

Руководство пользователя ПО «АРТ-Финтех»

Оглавление

| | |
|---|----|
| 1. Введение..... | 3 |
| 1.1 Описание функциональных характеристик ПО | 3 |
| 1.2 О системе | 3 |
| 1.3 Функциональные компоненты | 3 |
| 1.4 Описание функциональных компонент | 4 |
| 1.4.1 Универсальный банковский биллинг | 4 |
| 1.4.2 Пакеты и услуги | 6 |
| 1.4.3 Маркетплейс | 8 |
| 1.4.4 Оркестровщик микросервисов | 10 |
| 1.4.5 BPM-машина..... | 13 |
| 1.4.6 АРТ-лояльность | 15 |
| 1.4.7 Продуктовый каталог..... | 16 |
| 1.4.8 Система онлайн привлечения клиента (Sales Guru)..... | 19 |
| 1.4.9 Кредитный конвейер | 21 |
| 1.4.10 Конструктор отчетов..... | 22 |
| 1.4.11 Рабочее место операциониста..... | 23 |
| 1.4.12 Финансовый мониторинг..... | 25 |
| 1.4.13 Account processing | 27 |
| 1.4.14 Мобильный банк | 27 |
| 1.4.15 Интернет-банк | 30 |
| 7. Установка ПО | 30 |
| 7.1 Требования к системе | 30 |
| 7.2 Установка PostgreSQL 13 | 31 |
| 7.3 Установка инфраструктуры и сервисов среднего слоя, фронтального и мобильных приложений..... | 31 |
| 7.3.1 Конфигурация серверов..... | 31 |
| 7.3.2 Порядок установки..... | 31 |
| 7.3.3 Приложение для операционной системы «Android» | 57 |
| 7.3.4 Приложение для операционной системы «iOS» | 57 |

1. Введение

1.1 Описание функциональных характеристик ПО

Данное руководство представляет собой документацию, содержащую описание функциональных характеристик программного обеспечения «АРТ-Финтех» и информацию, необходимую для установки и эксплуатации программного обеспечения.

1.2 О системе

ПО «АРТ-Финтех» — комплекс программного обеспечения, включающий в себя набор функциональных компонент, предназначенных для решения разнообразных финансово-технических задач, как в комплексном взаимодействии друг с другом, так и в отдельном режиме.

Интерфейс АРТ-Финтех может быть настроен под потребности пользователей, функции разделены по правам доступа пользователей.

1.3 Функциональные компоненты

ПО состоит из различных функциональных компонент, которые могут устанавливаться и работать как вместе, так и отдельно друг от друга. Функциональные компоненты:

- **Универсальный банковский биллинг** — используется для настройки логики тарификации операций, процессов, транзакций и т.д.;
- **Пакеты и услуги** – компонента служит для объединения различных услуг, продуктов и сервисов финансовой организации в единый комплекс с целью их предложения клиентам и последующего сопровождения;
- **Маркетплейс** — представляет собой электронную платформу-супермаркет, на которой пользователи могут получать услуги различных организаций, банков, страховщиков так далее дистанционно, независимо от своего местонахождения и в формате 24/7. На передовой в этой системе находятся так называемые витрины, где представлены доступные для покупки продукты;
- **Оркестровщик микросервисов** – используется для управления микросервисами, их интеграции, маршрутизации различных процессов и маппинга данных;
- **ВРМ-машина** – в основе функционирования BPMN используется общепринятая нотация описания бизнес-процессов BPMN. На сегодняшний день используется версия нотации BPMN 2.0. Работа системы управления бизнес-процессами основана на моделировании бизнес-процессов организации с помощью наглядных схем (графических моделей).
- **АРТ-лояльность** - единая программа лояльности, действующая в рамках одной финансовой организации или национальной системы платежных карт «Мир» в целом.
- **Продуктовый каталог** – представляет собой единое хранилище продуктов без разделения продуктовой линейки по целевым системам. Компонента обеспечивает API для взаимодействия с целевыми системами, по которому целевые системы могут пополнять свои параметры продуктов и позволяет целевым системам добавлять новые параметры продуктов в каталог (при необходимости).
- **Система онлайн привлечения клиента (Sales Guru)** – пластичная экосистема для персонализации работы с клиентами;
- **Кредитный конвейер** – решение для оптимизации процесса кредитования для финансовых организаций;

- **Конструктор отчетов** – компонента, служащая для простого и легкого создания отчетов и их формирования в любом расширении (HTML, DOCX, PDF, XLSX и т.д.);
- **Рабочее место операциониста** – компонента, служащая для ведения основной работы с клиентами и договорами: заведение нового клиента, договора или операции, просмотр/изменение информации по клиенту, договору, операциям по договору и так далее.
- **Финансовый мониторинг** – компонента, позволяющая осуществлять контроль и надзор со стороны Банка за финансовыми операциями и денежными потоками клиентов.
- **Account processing** – компонента, предназначенная для расчета, обработки и передачи информации о текущем состоянии договоров, счетов и операций, без документарного оформления в режиме реального времени.
- **Мобильный банк** – решение, предназначенное для контроля клиентами Банка своих финансовых потоков.
- **Интернет-банк** – решение, позволяющее Банкам предоставлять своим клиентам дистанционное банковское обслуживание, а также доступ к счетам и операциям по ним.

Другие дополнительные функциональные элементы могут быть добавлены в ПО «АРТ-Финтех» с использованием комбинаций перечисленных компонент, а также в формате новых компонент.

