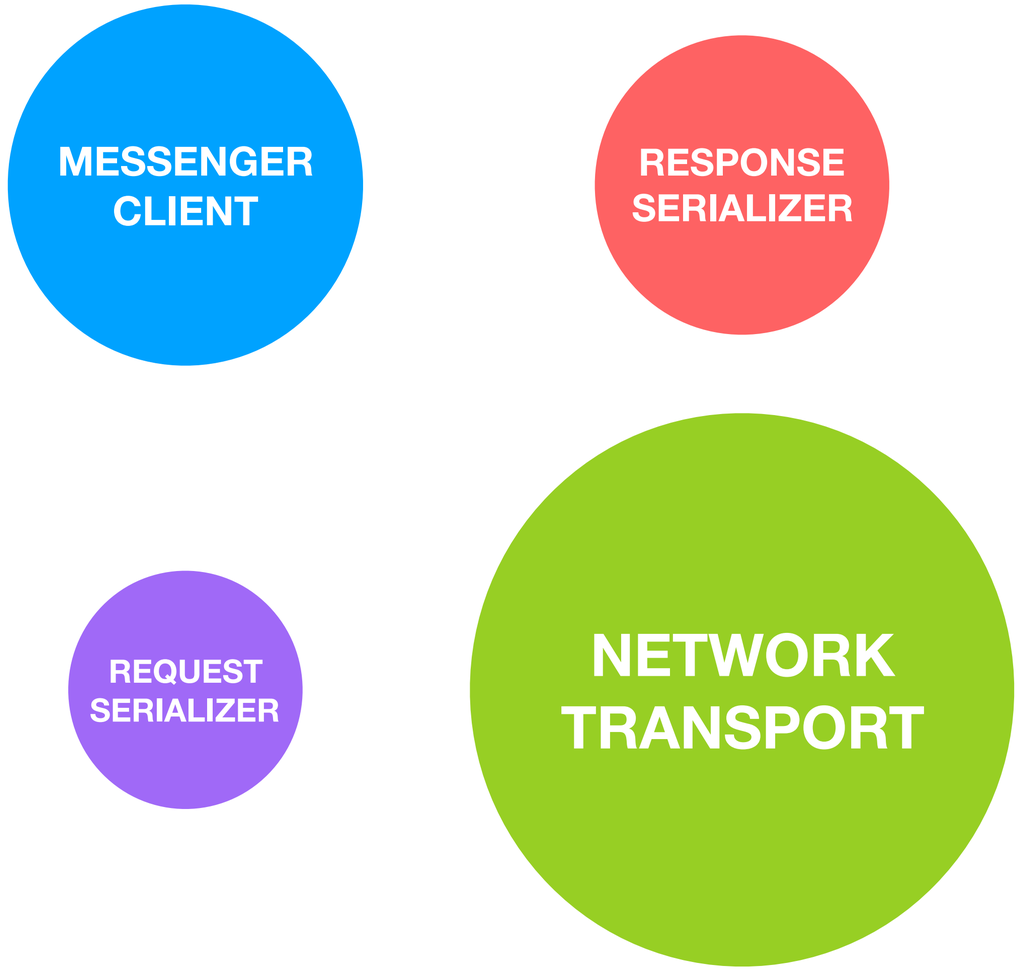
Проектирование сервисного слоя

Трюки и советы

Что такое API слой?

**Все что умеет делать Мессенджер,
но это еще не бизнес-логика.**


MESSENGER
CLIENT
RESPONSE
SERIALIZER
REQUEST
SERIALIZER
NETWORK
TRANSPORT


MESSENGER
CLIENT
RESPONSE
SERIALIZER
REQUEST
SERIALIZER
NETWORK
TRANSPORT

Сетевой транспорт

~~**TCP Socket**~~

Web Socket

HTTP (Polling / Long Polling)

?

Задача

**В приложении нужен чат,
никакого своего бекенда пока нет.**

Сетевой транспорт

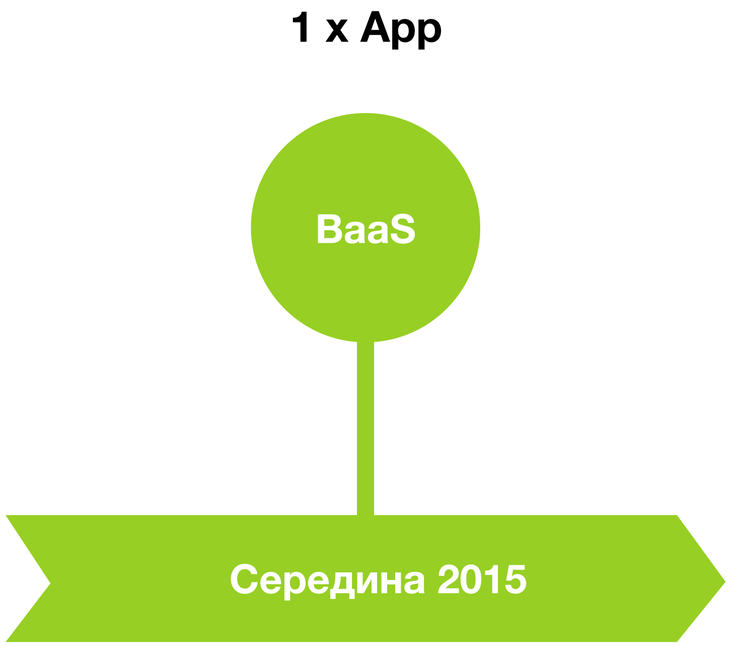
~~**TCP Socket**~~

Web Socket

HTTP (Polling / Long Polling)

BaaS (Backend as a Service)

Эволюция Мессенджера



PubNub®

“APIs for developers building secure realtime Mobile, Web, and IoT Applications.”

Backend as a Service

Отправить сообщение

Подписаться на новые сообщения

Получить историю переписки

PROFIT!

Backend as a Service

Минусы

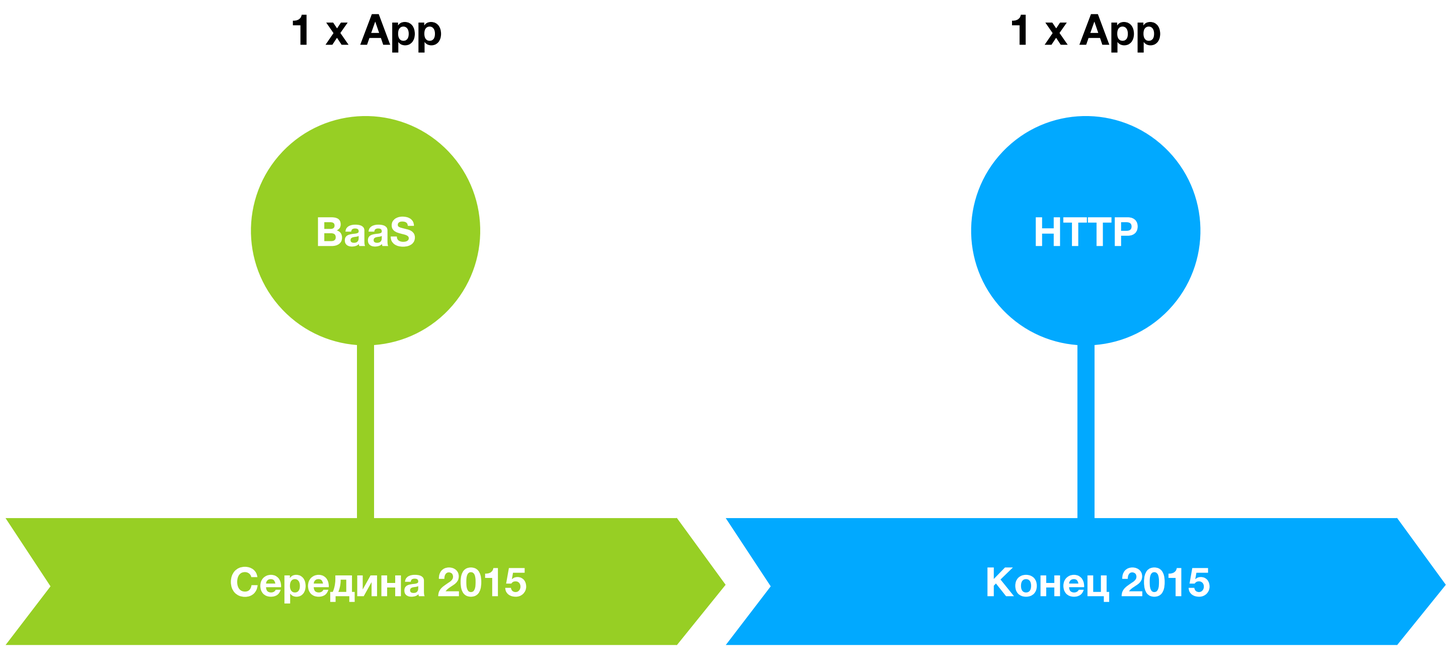
Не все сценарии можно реализовать
(статусы доставки / прочтения, черный список)*

