РАЗРАБОТКА СОВРЕМЕННОЙ РАСПРЕДЕЛЁННОЙ СИСТЕМЫ: ПРОБЛЕМЫ И РЕШЕНИЯ

Михаил Пересыпкин

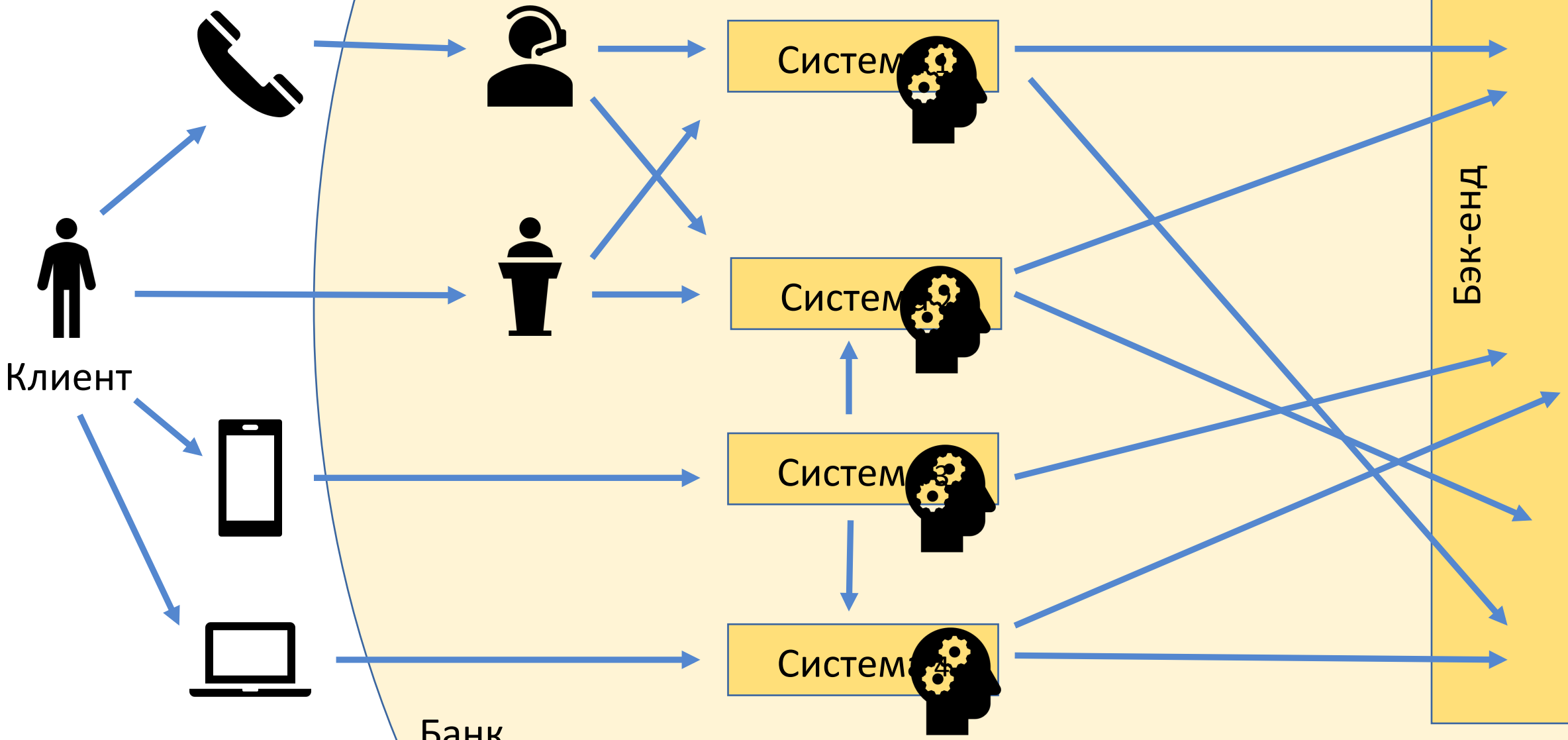
Единая Фронтальная Система

О себе

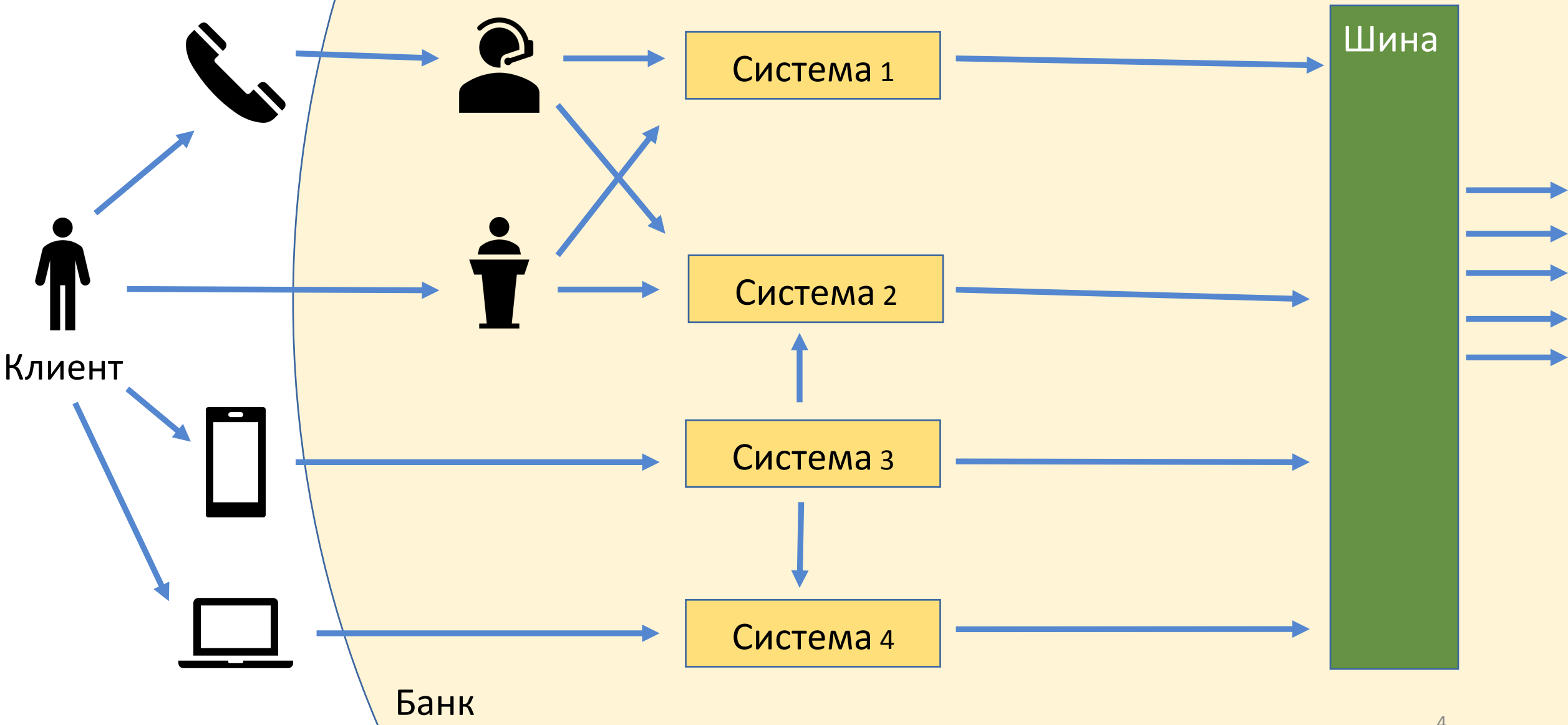
- С 1998 года разрабатывал на C/C++, что-то слышал про Java — 100
- С 2000 по 2010: Ростелеком: стек J2SE/J2EE+Oracle — 10 000
- С 2011 по 2014: Альфа-Капитал: ReactJS+J2SE+Oracle — 100 000
- С 2015 по 2016: Сбербанк-Технологии: ЕФС для Чехии и Хорватии — 1 000 000
- С 2017: Сбербанк-Технологии: ЕФС для Сбербанка РФ — 50 000 000