Особенностью ПО «АРТ-Финтех» является гибкая настройка прав доступов пользователей. Каждая функциональная компонента имеет свои инструменты для настройки права доступа для пользователей к различным функциям компоненты, а также настройки рабочих меню, обеспечивающих доступ к данным и объектам, которые обрабатываются и сопровождаются в конкретной компоненте.

1.4 Описание функциональных компонент

В данном разделе приведены примеры реализованных решений на базе ПО «АРТ-Финтех».

1.4.1 Универсальный банковский биллинг

Задачей Универсального Банковского Биллинга является сведение настройки логики работы параметров тарифов к настройке с помощью low-code, что потребует минимального привлечения программистов. Основные работы по настройке можно будет поручить бизнес-администраторам либо другим ответственным сотрудникам.

Данная функциональная компонента имеет следующую структуру:

- Сборник тарифов – хранение значений тарифов. В сборнике можно выделять группы тарифов, в зависимости от бизнес-продукта, подразделения Банка и т.п.
- Конфигуратор тарифов/счетчиков – генератор кода для расчета тарифов, в результате его работы генерируется исполняемый java код.
- Исполняемый java код – сгенерированный код. Данный код может быть оформлен в виде отдельного микросервиса или в виде кода, который встраивается в существующий микросервис.

В сборнике тарифов ведутся пункты тарифных планов и значения параметров тарифов (рисунок 1.1).

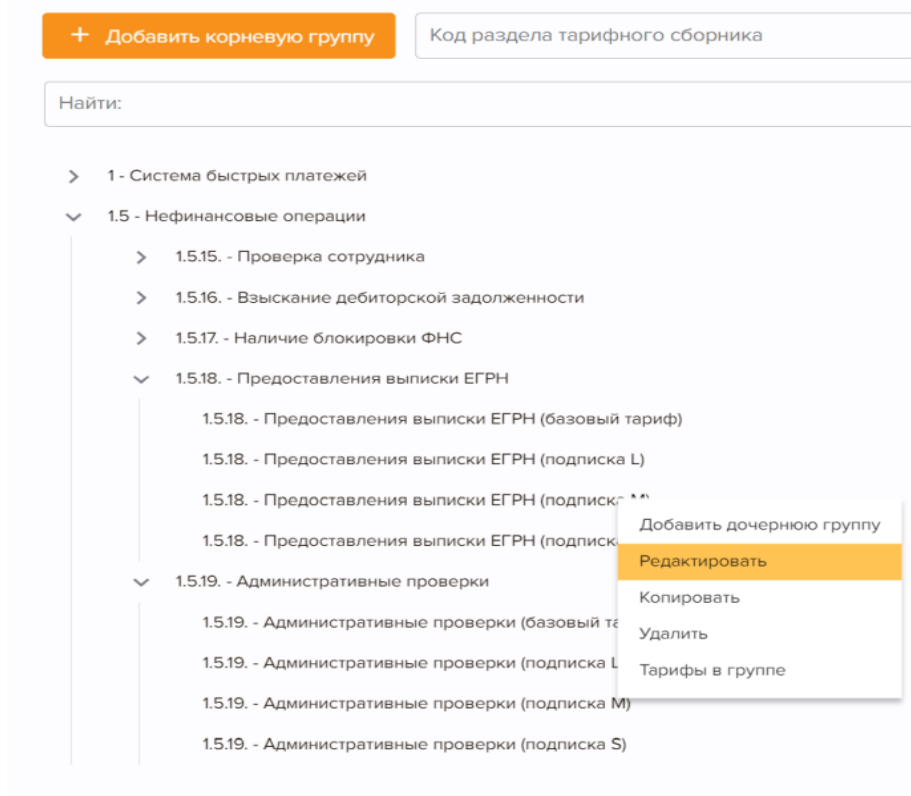


Рисунок 1.1 Группы тарифов

Сборник тарифов оптимизирован для использования бизнес-пользователями, не обладающими навыками программирования. При настройке тарифа пользователю отображается только определенный набор параметров для данной группы тарифов (рисунок 1.2).

| | | |
|--------------------------------|----------------------|-------------------|
| ID тарифа | Рейтинг | Статус тарифа * |
| 2200 | 40 | Действующий |
| Описание | | |
| комиссия 4.00% доп. 190.00 RUR | | |
| Комментарий | | |
| Параметры для условия | | |
| Мин. сумма операции | Макс. сумма операции | Валюта операции |
| 0.00 | 0.00 | RUR |
| Параметры для результата | | |
| Комиссия, % | Доп. сумма комиссии | Мин. сумма комис. |
| 4.00 | 190.00 | 0.00 |

Рисунок 1.2 Настройка тарифа

В процессе настройки параметров тарифов, в системе задаются шаблоны low-code для описания параметров, таким образом, пользователю не надо запоминать все возможные варианты. Шаблоны выбираются из соответствующего списка (рисунок 1.3.). В итоге, расчет тарифов осуществляется в соответствии с описанной логикой работы параметров тарифов,

которая настраивается с помощью low-code. Например, прописывается определение бюджетного платежа, принцип расчета суммы комиссии и т.п.

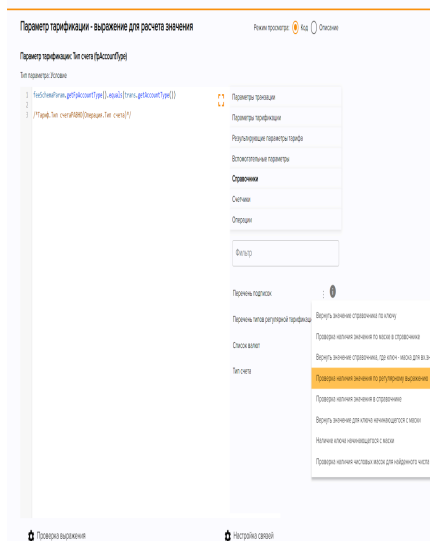


Рисунок 1.3 Настройка параметров тарифов

Поскольку система позволяет выбирать для каждой генерации кода шаблоны и группы тарифов, то она может являться поставщиком кода расчета тарифа для других систем Банка.