Нет контроля над перепиской
(пре-обработка / спам)*

Забавный факт

Мы использовали parse.com и он закрылся :(
На BaaS надейся…

Эволюция Мессенджера



Backend в разработке

HTTP

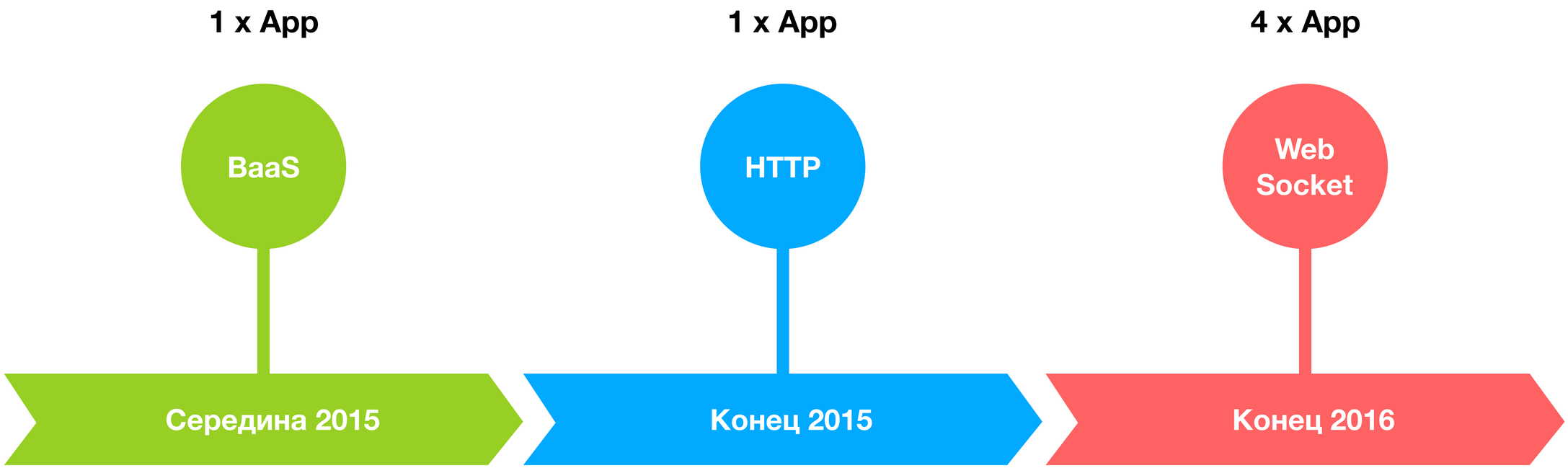
Минусы

Большие задержки
(Polling / Long polling - стратегия обновлений)

Лишний трафик, накладно делать некоторые сценарии
(http заголовки + хендшейки на соединение)

Лишние запросы
(новых событий может не быть)

Эволюция Мессенджера



Backend в разработке

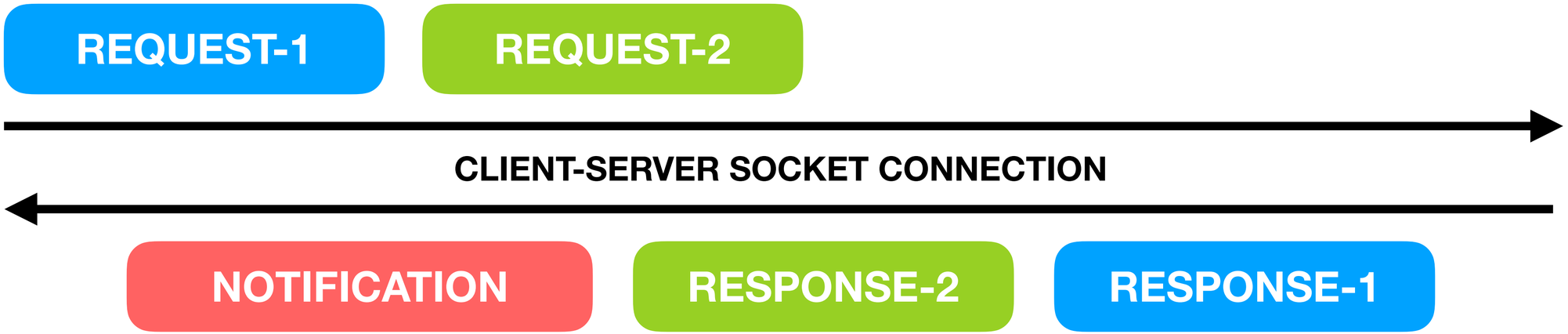
Забавный факт

Пришел веб-сокет, но приложение уже написано на BaaS - мы не хотим переписывать…

Web Socket

Нюансы работы с веб-сокетами

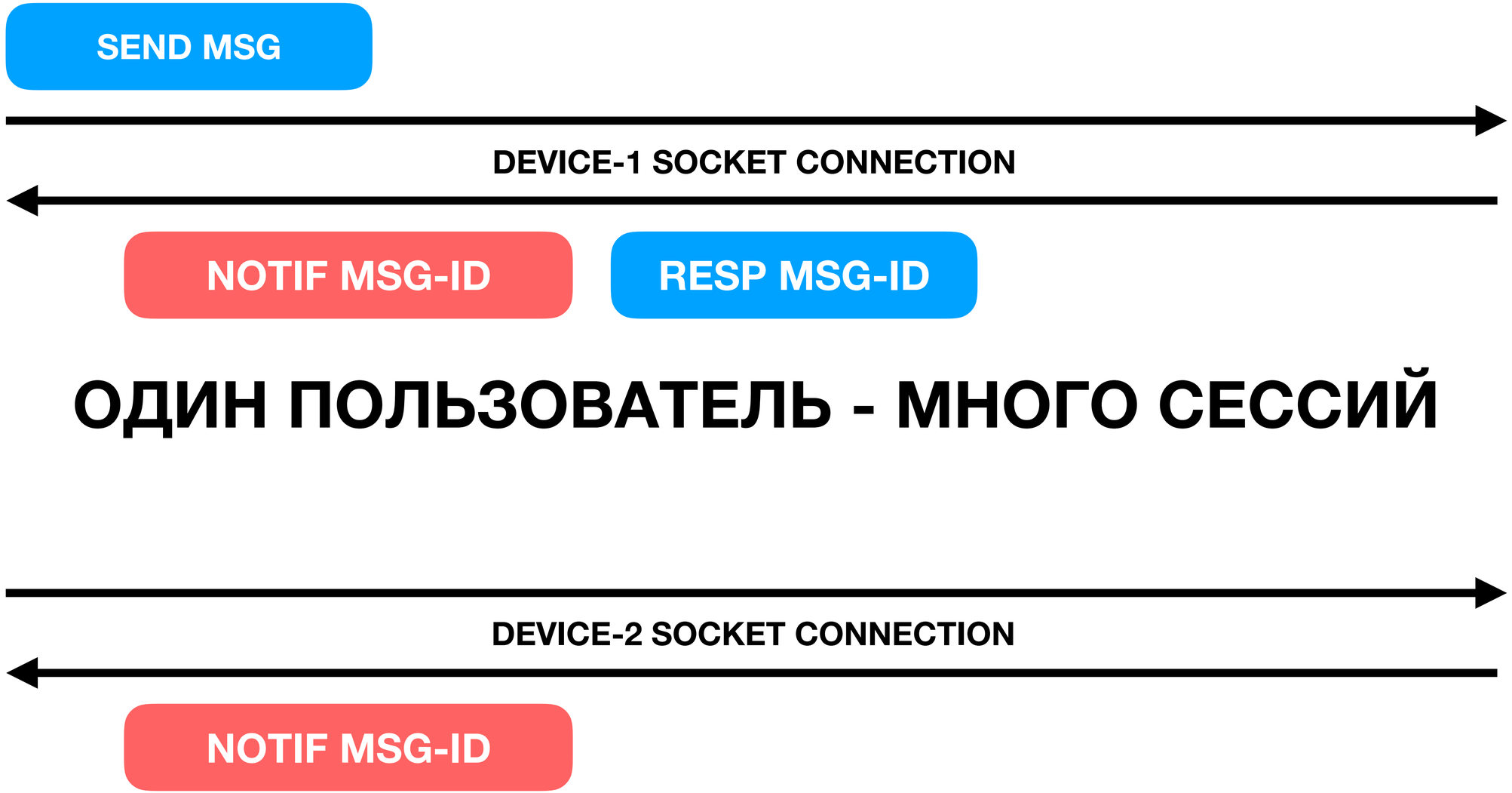
Протокол взаимодействия клиента и сервера