Единая Фронтальная Система

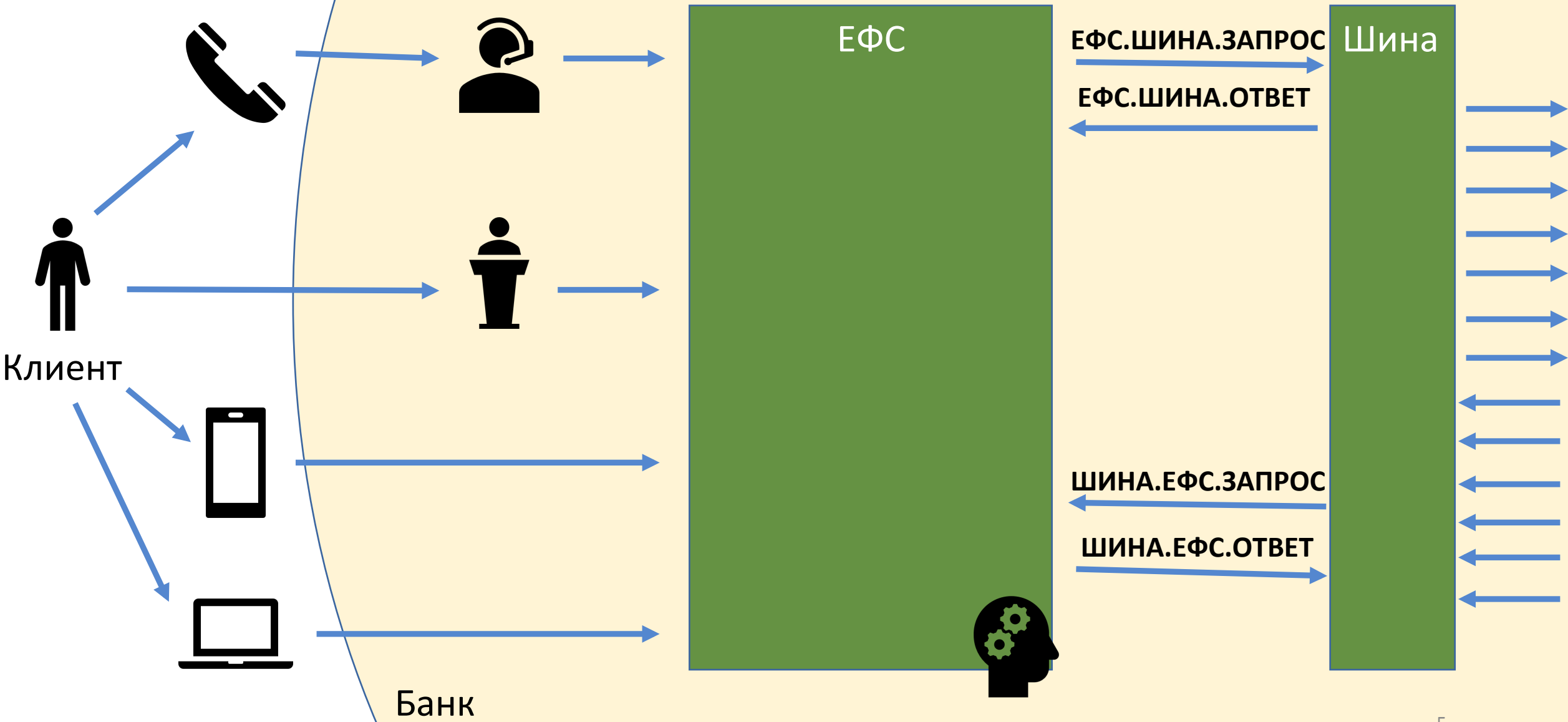
О Единой Фронтальной Системе



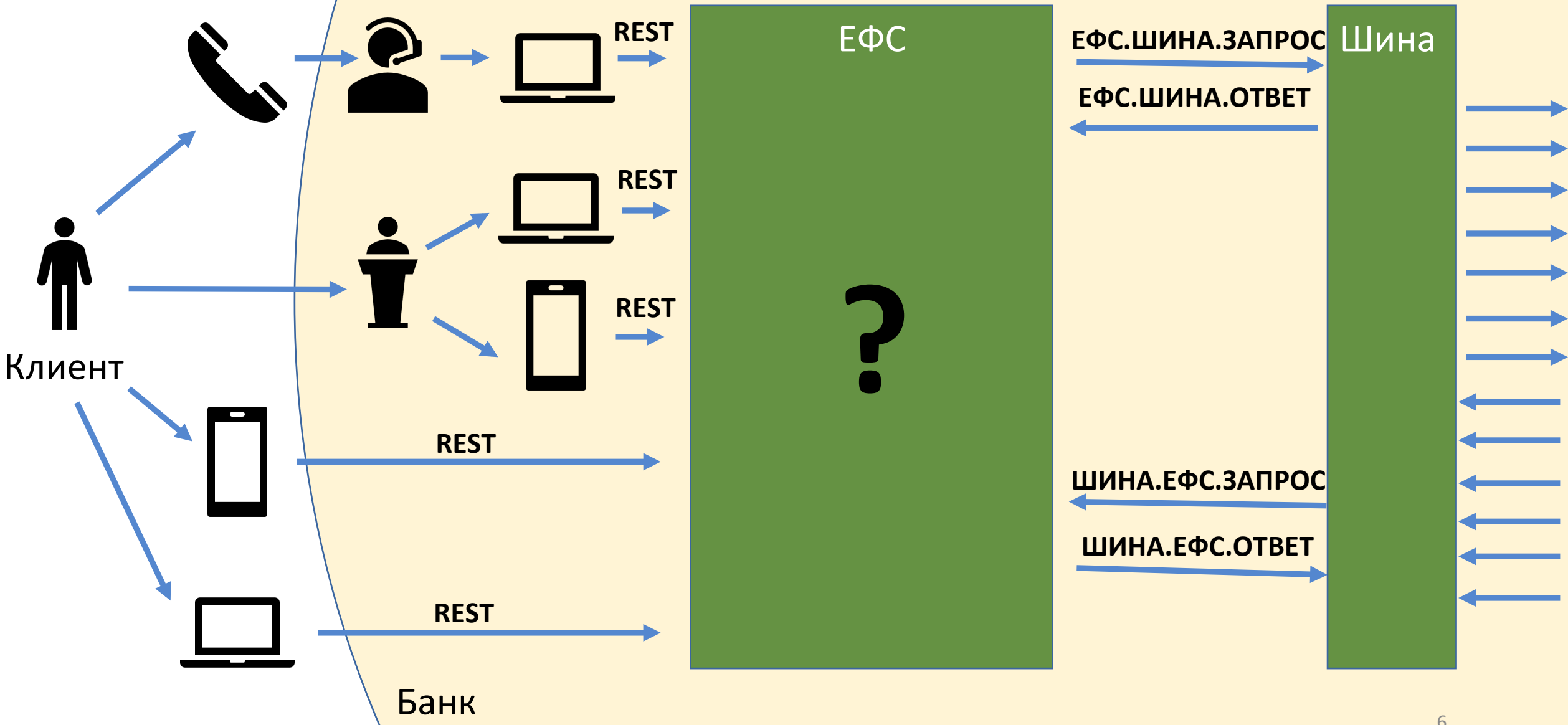
О Единой Фронтальной Системе



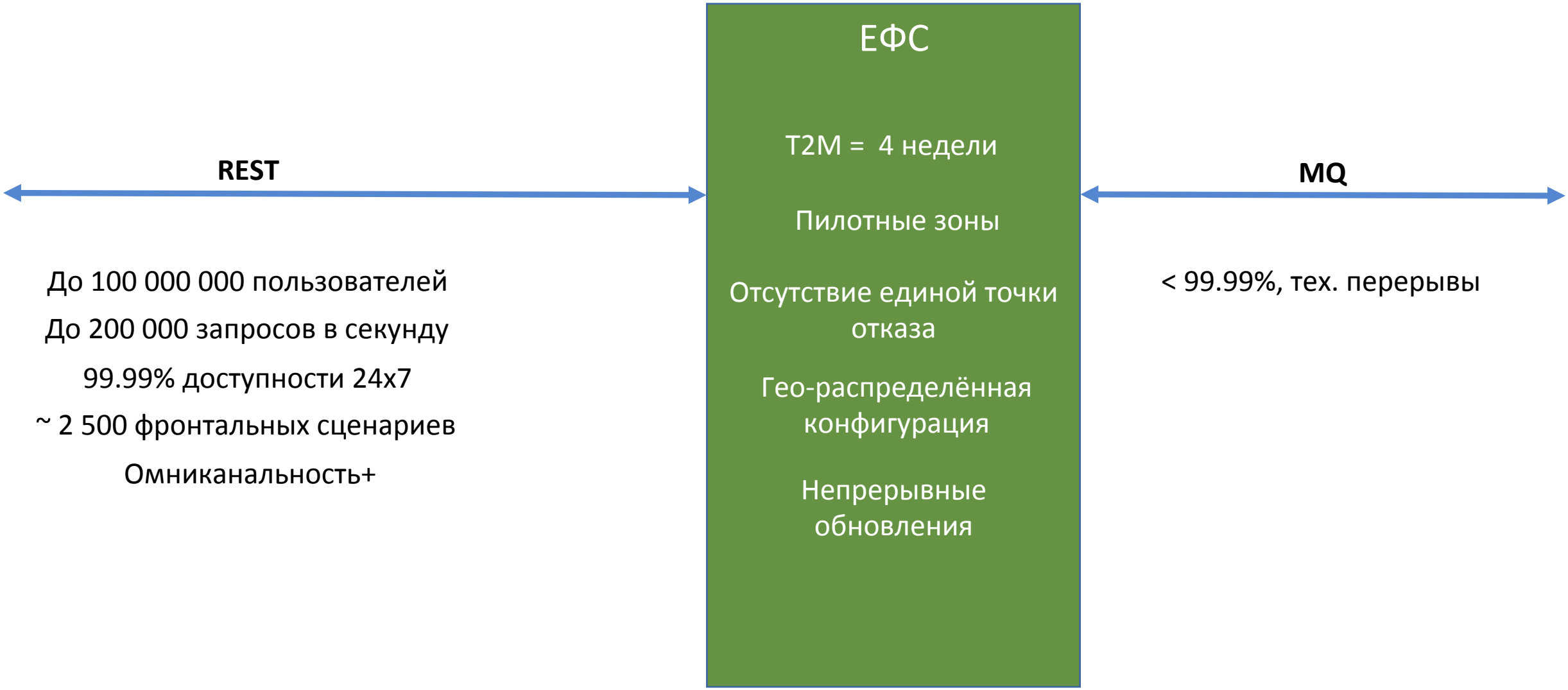
О Единой Фронтальной Системе



О Единой Фронтальной Системе



Постановка задачи