В зависимости от потребностей/конфигурации системы-потребителя можно поставлять как «исходный код», который будет встраиваться в нее, так и отдельный микросервис.

Также благодаря выбору групп тарифов при генерации кода, можно формировать поставки кода по расчету тарифов в зависимости от бизнес-области их применения (тарифы по карточным операциям, по расчетным счетам и т.п.).

1.4.2 Пакеты и услуги

Задачей компоненты «Пакеты и услуги» является создание гибкой логики настройки работы разнообразных процессов организаций, представляющих набор взаимосвязанных продуктов и сервисов. Настройка производится посредством low-code. Данная компонента может включать в себя настройки подписок, опций и т.д. и также включает в себя исполняемый код (рисунок 2.1).

| ID | Код | Наименование | Дата начала действия | Дата окончания действия |
|----|--------|--------------|----------------------|-------------------------|
| 17 | 1101.3 | L | 01.12.2021 | 01.01.2300 |
| 16 | 1101.2 | M | 01.12.2021 | 01.01.2300 |
| 15 | 1101.1 | S | 01.12.2021 | 01.01.2300 |
| 11 | 9999 | тест267 | 04.11.2021 | 14.12.2299 |
| 4 | 999999 | тест34 | 01.11.2021 | 01.01.2300 |

Рисунок 2.1 Перечень пакетов

Настройка параметров пакетов также производится посредством шаблонов low-code. Они описывают параметры настроек пакетов, таким образом, пользователю не надо запоминать все возможные варианты. Шаблоны выбираются из соответствующего списка (рисунок 2.2.). В итоге, связь продуктов, сервисов и процессов, взаимодействующих с ними, таких, к примеру, как периодичность взимания комиссий и их стоимость, автопродлонгации услуг, связанные бизнес-процессы и так далее, осуществляется в соответствии с описанной логикой работы параметров пакетов.

Редактирование пакета

| | |
|--|---|
| ИД | Код * |
| <input type="text" value="17"/> | <input type="text" value="1.10.1.3"/> |
| Наименование * | |
| <input type="text" value="L"/> | |
| Описание | |
| <input type="text" value="Подписка L"/> | |
| Дата начала действия | Дата окончания действия |
| <input type="text" value="01.12.2021"/> | <input type="text" value="01.01.2300"/> |
| Тип пакета * | Тип изменения пакета * |
| <input type="text"/> | <input type="text" value="Не настраиваемый"/> |
| Стоимость подключения пакета | Валюта |
| <input type="text" value="1 990.00"/> | <input type="text" value="RUR"/> |
| Статус * | Тип использования пакета |
| <input type="text" value="Действующий"/> | <input type="text" value="Альфа безопасность"/> |

Параметры пакета

| | |
|---|--------------------------------|
| Кол-во просроченных периодов для отключения | <input type="text" value="2"/> |
|---|--------------------------------|

Рисунок 2.1 Настройка параметров пакетов

В качестве дополнительного пула настроек пакетов в компоненты предусмотрены опции пакетов. Аналогично, функционал данного раздела включает в себя шаблоны, описывающие ряд настроек параметров пакетов в стиле low-code.

Редактирование опции

| | | |
|------------------------|--------------------------------|------------------------------------|
| ID | Код опции * | |
| 5 | 99911 | |
| Наименование опции * | | |
| тест11 | | |
| Описание опции | | |
| тест11 | | |
| Тип опции * | Тип использования опции | Статус |
| Услуга | Общего назначения | Действующая |
| Стоимость | Валюта | Рейтинг * |
| 4 000.00 | RUR | 10 |
| Дата начала действия * | Дата окончания действия * | Родительская опция |
| 02.12.2021 | 01.01.2300 | 9991 (тест1) |
| Тип отображения | Тип заполнения * | Тип заполнения дочерних объектов * |
| число | Можно не выбирать данный пункт | Нет дочерних веток |
| Параметры опции | | Нет данных для отображения |

Рисунок 2.1 Настройка параметров опции

1.4.3 Маркетплейс

Компонент имеет следующие возможности для настройки:

- Настраиваемые веб-интерфейсы системы и стилизация под брэндбук заказчика;
- Наличие мобильных приложений iOS/Android как для продавцов, так и для покупателей;
- Законченные и кастомизируемые бизнес-процессы ведения сделок средствами BPM-нотации;
- Настройка способов оплаты товаров;
- Возможность проведения взаиморасчетов между участниками.
- Автоматизированный складской учет;
- Возможность создания акций и рекламных кампаний;
- Личные кабинеты поставщиков(торговцев) товаров и услуг;
- Личные кабинеты покупателей;
- Личные кабинеты сотрудников (администраторы, бухгалтеры, курьеры, экспедиторы и пр.);
- Ролевая и рейтинговые модели пользователей системы и связей между ними;
- Интеграционные возможности для взаимодействия с внешними сервисами (государственные, финансовые и др.).

Компонента имеет возможность (рисунок 3.1):

- изменения и настройки внешнего вида интерфейсов ПО;
- свободной каталогизации товаров и услуг с неограниченной вложенностью.