REQUEST-1
REQUEST-2
CLIENT-SERVER SOCKET CONNECTION
NOTIFICATION
RESPONSE-2
RESPONSE-1

JSON-RPC

```
--> {"method": "sum", "params": [41, 1], "id": 1}
<-- {"result": 42, "id": 1}
```

Дублирование сообщений в UI


SEND MSG
DEVICE-1 SOCKET CONNECTION
NOTIF MSG-ID
RESP MSG-ID
ОДИН ПОЛЬЗОВАТЕЛЬ - МНОГО СЕССИЙ
DEVICE-2 SOCKET CONNECTION
NOTIF MSG-ID

Дублирование сообщений

Решения

Добавить в тело сообщения *random id*, назначенный на клиенте

Обрабатывать нотификации после того, как получены все ответы к запросам

Поддержка соединения

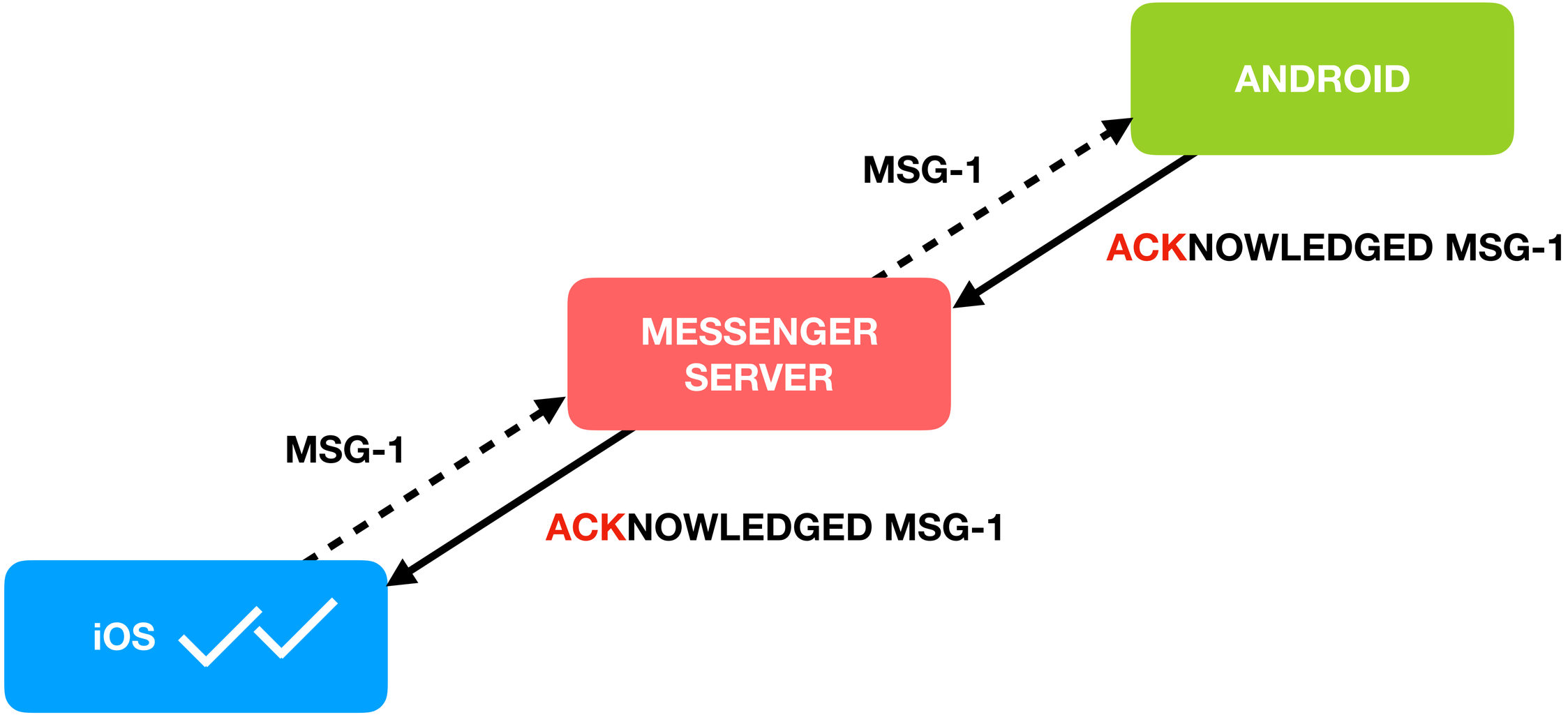
Ping / Heartbeat

Реконнекты

Background Task

Web Socket

Ack пакетов - подтверждение получения пакетов клиентом


ANDROID
MSG-1
ACKNOWLEDGED MSG-1
MESSENGER SERVER
MSG-1
ACKNOWLEDGED MSG-1
iOS

Web Socket & Polling?

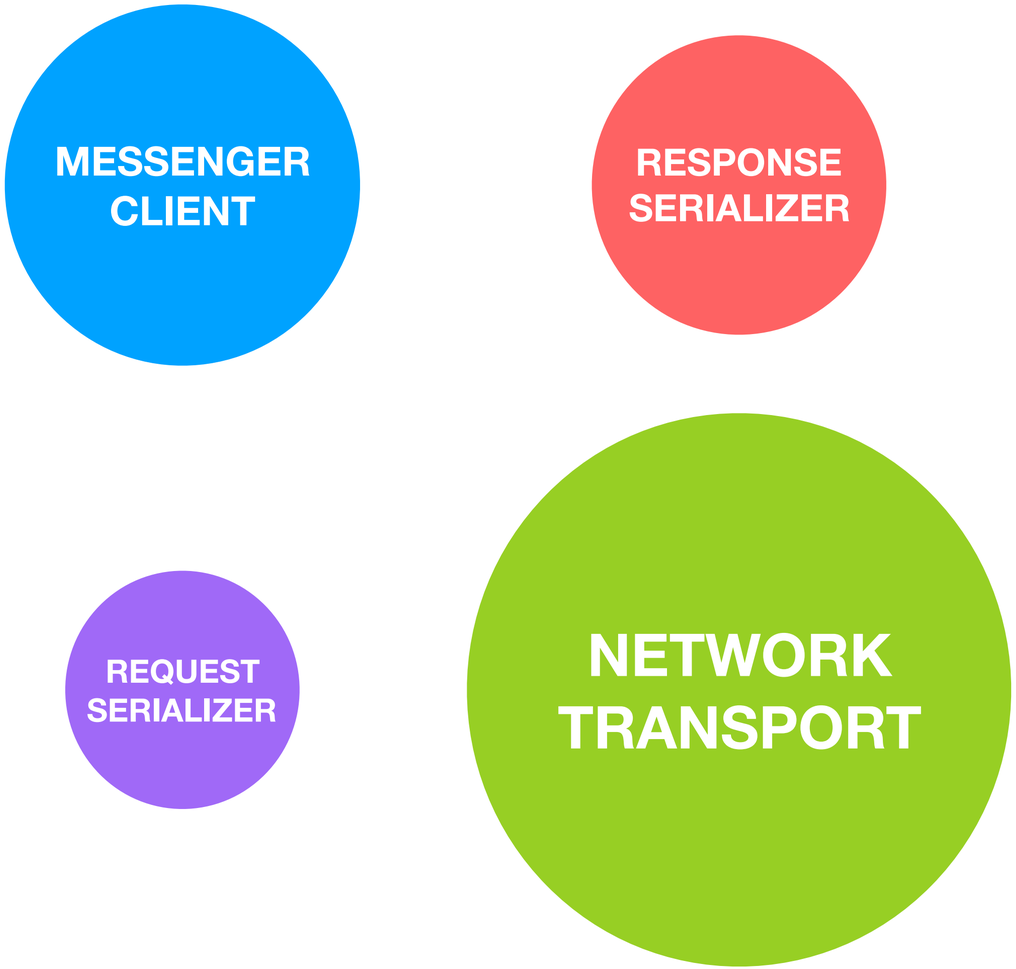
Обработка ошибок

Коды ошибок при разрыве
(401 / рефреш токена)

Очередь на запросы к серверу
(assert на превышение)