До 100 000 000 пользователей

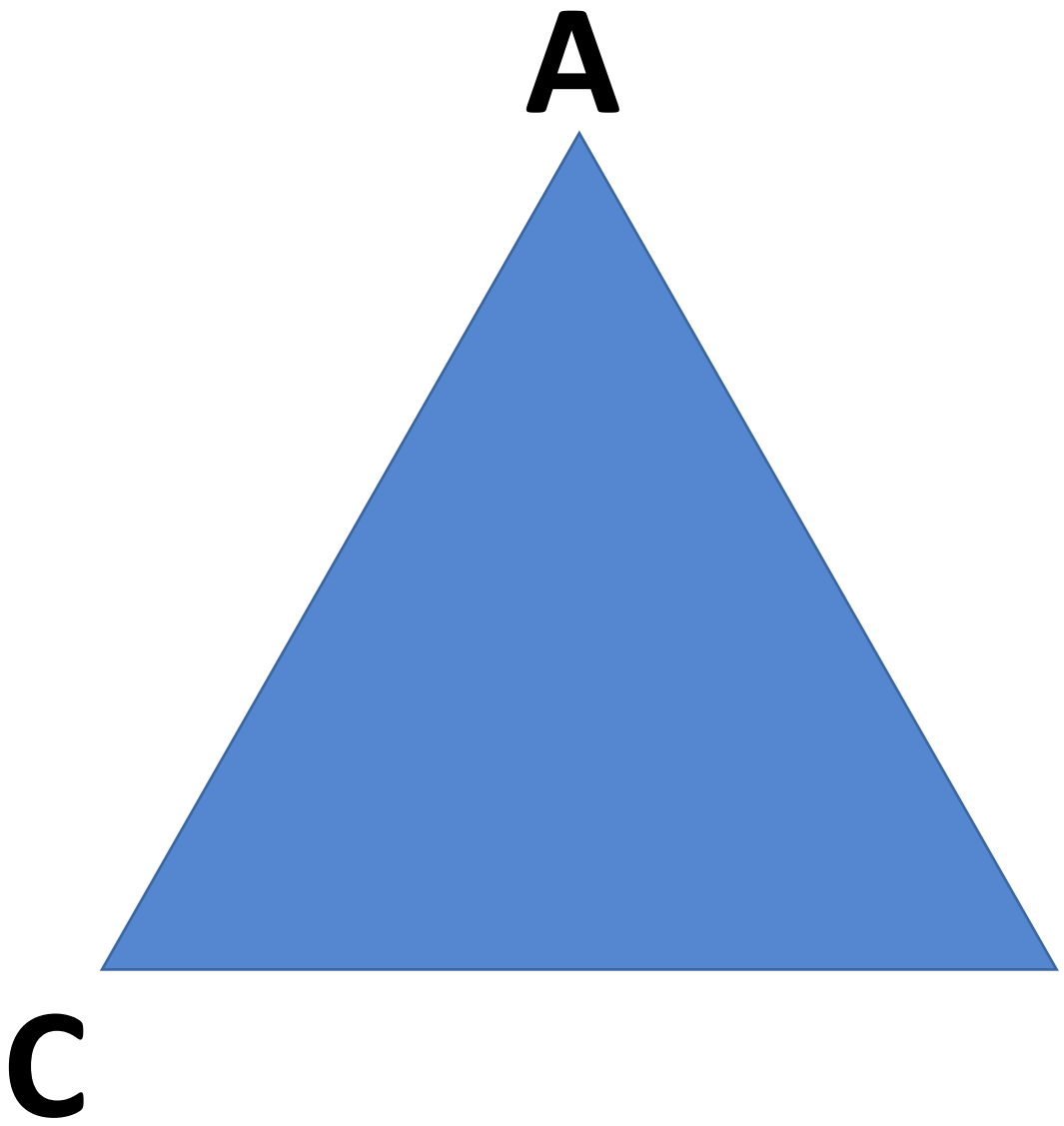
До 200 000 запросов в секунду

99.99% доступности 24x7

~ 2 500 фронтальных сценариев

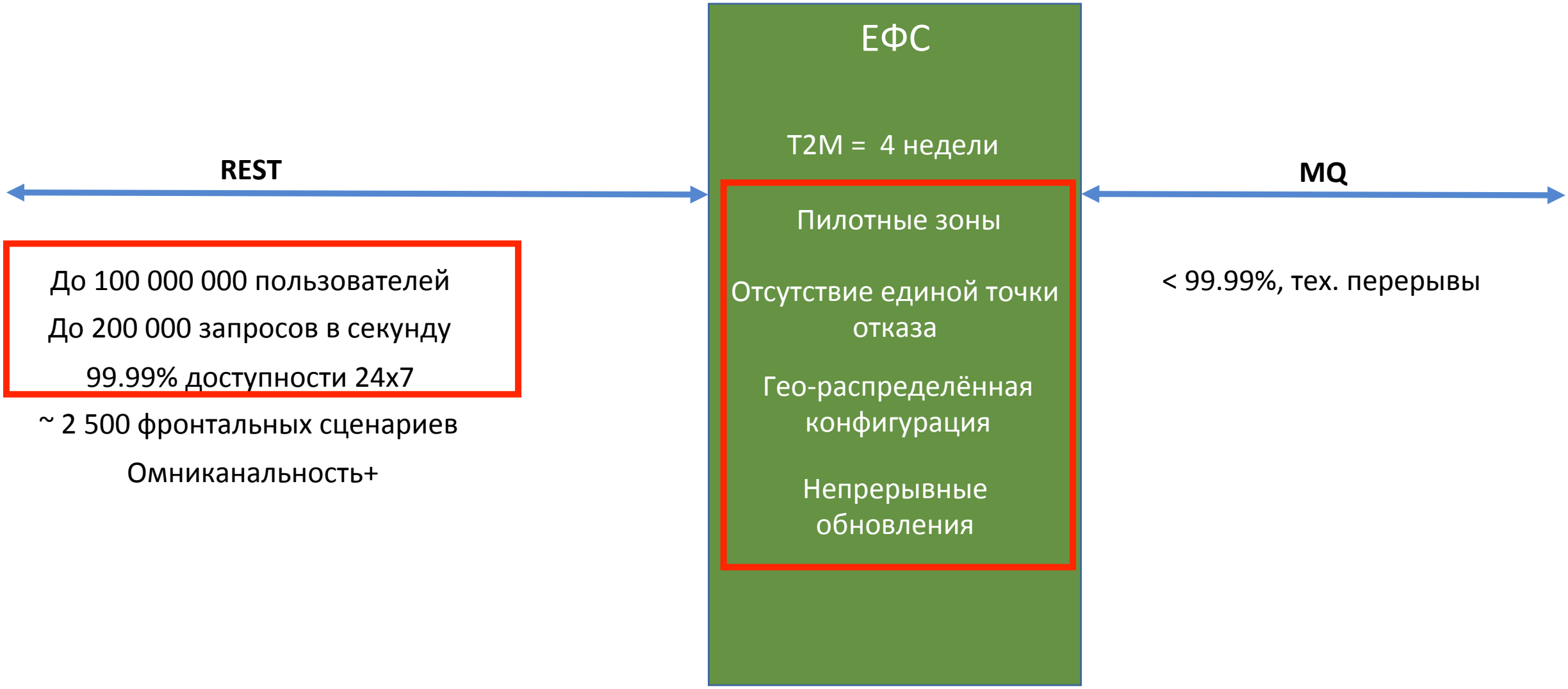
Омниканальность+

< 99.99%, тех. перерывы

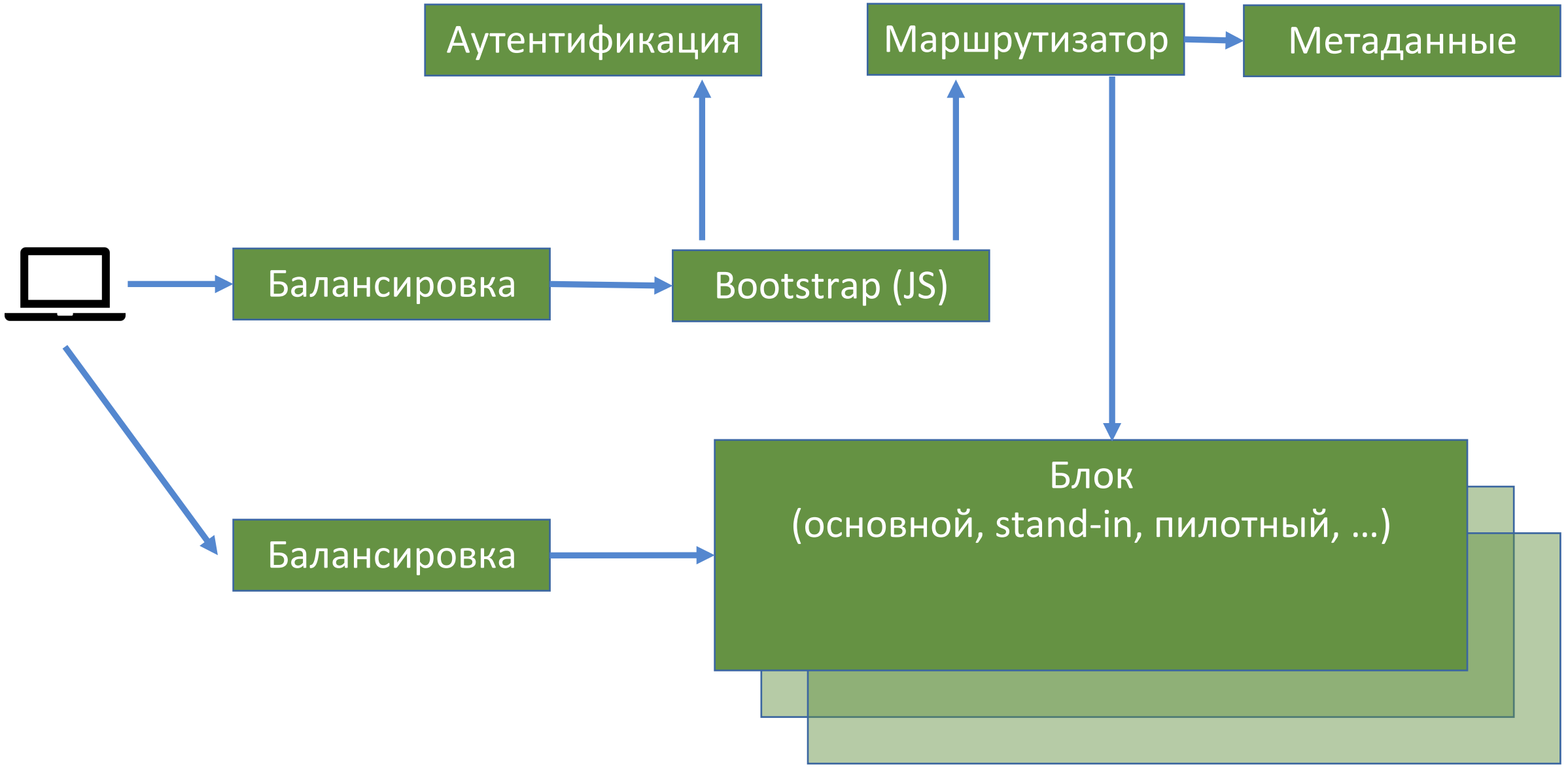


AP

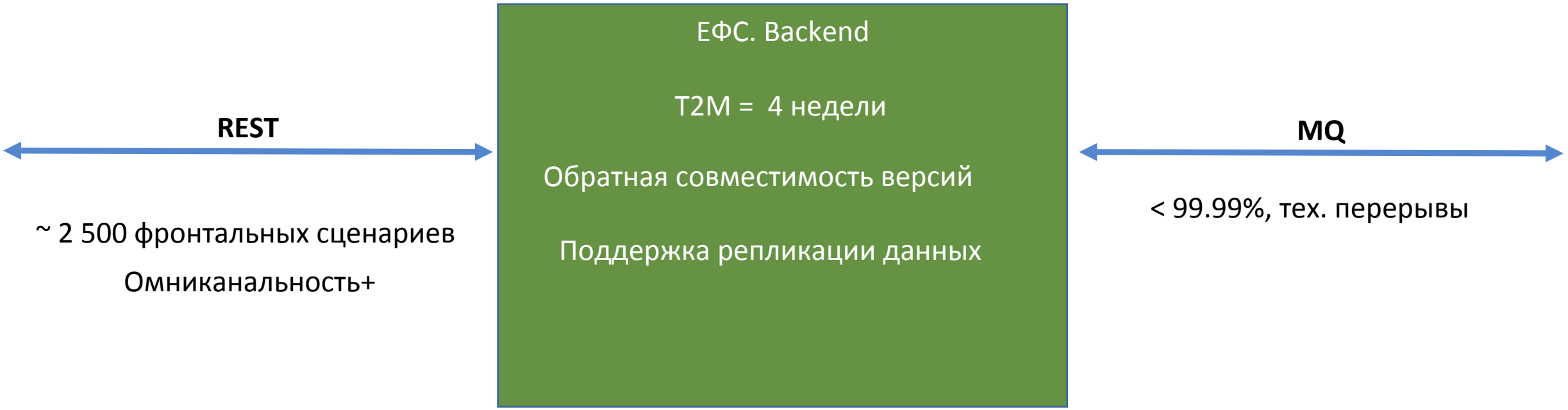