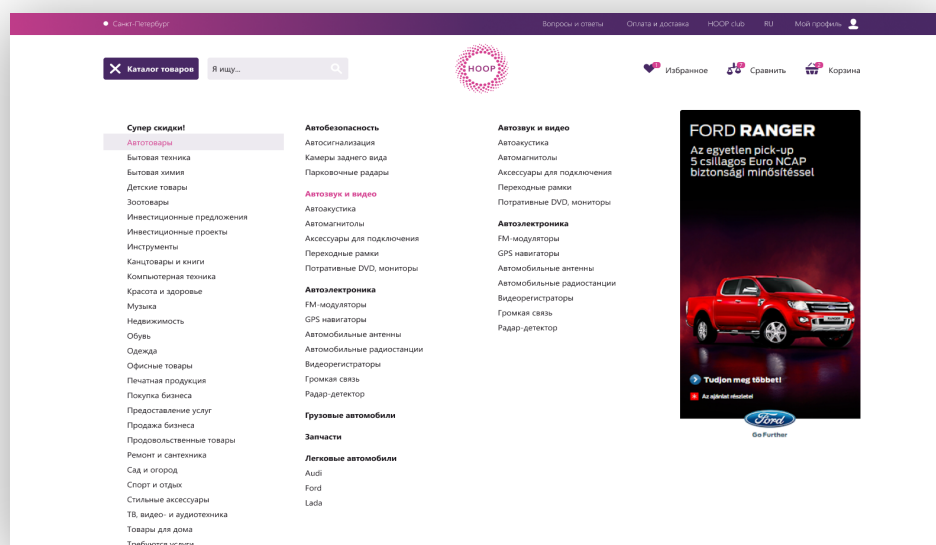


Рисунок 3.1 Компонента «Маркетплейс»

Компонента маркетплейса имеет следующие преимущества (рисунок 3.2):

- Наличие общего мобильного приложения для покупателей продукции. С корзиной, онлайн-оплатой, отслеживанием состояния заказов.
- Приложения для участников платформы с разделением по ролям в рамках организации.

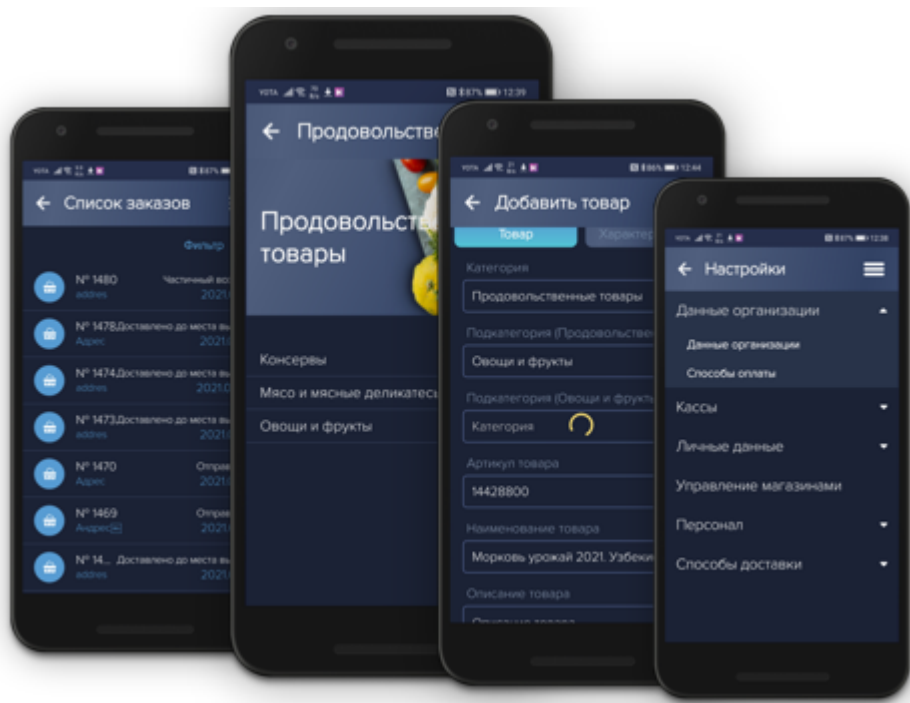


Рисунок 3.2 Мобильные приложения «Маркетплейс»

В Маркетплейсе реализованы следующие особенности взаимозачетов (рисунок 3.3):

- ПО имеет мощный бэк-офис с возможностью ведения бухгалтерского учета сделок и последующей выгрузки информации по сделкам и фин. позициям участников.
- Возможность создания акций на те или иные позиции в каталоге с контролем их продолжительности
- Связывание товаров и услуг как в рамках акций, так и по умолчанию.