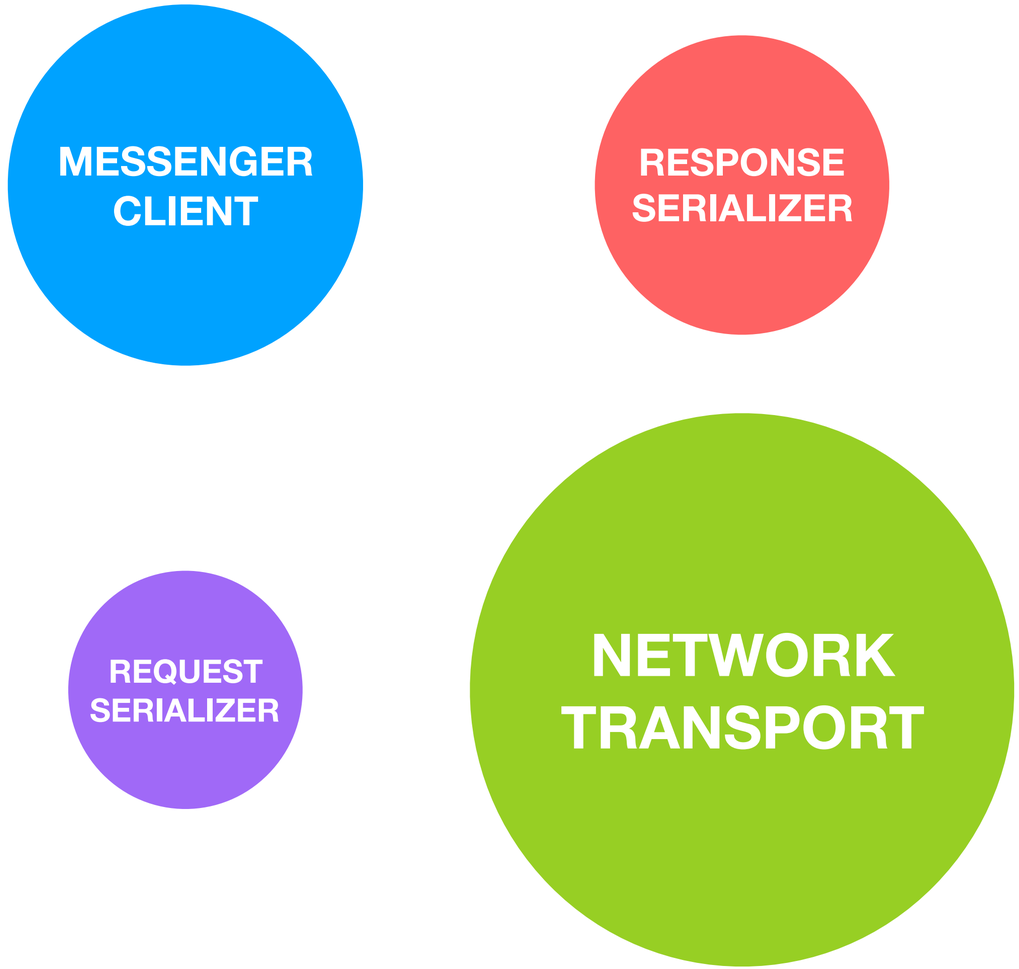
Итог

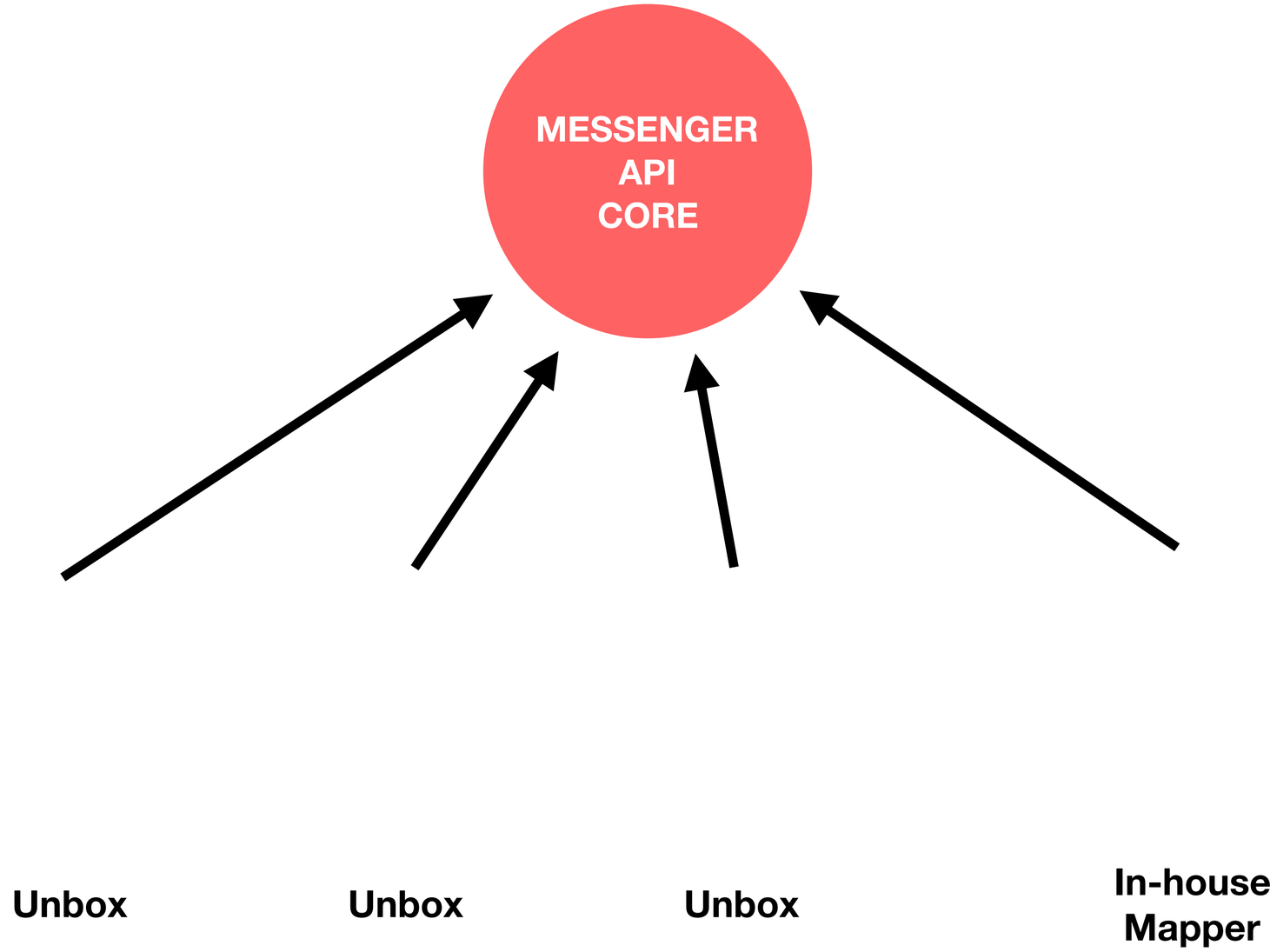
Выбирайте сетевой транспорт под задачу
(TCP Socket / Web Socket / HTTP / BaaS)


MESSENGER
CLIENT
RESPONSE
SERIALIZER
REQUEST
SERIALIZER
NETWORK
TRANSPORT

socket.isConnecting?

```
enum MessengerRequestState {

    case ready
    case delayed
}
```