Постановка задачи



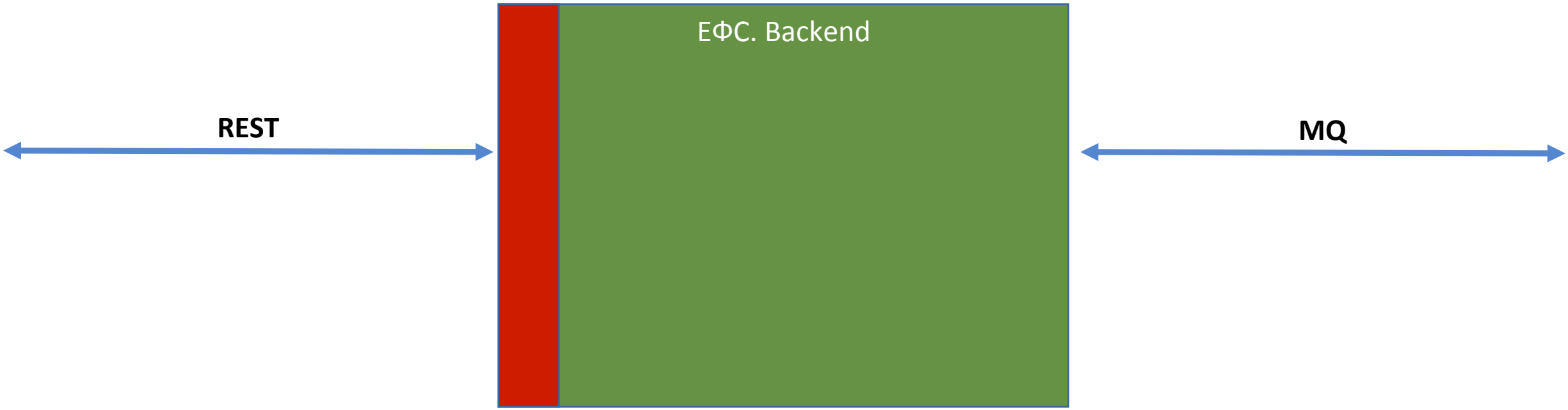
Многоблочность



А что внутри?


REST
~ 2 500 фронтальных сценариев
Омниканальность+
ЕФС. Backend
T2M = 4 недели
Обратная совместимость версий
Поддержка репликации данных
MQ
< 99.99%, тех. перерывы

Фронт



Как выбирали решение?