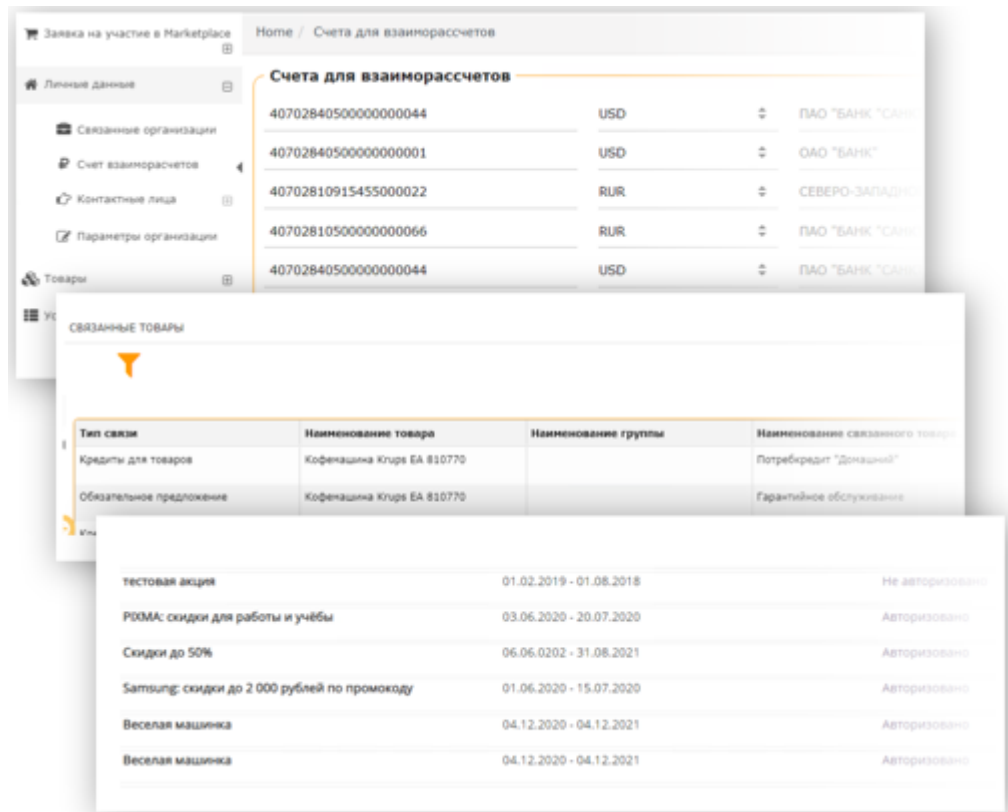


Рисунок 3.3 Интерфейсы отображения системы взаиморасчетов

1.4.4 Оркестровщик микросервисов

Функционал этой компоненты позволяет решать следующие задачи:

- Выполнять оркестровку микросервисов любой сложности;
- Осуществлять полное документирование микросервисов, маппинга, роутинга;
- Настраивать интеграцию микросервисов в концепции low-code в нотации BPMN;
- Компилировать настроенные правила интеграции в исполняемый код в микросервисной парадигме;
- Настраивать маппинг данных в концепции low-code с формированием кода в микросервисной парадигме;
- Создавать пользовательские интерфейсы в оркестровщике (генерация на основе источника данных);
- Настраивать маппинг.

В Оркестровщике осуществляется ведение репозитория микросервисов, команд микросервисов, роутов, endpoint, конфигураций, стеков и других сущностей, и элементов системы с поддержкой их версииности (рисунок 4.1).

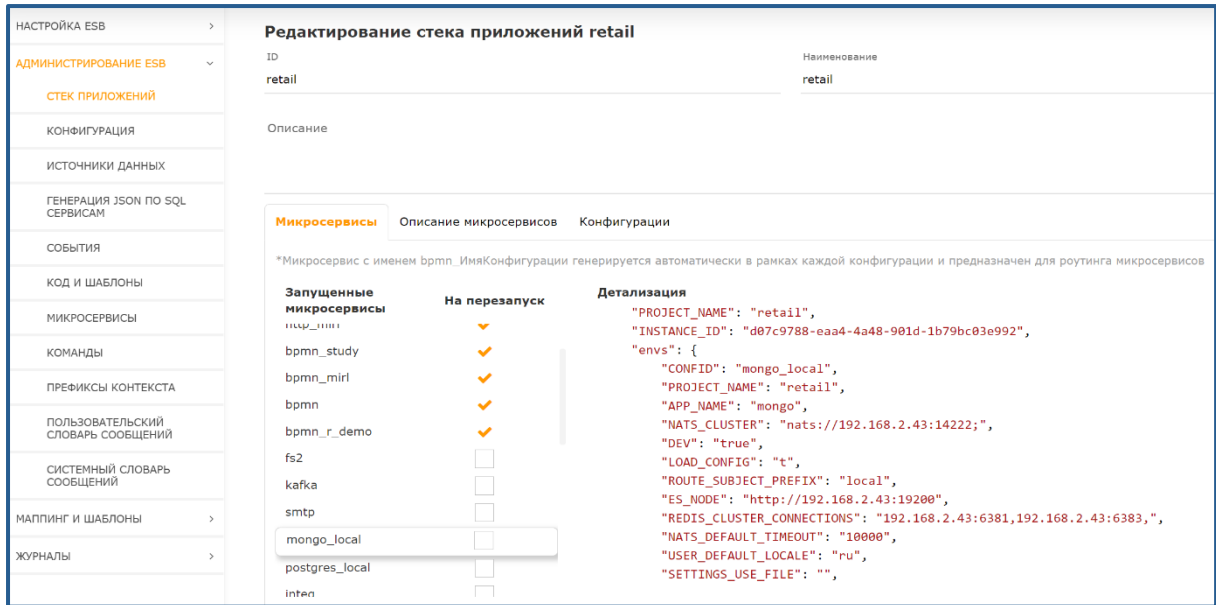


Рисунок 4.1 Репозиторий оркестровщика

В системе можно описывать как структуру данных пришедших на вход endpoint, так и внутренний контекст данных BPMN. Имеется возможность использования эталонной модели данных (рисунок 4.2).