MESSENGER CLIENT
RESPONSE SERIALIZER
REQUEST SERIALIZER
NETWORK TRANSPORT


MESSENGER
API
CORE
Unbox
Unbox
Unbox
In-house
Mapper

Проблема

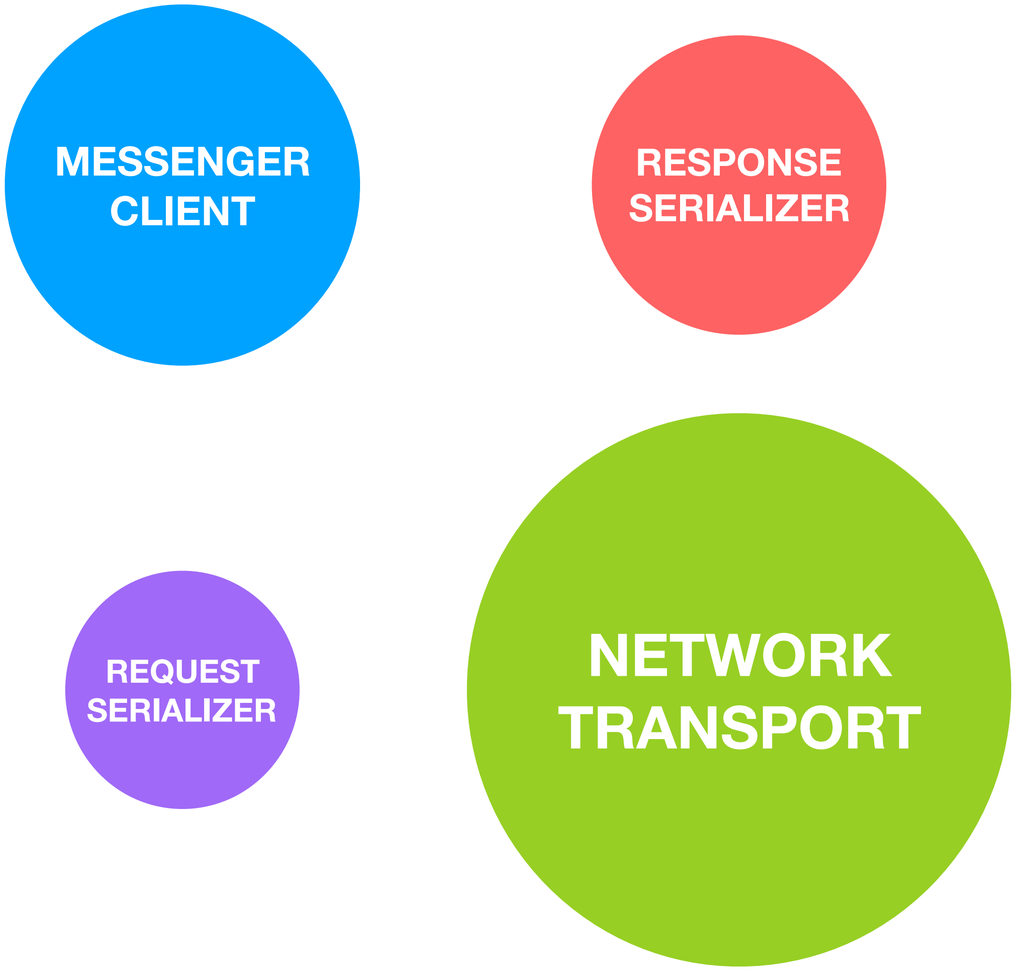
Будут ли приложения определять свой контекст моделей данных?

```swift
protocol MessengerResponseMappable: class  {

    static func map(_ dictionary: [String: String]) -> Self?
}


final class MessengerResponseMapperImpl {

    func map<T: MessengerResponseMappable>(
        responseData: Data,
        completion: @escaping (_ result: Result<T>) -> ())
    {
        let dictionary = jsonParser.parse(data: responseData)

        let value = T.map(dictionary)

        completion(.value(value))
    }
}
```

```
extension MessengerResponseMappable where Self: Unboxable {

    static func map(_ dictionary: [String: String]) -> Self? {
        return try? unbox(dictionary: dictionary)
    }
}
```

Итог

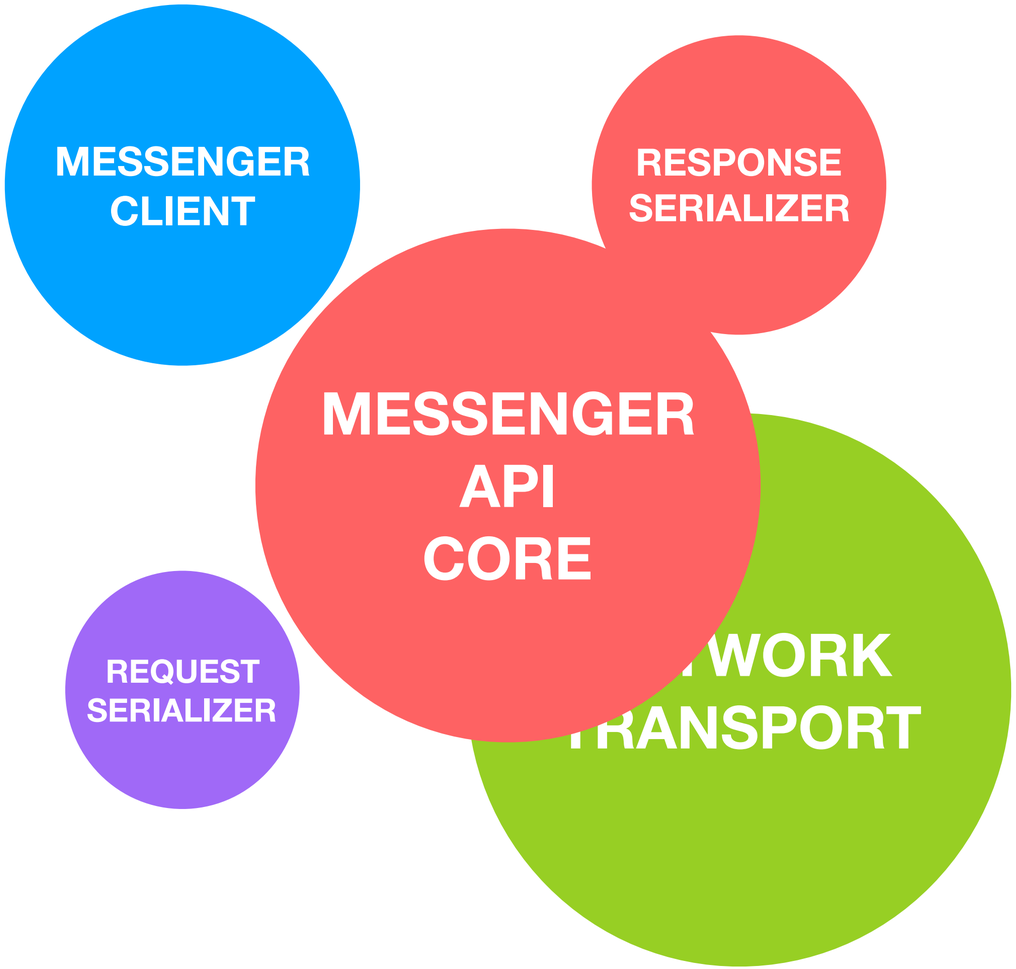
Не завязывайте маппинг ответов сетевого слоя на конкретную реализацию


MESSENGER
CLIENT
RESPONSE
SERIALIZER
REQUEST
SERIALIZER
NETWORK
TRANSPORT

```
open class MessengerClient {

    /*
     Inject:
         Network transport
         Request serializer
         Response serializer
    */

    func getChats<T: MessengerResponseMappable>(
        completion: @escaping (_ result: Result<[T]>) -> Void) {
    }
}
```

```
messengerClient.getChats(
    completion: { (result: Result<[Chat]>) in

    }
)
```

```
messengerClient.getChats(
    completion: { (result: Result<[OtherChat]>) in

    }
)
```


MESSENGER
CLIENT
RESPONSE
SERIALIZER
MESSENGER
API
CORE
REQUEST
SERIALIZER
WORK
TRANSPORT

Вывод

Выносите слой работы с API
в отдельный модуль
(мы используем Private CocoaPods)

План

Проектирование API слоя взаимодействия с сервером

Проектирование сервисного слоя

Трюки и советы

Сервисный слой

Легко тестировать

Легко расширять

Легко переиспользовать

Легко композировать сервисы друг с другом

Сервисный слой

Легко тестировать

Легко расширять

Легко переиспользовать

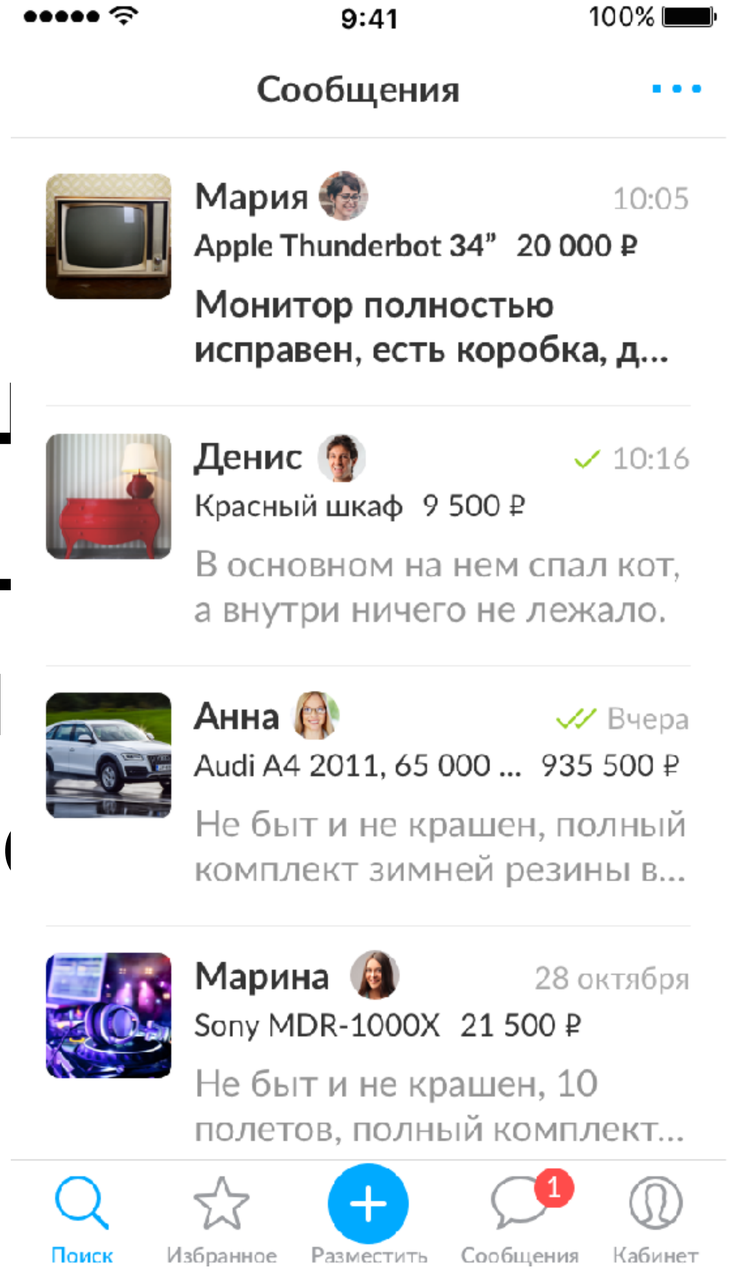
Легко композировать сервисы друг с другом