Очень много объектов, очень много процессов, очень быстро разрабатывать

Переиспользование на уровне UI-компонент

Переиспользование на уровне процессов/подпроцессов

Инструментальные средства

Activiti !

Описание процессов не декомпозируется на отдельные компоненты

Не выдержал нагрузки. Узкое место – БД.

Сложно переделать на JCache

Spring Webflow !!

Расчёт на Sticky Session

JCache

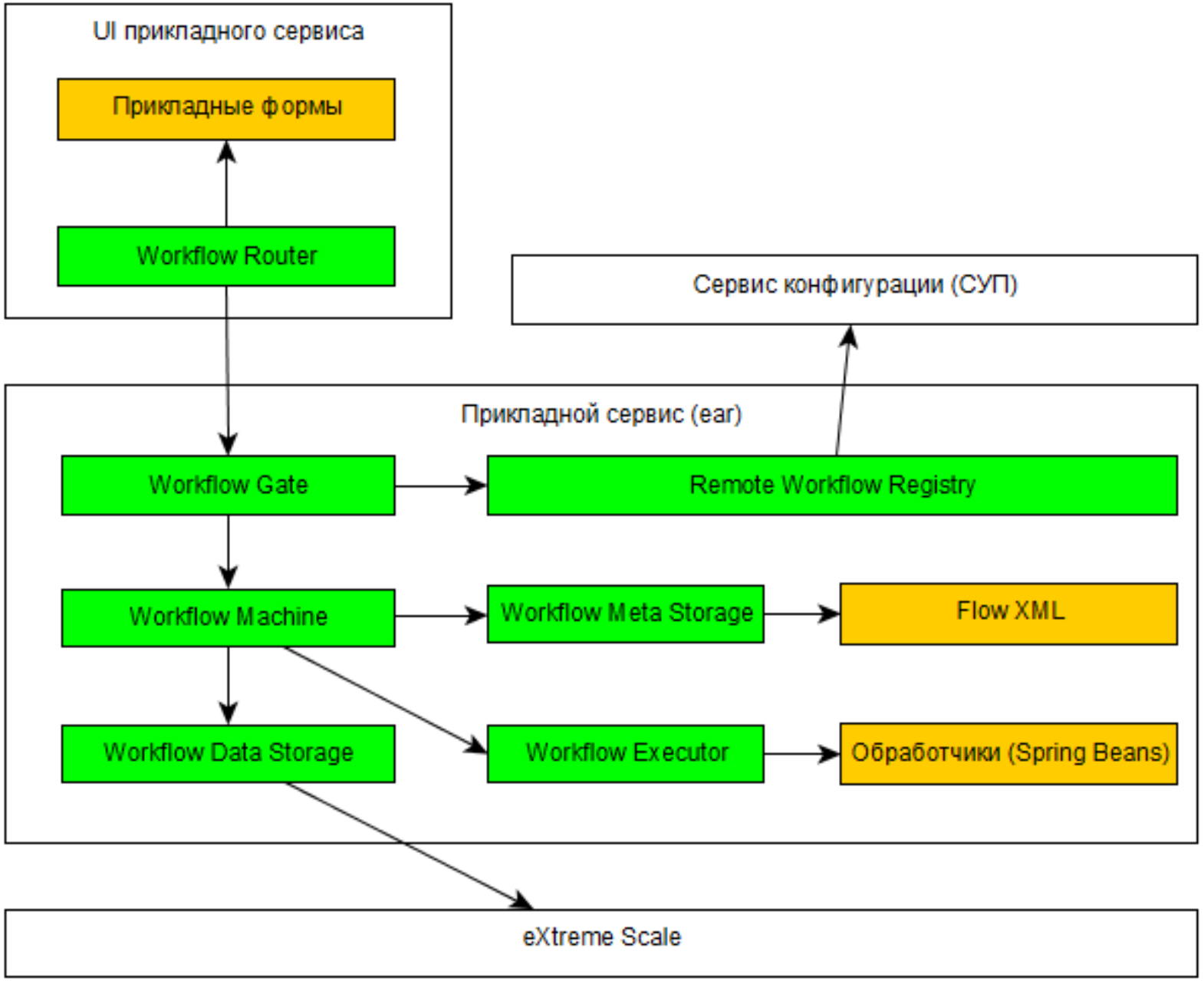
ServerSide Rendering -> SPA

Потеряли возможность обновляться

Результат

```
<flow ...>
        <input name="selectedDeposit" value="flowScope.selectedDeposit" required="true" />
        <input name="initAction" value="flowScope.action" required="true" />
        <on-start>
                <set name="operationName" value="'Печать операций в сберкнижку'"/>
                <set name="selectedDeposit" value="new DepositAgreementForSavingDto(selectedDeposit)" />
                <set name="flowScope.actionStatus« value="T(SbookFlowActionStatus).NOT_STARTED" />
                <set name="flowScope.initFlag" value="true" />
        </on-start>
        <action-state id="loadDeposits">
                <evaluate expression="sbookFlowService.initDepositForSaving()"/>
                <transition on="OK" to="updateUnseenOpCount"/>
        </action-state>
        <action-state id="updateUnseenOpCount">
                <evaluate expression="sbookFlowService.updateDepositUnseenOpCount(selectedDeposit)"/>
                <transition on="OK" to="resolveNextState"/>
        </action-state>
        <view-state id="sbookPrintView">
                <on-render>
                        <set name="viewScope.response" value="flowResponse"/>
                </on-render>
                <transition on="next" to="sbookPrintConfirmView">
                        <evaluate expression="jsonParser.getFromJson(T(FlowRequest))“ result="flashScope.errorMsg" />
                </transition>
        </view-state>
```

Workflow


UI прикладного сервиса
Прикладные формы
Workflow Router
Сервис конфигурации (СУП)
Прикладной сервис (ear)
Workflow Gate
Remote Workflow Registry
Workflow Machine
Workflow Meta Storage
Flow XML
Workflow Data Storage
Workflow Executor
Обработчики (Spring Beans)
eXtreme Scale

Стало

```
<flow…>

   <state name="login" title="Введите Ваш логин и пароль">
      <event name="authorize" handler="loginHandler#login"/>
      <state-transition name="success" to="smsAuthenticate"/>
      <state-transition name="error" to="login"/>
      <state-transition name="blocked" to="end"/>
   </state>

   <state name="smsAuthenticate" title="Подтвердите вход">
      <event name="smsAuthenticate" handler="loginHandler#smsAuthenticate"/>
      <event name="return" handler="loginHandler#back"/>
      <state-transition name="success" to="end"/>
      <state-transition name="back" to="login"/>
      <state-transition name="error" to="login"/>
   </state>

</flow>
```