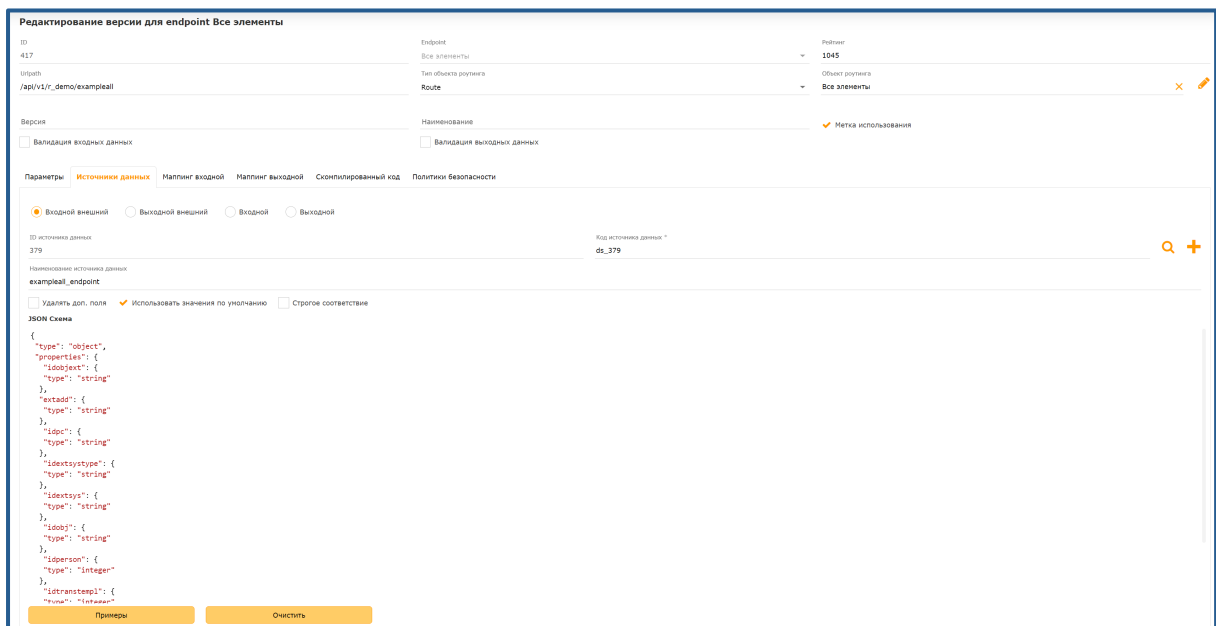


Рисунок 4.2 Модели данных

С помощью Оркестровщика настраиваются сложные взаимодействия множества систем с различными наборами входных и выходных данных и форматами их источников. Также осуществляется реализация параллельных процессов, прерывание, ожидание сигналов от внешних систем, таймеры и т.д. без написания кода (рисунок 4.3).

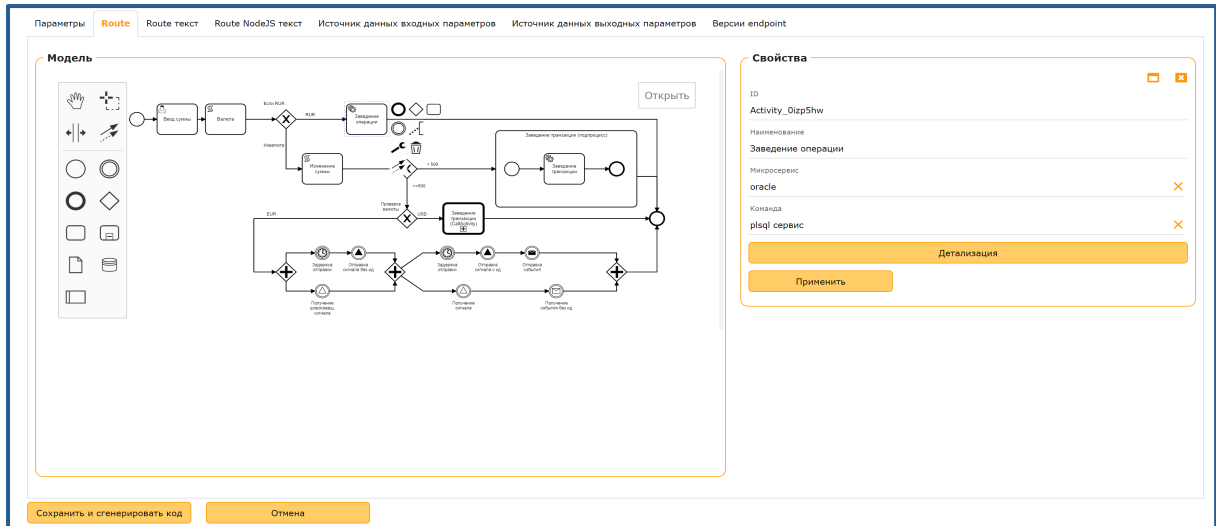


Рисунок 4.3 Визуальное построение интеграционных процессов

Настройка элементов роута осуществляется в стиле low-code с подсказками как по структуре команды микросервиса, так и по структуре внутреннего контекста BPMN (рисунок 4.4.).