НОВОЕ СООБЩ

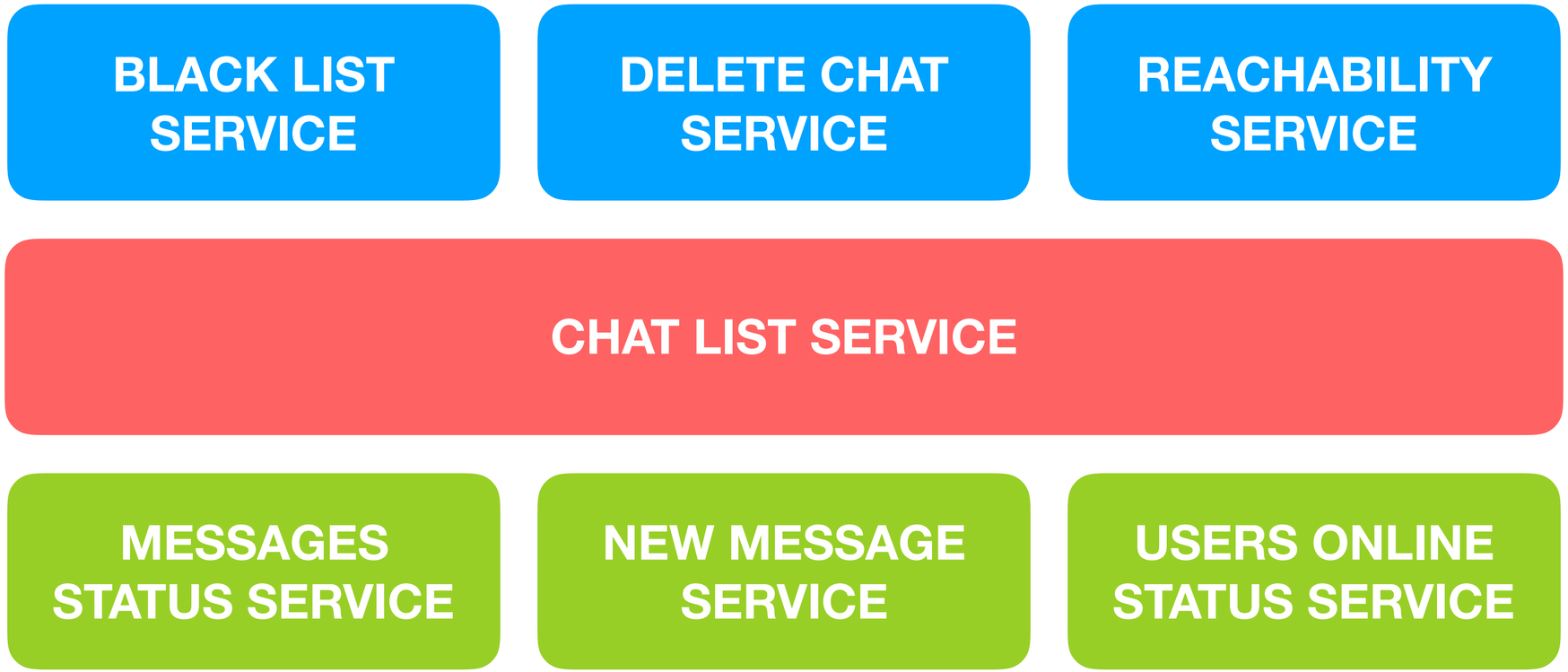
СТАТУС СООБЩ

ЮЗЕР ОНЛАЙН

ЧЕРНЫЙ СПИС



МИКРОСЕРВИСНАЯ АРХИТЕКТУРА


BLACK LIST SERVICE
DELETE CHAT SERVICE
REACHABILITY SERVICE
CHAT LIST SERVICE
MESSAGES STATUS SERVICE
NEW MESSAGE SERVICE
USERS ONLINE STATUS SERVICE

FAVOR COMPOSITION

Как связать все вместе?

OBSERVER PATTERN

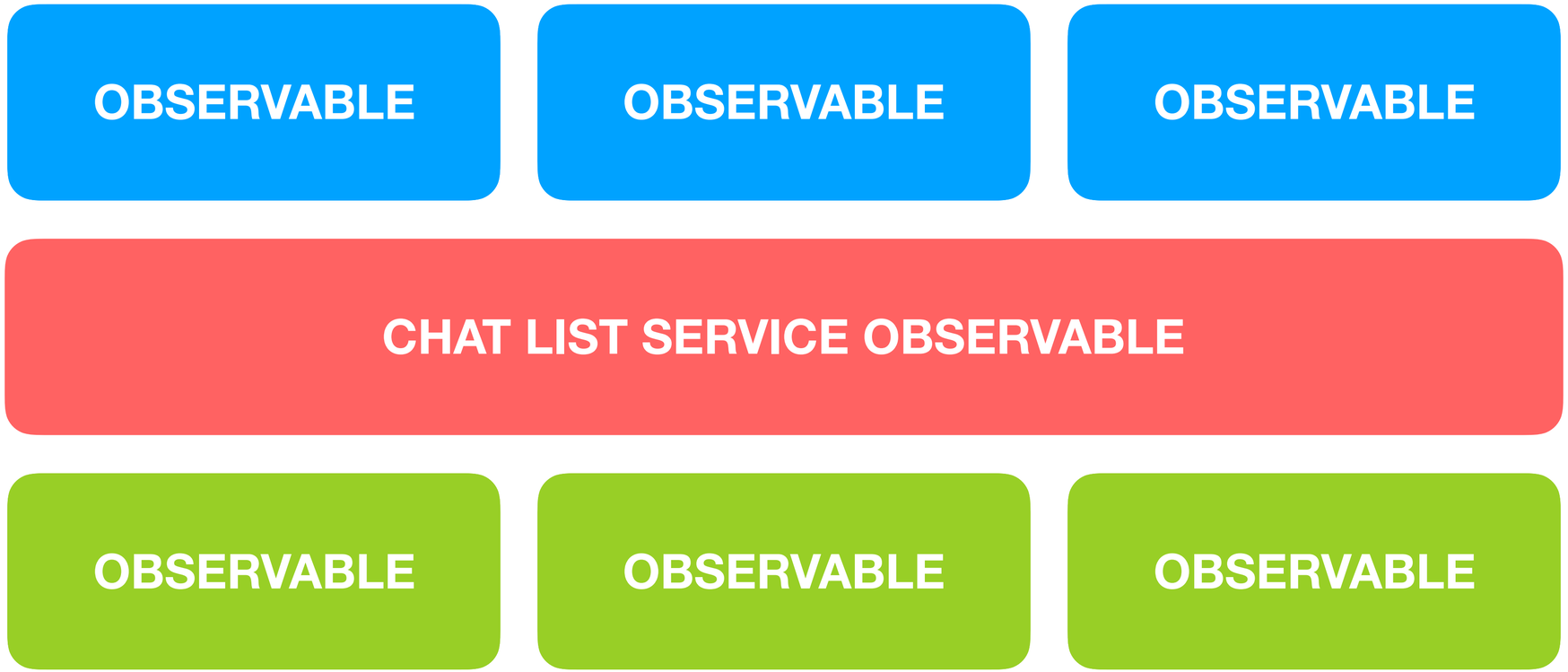
Шаблон проектирования Наблюдатель

```
protocol UsersOnlineStatusObservable: class {

    func add(
        observer: AnyObject,
        onUsersOnlineStatusChange:
            @escaping (([User]) -> ())
    )

    func remove(observer: AnyObject)
}
```

```
protocol UsersOnlineStatusService: class {

    func startPollingOnlineStatuses()
}
```

```
final class UsersOnlineStatusServiceImpl:

    UsersOnlineStatusService,
    UsersOnlineStatusObservable {


}
```

```
// MARK: - state
private var chats: [Chat] {
    didSet {
        notifyObservers()
    }
}

func subscribeOnUpdates() {

    usersOnlineStatusServiceObservable.add(
        observer: self,
        onUsersOnlineStatusChange: { users in
            // update chats
        }
    )
    newMessagesServiceObservable.add(
        observer: self,
        onNewMessage: { newMessage in
            // update chats
       }
    )
}
```