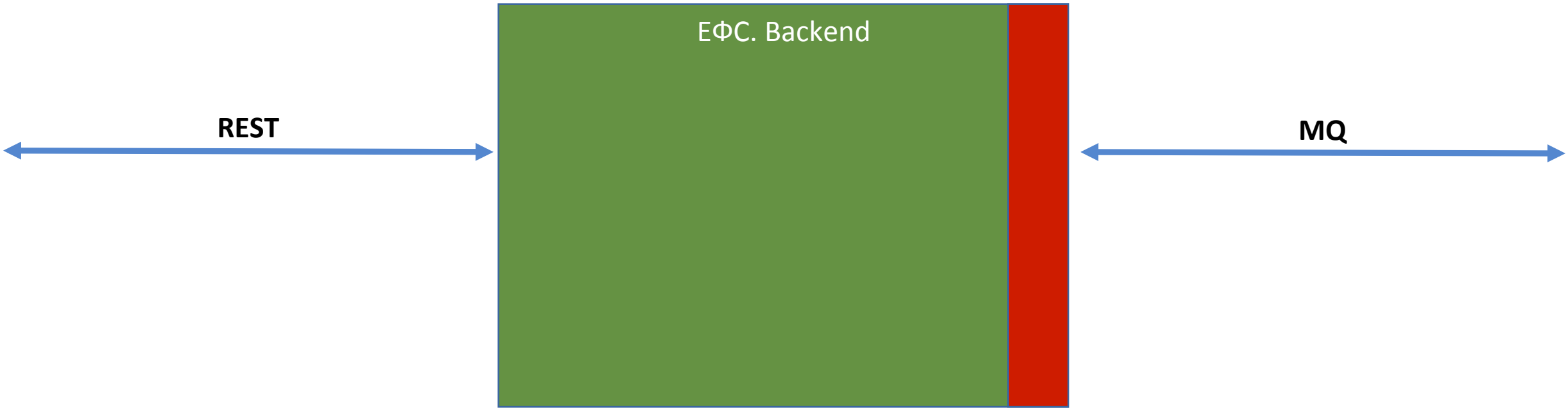
Стало

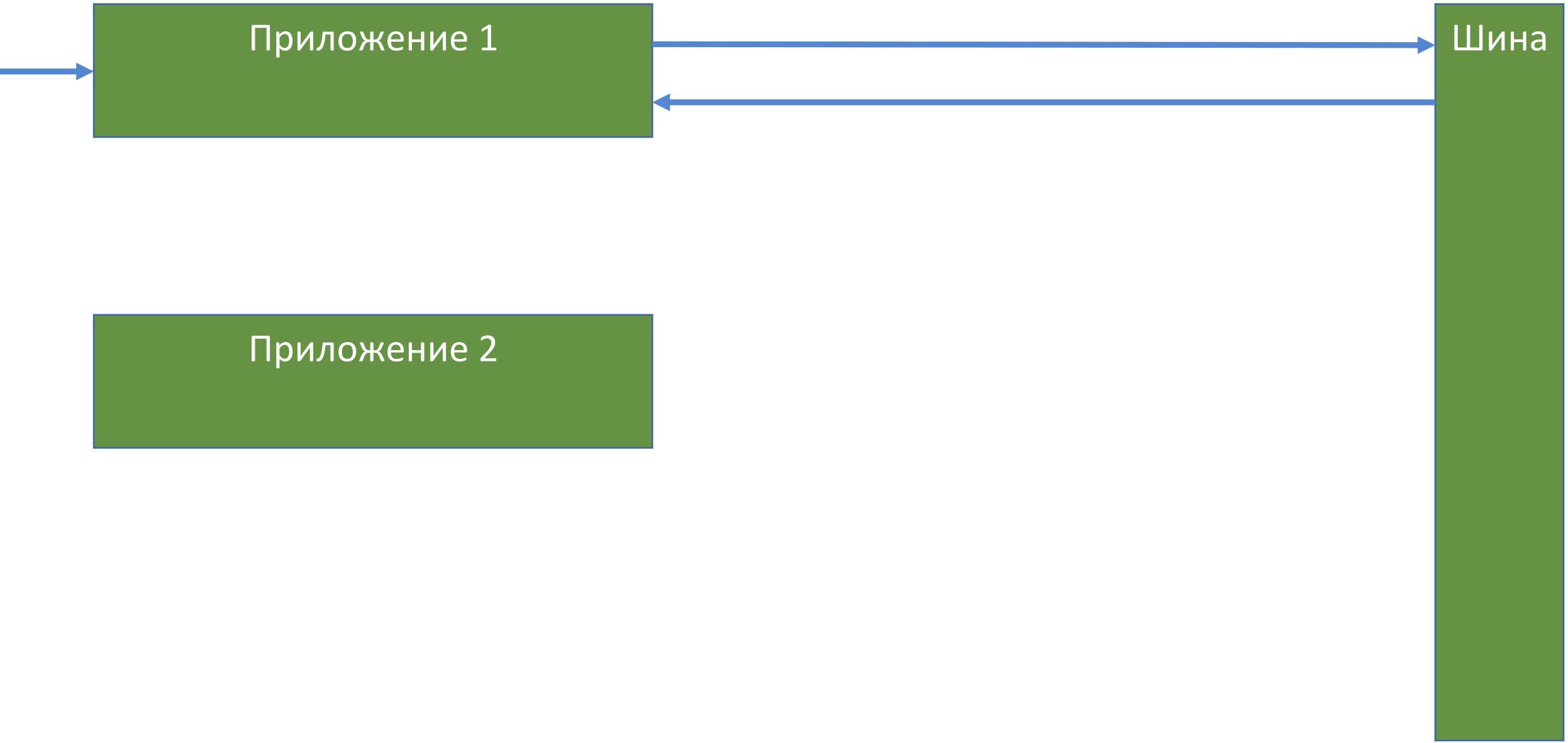
```
public Transition smsAuthenticate() {
        String smsCode = context.parseInputToJson(User.class).getCode();
        if (loginService.smsLogin(smsCode)) {
            logService.info(LoggerMessage.AUTHENTICATION_SUCCESS);
            return go("success");
        } else {
            logService.info(LoggerMessage.WRONG_SMS_CODE);
            return go("error");
        }
    }
```

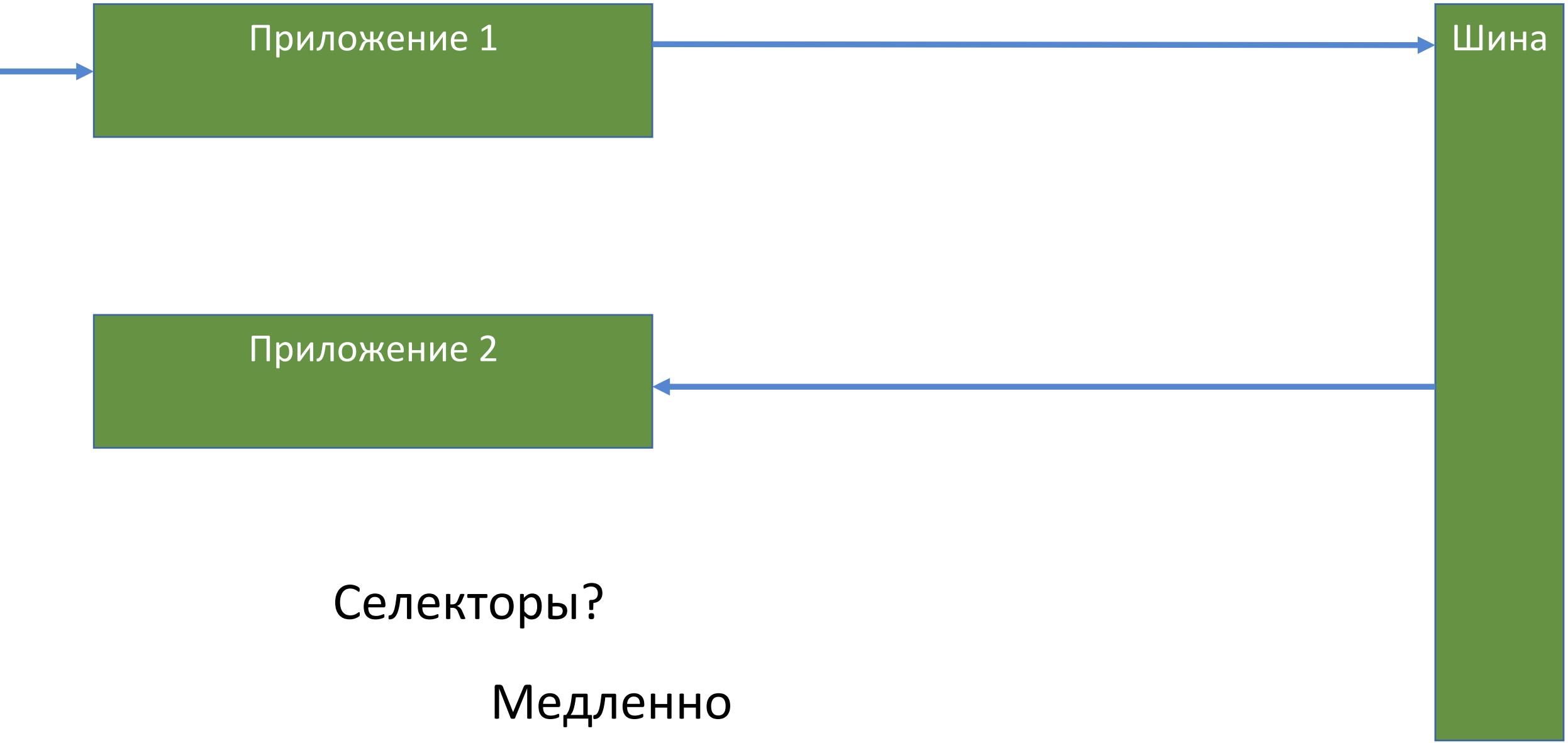
В цифрах

- Размер исходников Webflow ~= 35 500 LoC
- Наши патчи в Webflow ~= 3000 LoC
- Workflow на данный момент = 5500 LoC
- 1 ч/месяц на старте