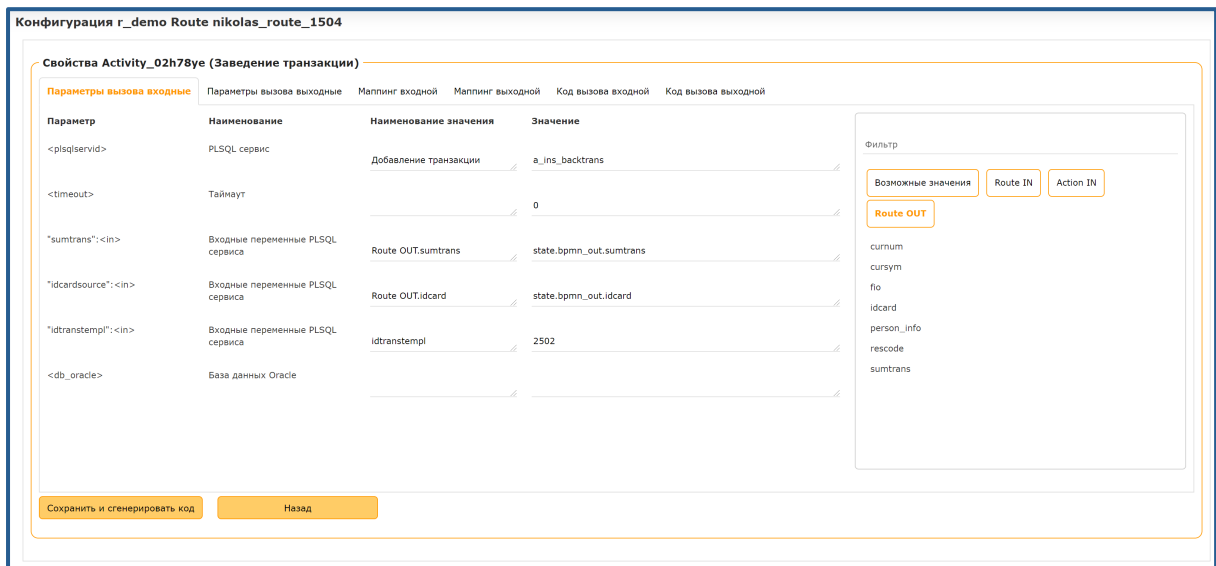


Рисунок 4.4 Настройка параметров вызова команды микросервиса

Результатом всех настроек системы является сгенерированный нативный код той среды, на которой будет выполняться решение (рисунок 4.5).

Конфигурация r_demo Route nikolas_route_1504

Параметры Route Route текст **Route NodeJS текст** Источник данных входных параметров Источник данных выходных параметров Версии endpoint

```

const xmlHash = '67cc52ca746c7c9cd44e2238c12a3820bd609bbfdd70497f3957dd3182d95c20';
const compilerVersion = '0dfff4f3bef4ce65232970519ef59a3e8a4de6ffbd35474afa4c3d6f9dd0c2a0';
const $artms = require('artms-core');
let artms = require('artms-core');
const { ErrorFactoryBase } = require('artms-core/commands/CommandError');
const $App = require('../App');
let App = require('../App');
const bpmnstate = require('../bpmnState')
const lodash = require('lodash');
const { Updater } = require('../mongo');
let routeFileName = require('path').basename(__filename).replace(/.js$/, '');

// -- imports --
let mapOutUsertask_Activity_0q1y1pv;
try { mapOutUsertask_Activity_0q1y1pv = require(`./${routeFileName}_Activity_0q1y1pv.map.out.js`) } catch (e) { mapOutUsertask_Activity_0q1y1pv = (state, result) };

let mapOutCallActivity_Activity_1k0ksat;
try { mapOutCallActivity_Activity_1k0ksat = require(`./${routeFileName}_Activity_1k0ksat.map.out.js`) } catch (e) {
  mapOutCallActivity_Activity_1k0ksat = (state, subState) => {
    if (subState.error)
      throw subState.error;
    state.bpmn_out = subState.bpmn_out;
    state.bpmn_in = subState.bpmn_in;
  }
};
let mapInCallActivity_Activity_1k0ksat;
try { mapInCallActivity_Activity_1k0ksat = require(`./${routeFileName}_Activity_1k0ksat.map.in.js`) } catch (e) {
  mapInCallActivity_Activity_1k0ksat = (state, subState) => {

```

Рисунок 4.5 Сгенерированный код

1.4.5 BPM-машина

Функциональная компонента реализует концепцию BPMN, что позволяет строить гибкие адаптивные информационные потоки, способные оперативно меняться вместе с изменением бизнес-процессов компании. Благодаря использованию BPMN можно навести порядок в бизнес-процессах организации, сделать их выполнение четким и формальным (рисунок 5.1).