OBSERVABLE
OBSERVABLE
OBSERVABLE
CHAT LIST SERVICE OBSERVABLE
OBSERVABLE
OBSERVABLE
OBSERVABLE

ObserverList<T>.swift

http://gist.github.com

Выводы

Микросервисы - небольшие блоки, соединяющиеся в единую систему

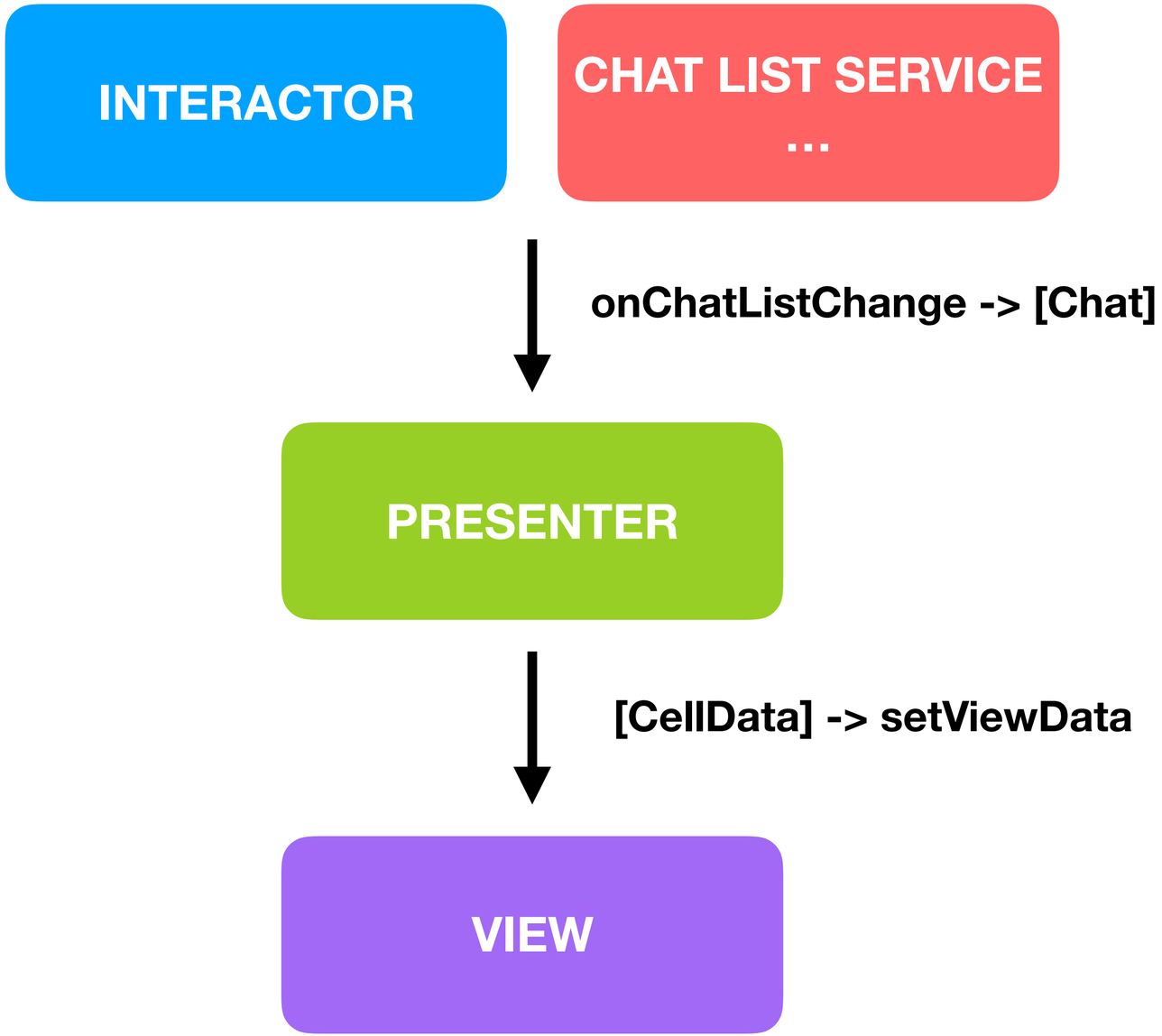
Консистентность - все сервисы спроектированы одинаково

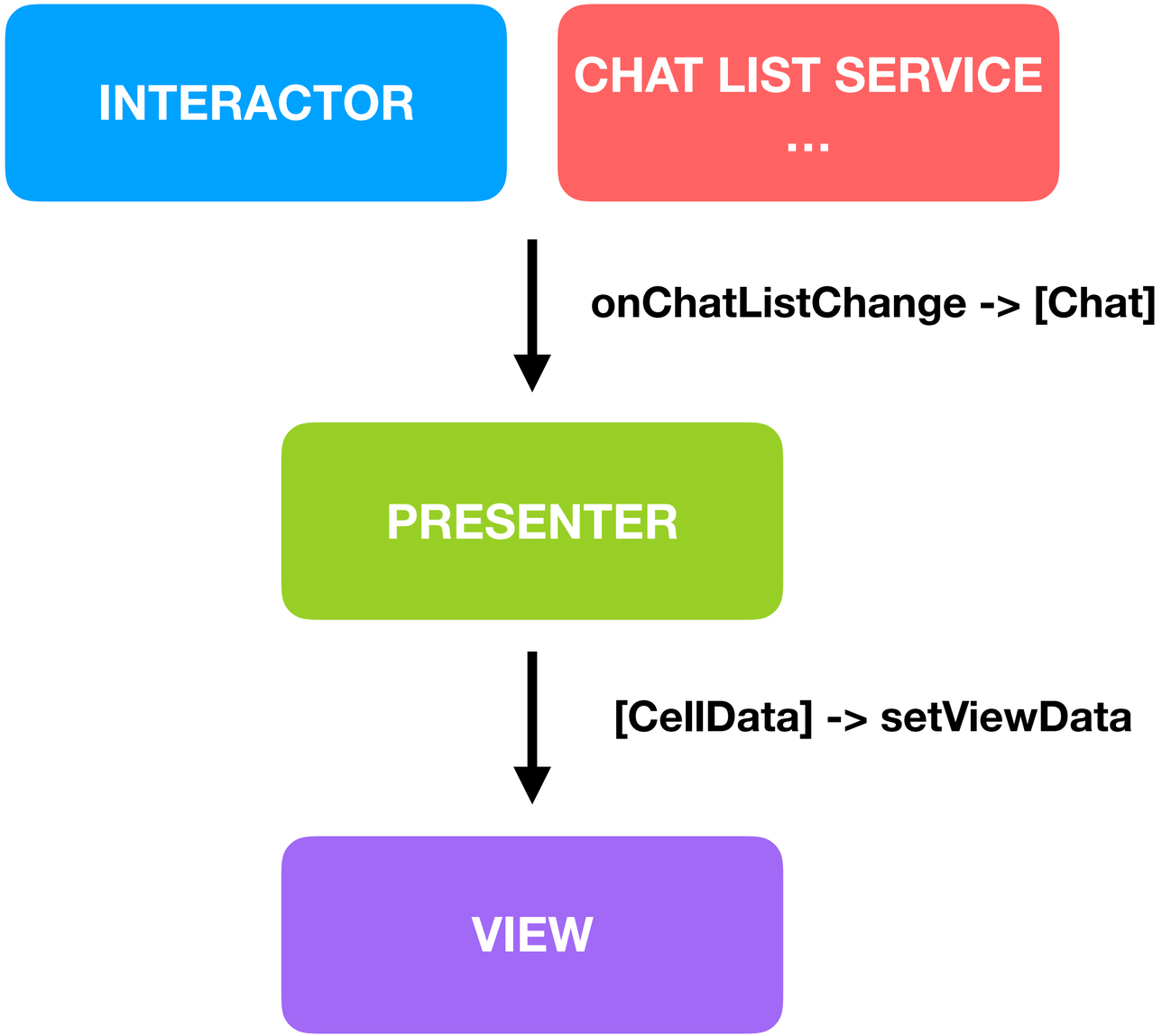
План

Проектирование API слоя взаимодействия с сервером

Проектирование сервисного слоя

Трюки и советы


INTERACTOR
CHAT LIST SERVICE
...
onChatListChange -> [Chat]
PRESENTER
[CellData] -> setViewData
VIEW