Интеграция

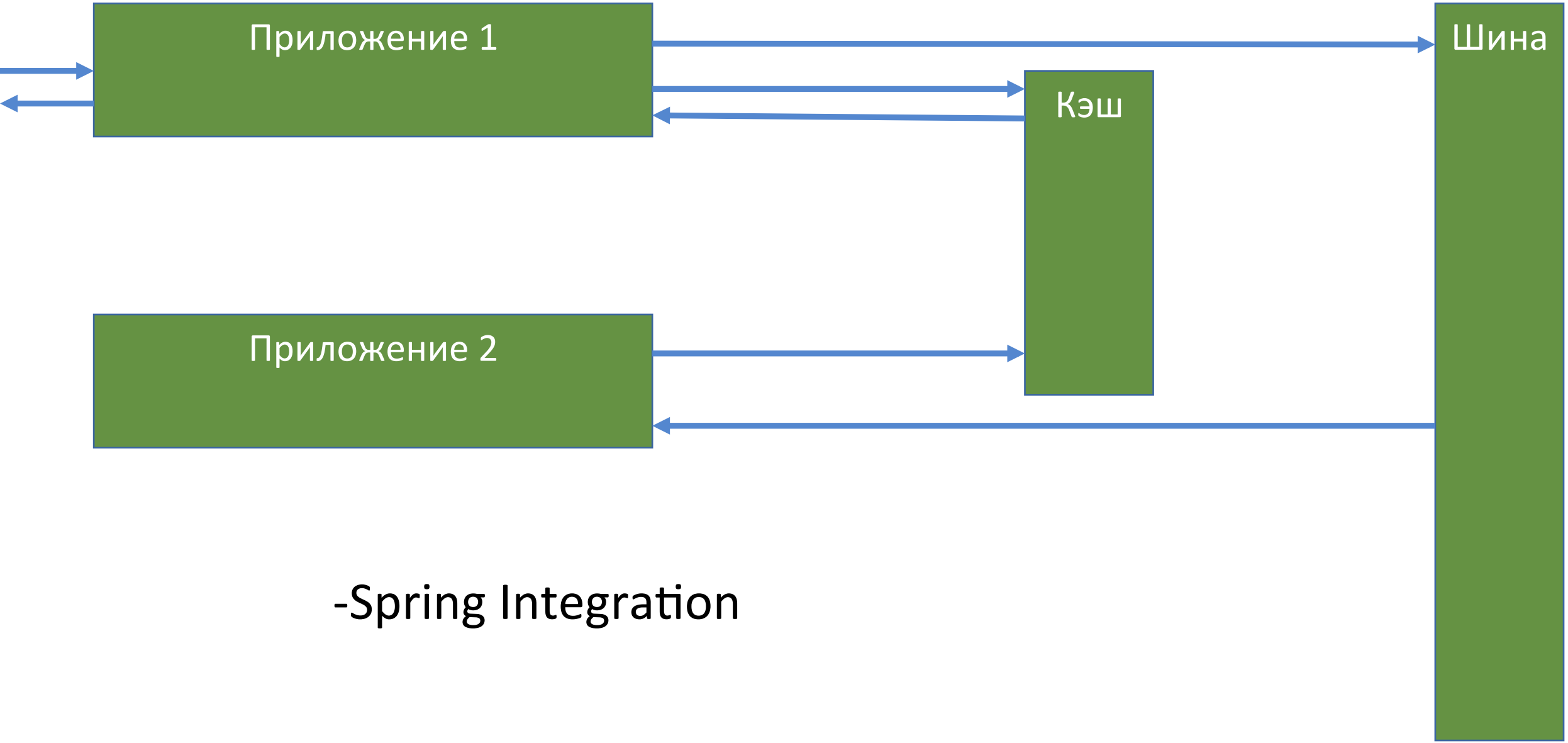



Приложение 1
Шина
Приложение 2

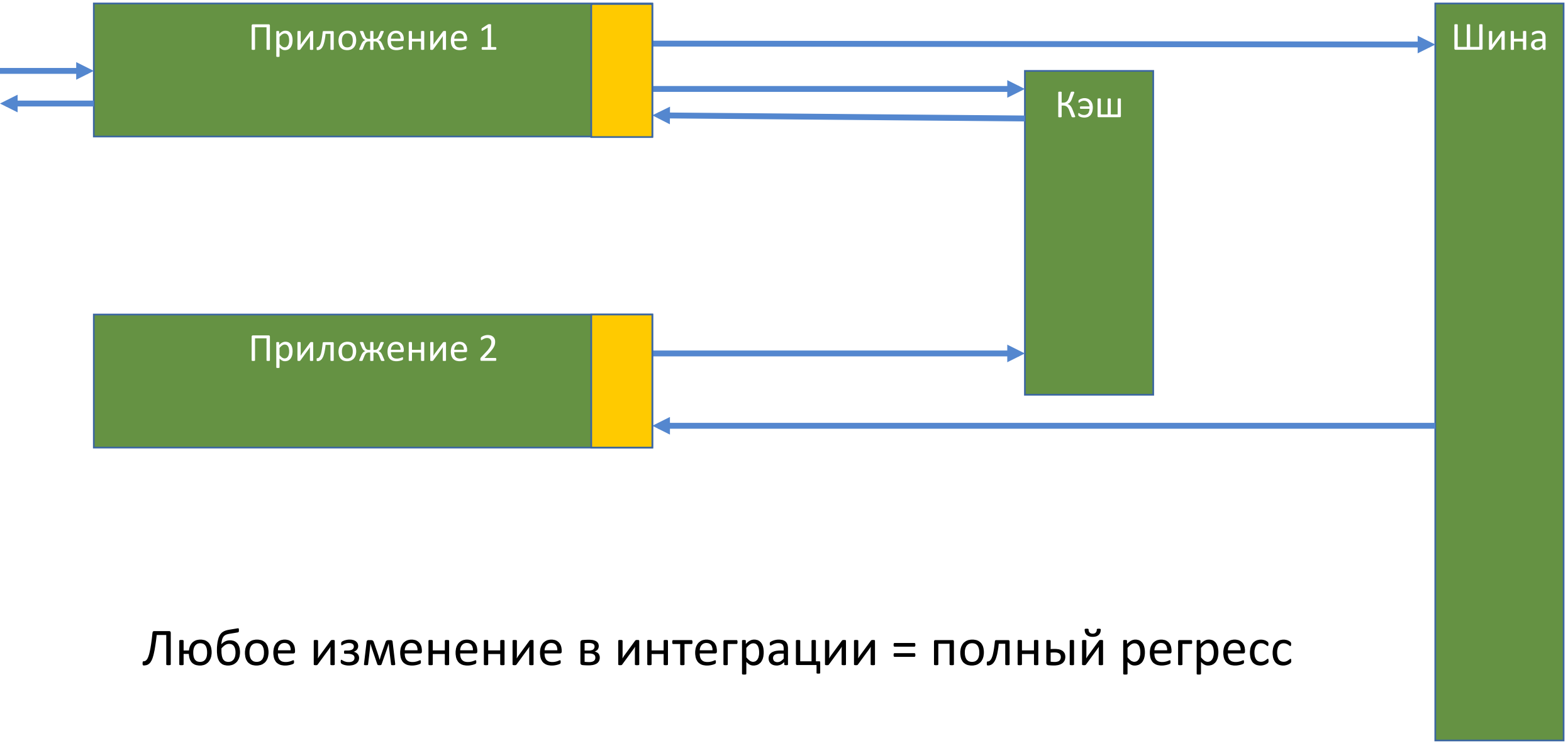


Селекторы?

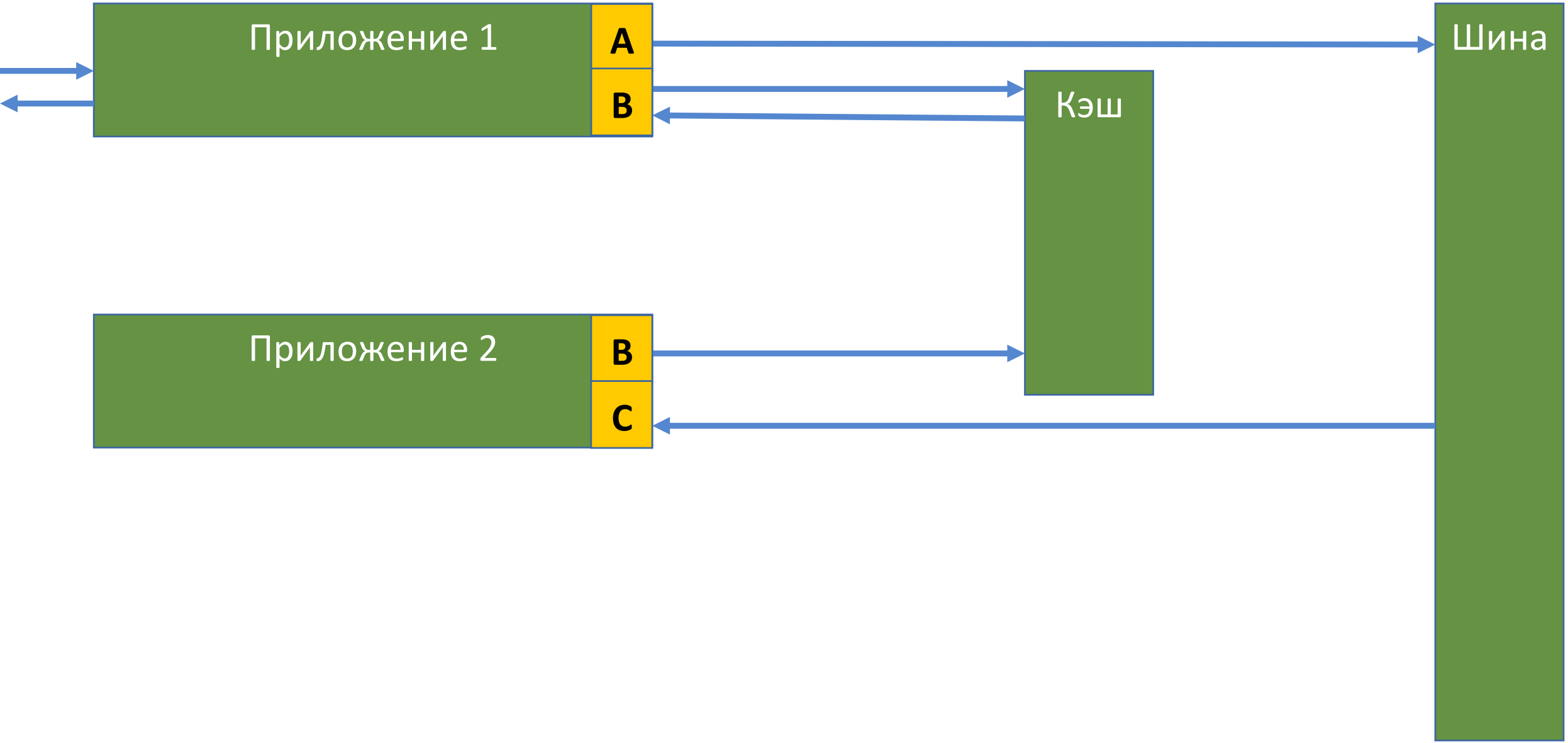
Медленно



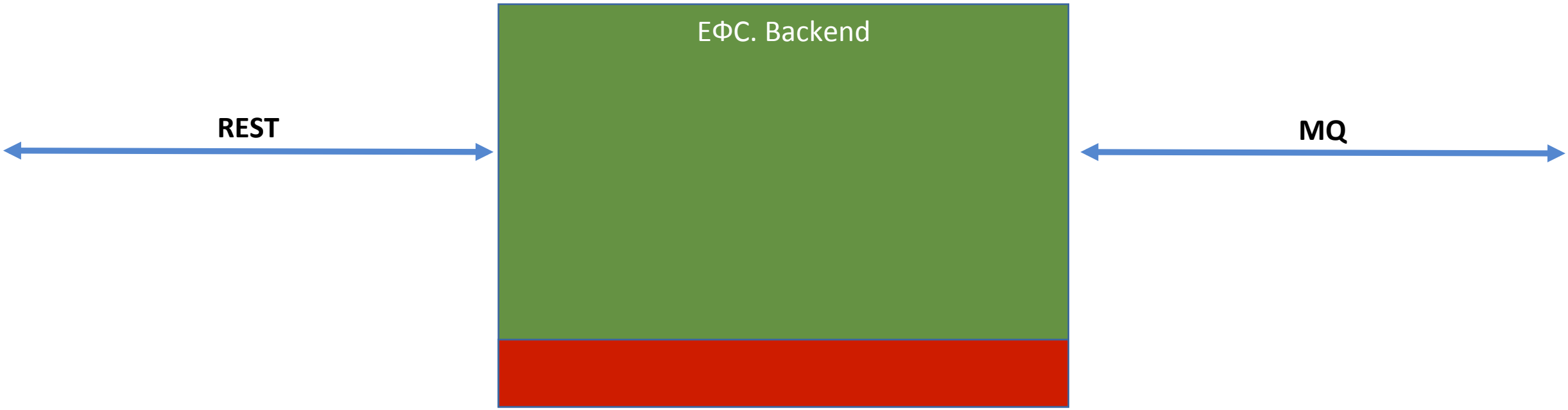
-Spring Integration



Любое изменение в интеграции = полный регресс


Приложение 1
A
B
Кэш
Шина
Приложение 2
B
C

Базовые системные сервисы



Базовые системные сервисы

Изоляция от инфраструктуры

Журналирование и аудит

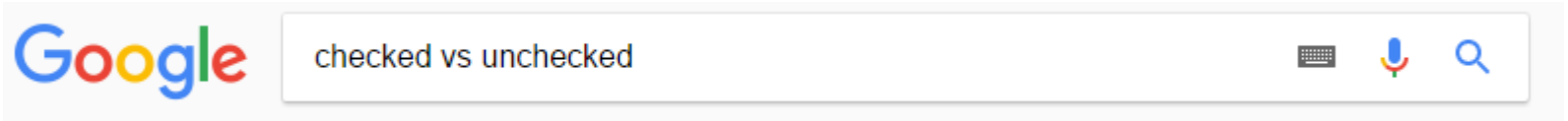
Управление параметрами

Периферия

White-list зависимостей

Абстракции (технологические и прикладные)

За гранью абстракции ;)

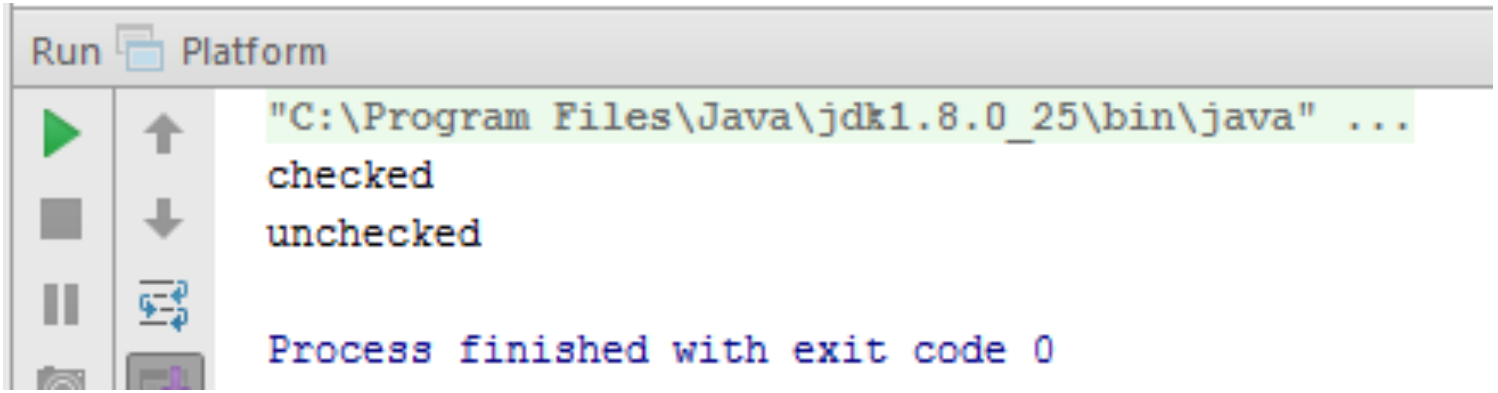



```
public void hi() {
    ...
}
```

```
public static void main(String[] args) {
    try {
        new Platform(Exception.class).hi();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }

    try {
        new Platform(RuntimeException.class).hi();
    } catch (RuntimeException ex) {
        System.out.println(ex.getMessage());
    }
}
```

```java
public class Platform<T extends Exception> {
    private Class<T> clazz;

    public Platform(Class<T> clazz) {
        this.clazz = clazz;
    }

    public void hi(String message) throws T {
        throw exception(message);
    }

    private T exception(String message) {
        try {
            T ex = clazz.getConstructor(String.class).newInstance(message);
            return ex;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
```

```
public static void main(String[] args) {
    try {
        Platform<Exception> v1 = new Platform<>(Exception.class);
        v1.hi("checked");
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }

    try {
        Platform<RuntimeException> v2 = new Platform<>(RuntimeException.class);
        v2.hi("unchecked");
    } catch (RuntimeException ex) {
        System.out.println(ex.getMessage());
    }
}
```



Спасибо за внимание!

Михаил Пересыпкин
MAPeresypkin.SBT@sberbank.
/mikhail.peresypkin

Роман Шеховцов
RVShehovtsov.SBT@sberbank.

Александр Абросимов
ASAbrosimov.SBT@sberbank.r

Наши публикации:

https://habrahabr.ru/company/efs/

http://www.highload.ru/2016/abstracts/2308.html

https://www.youtube.com/watch?v=_Hc0WkBUmcQ