INTERACTOR
CHAT LIST SERVICE
...
onChatListChange -> [Chat]
PRESENTER
[CellData] -> setViewData
VIEW

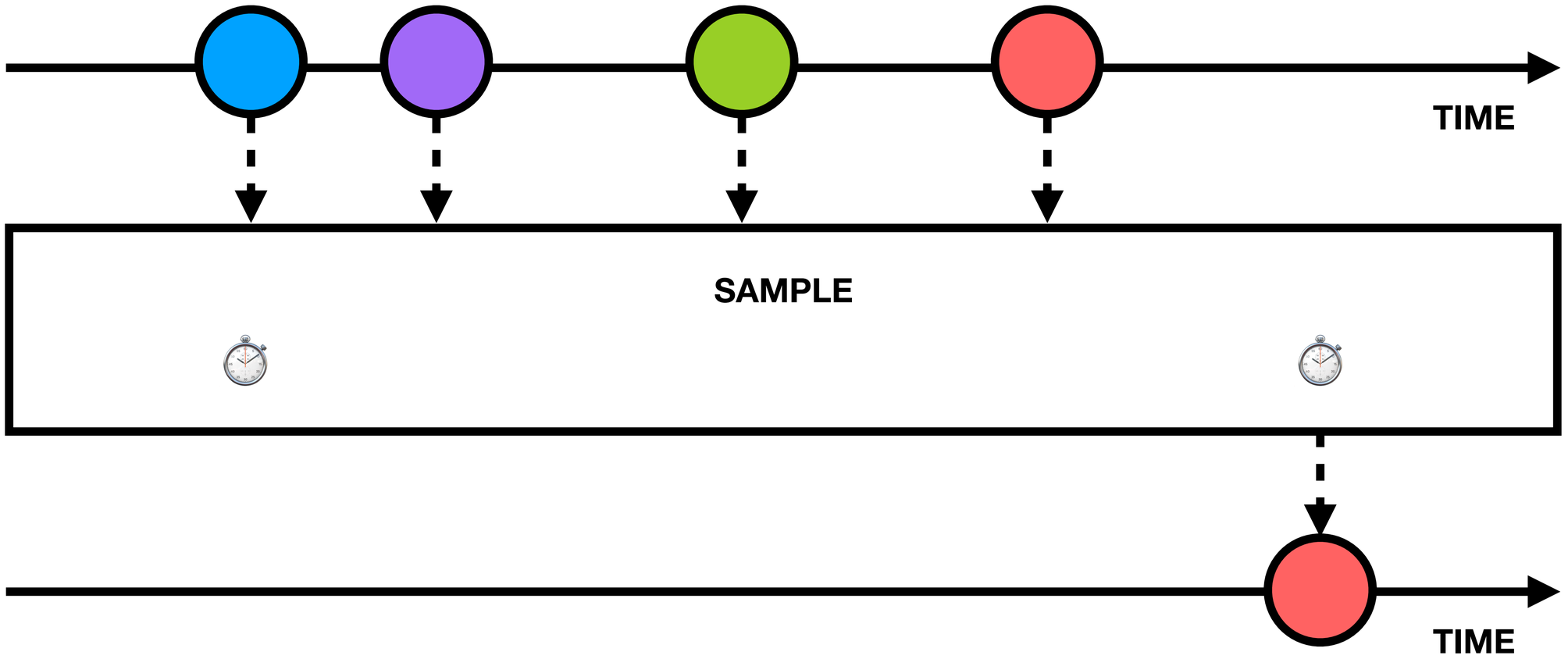
Проблема

Слишком частые обновления UI

DEBOUNCE THROTTLE SAMPLE DELAY

Sample Operator

Emit the most recent items emitted by an Observable within periodic time intervals


TIME
SAMPLE
TIME

```
let sampler = Sampler(delay: 0.5)

sampler.sample {

    // update ui here...
}
```

Sampler.swift

http://gist.github.com

Проблема

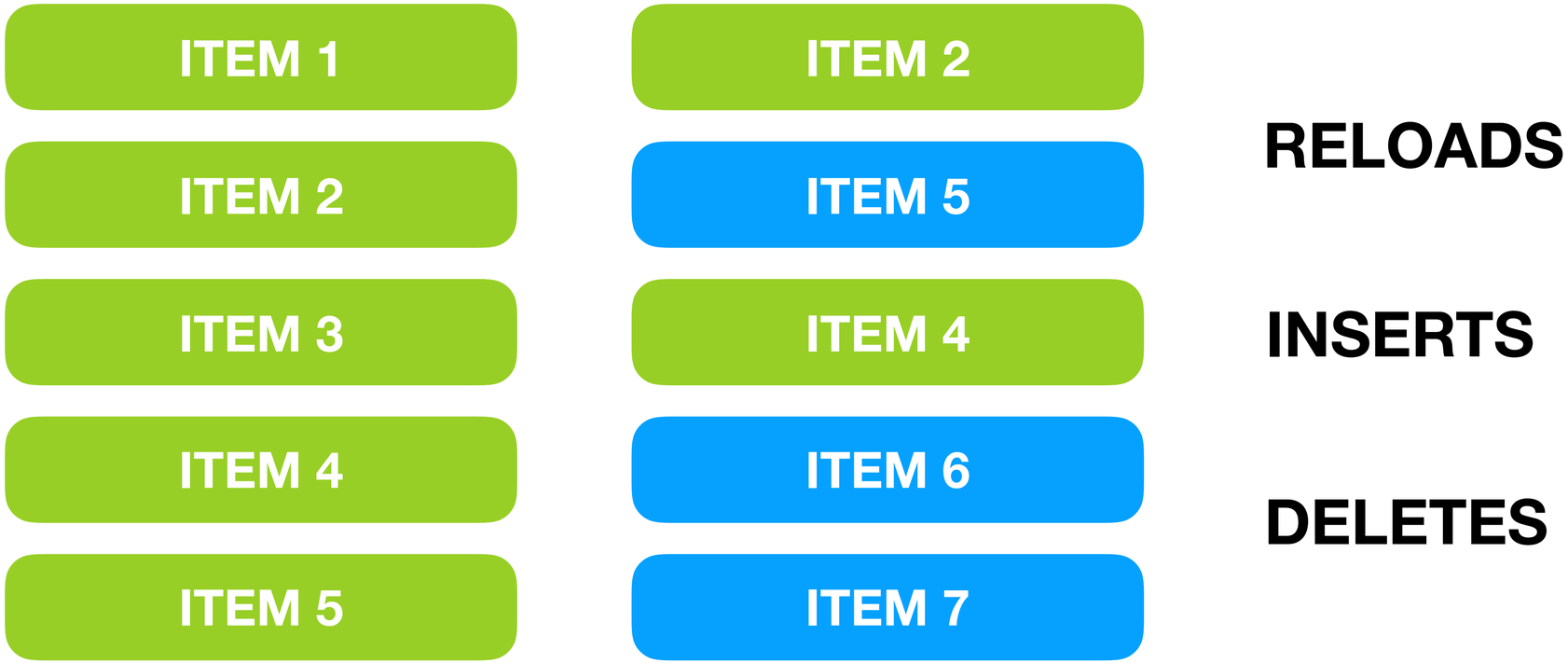
Обновление UI, если пользователь покинул экран

```
final class ChatListViewController: UIViewController {

    func processOnChatListUpdates(_ chats: [ChatViewData]) {

        invokeClosuresHelper.invokeOrDelayIfViewIsNotReady {
            // do ui updates...
        }
    }

    override func viewDidAppear(_ animated: Bool) {
        super.viewDidAppear(animated)

        invokeClosuresHelper.setViewIsReady(true)
    }
}
```

Проблема

UITableView / UICollectionView
Обновление элементов, поиск index path изменившихся

Diff алгоритм



Выводы

Выбирайте инструмент под задачу

Разделяйте логику работы с API и бизнес логику

Закладывайте масштабируемость

Вопросы?

@MAX_SOKOLOV
https://twitter.com/max_sokolov

/MAXSOKOLOV
https://github.com/maxsokolov
https://gist.github.com/maxsokolov

masokolov@avito.ru