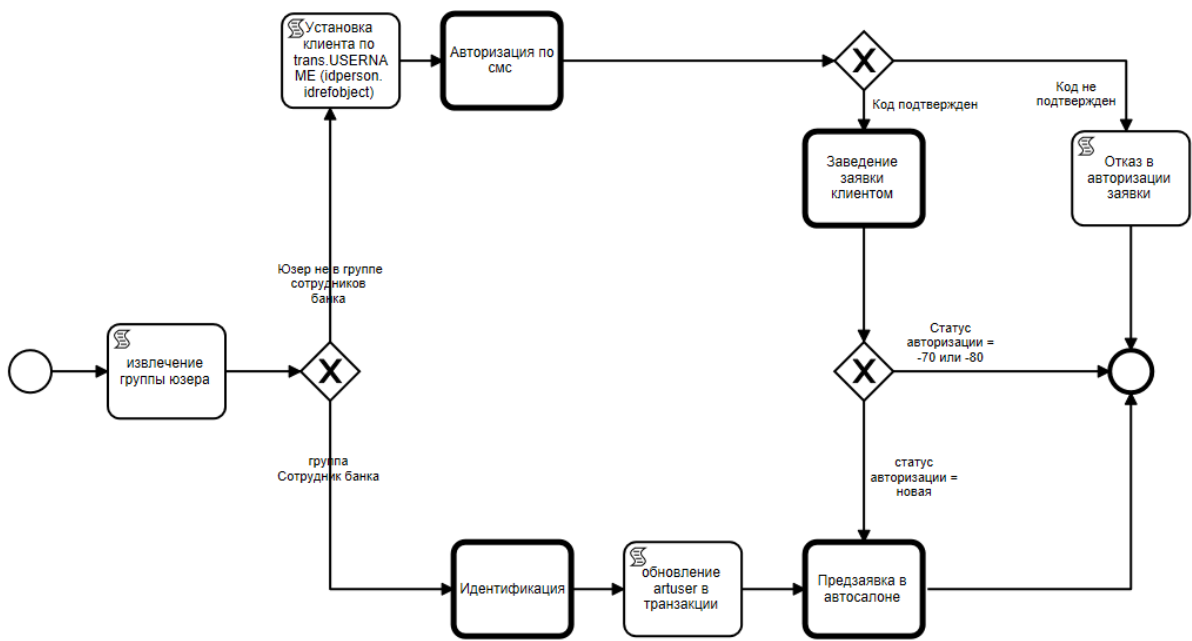


Рисунок 5.1 BPMN-схема

Проектирование бизнес-процессов может выполнять бизнес-аналитик без участия программиста. В BPMN-машине процессы описываются на языке схем (bpmn), привычном для аналитика и понятном руководству организации.

Схема (графическая модель) бизнес-процесса строится путем перетаскивания мышью нужных элементов с палитры в область данных. Элементы схемы соединяются потоками (Sequence Flow). Список используемых элементов можно найти в справочнике BPMN-элементов.

При сохранении схемы формируется XML-файл заданного формата, в котором передаются его уникальный номер, наименование, атрибуты и пр.

После сохранения схемы, её необходимо скомпилировать и определить использование (рисунок 5.2).

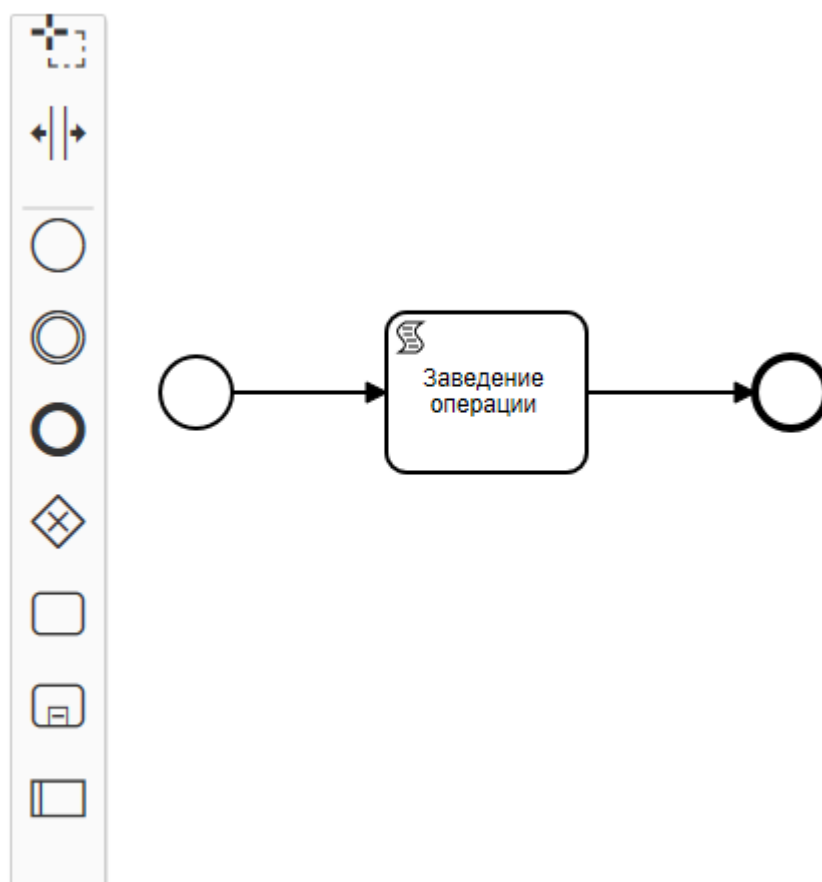


Рисунок 5.2 Создание и редактирование BPMN-схемы

У одной BPMN-модели может быть несколько версий, но только одна версия может использоваться. Это позволяет выполнить компиляцию новой или отредактированной версии без остановки выполнения BPMN-процессов. Если BPMN-процесс начал выполнение по одной версии BPMN-модели, то он не прекратит свое выполнение при назначении использования в это время другой версии BPMN-модели (рисунок 5.3).

| ID | Описание | Статус компиляции | Дата начала | Дата завершения | Дата запуска | Используется |
|------|--|-------------------|-------------|-----------------|--------------|-------------------------------------|
| 8500 | Кредитный конвейер cgm v3 + Андеррайтинг (...) | Скомпилирована | 03.11.2020 | | | <input checked="" type="checkbox"/> |
| 8421 | Кредитный конвейер cgm v3 + Андеррайтинг (...) | Скомпилирована | | | | <input type="checkbox"/> |
| 8380 | Кредитный конвейер cgm v3 + Андеррайтинг(т... | Новая | | | | <input type="checkbox"/> |
| 8360 | Кредитный конвейер cgm v3 + Андеррайтинг | Скомпилирована | 14.12.2018 | 03.11.2020 | | <input type="checkbox"/> |
| 8340 | Кредитный конвейер cgm v2 + Андеррайтинг | Скомпилирована | 14.12.2018 | 14.12.2018 | | <input type="checkbox"/> |
| 8200 | Кредитный конвейер cgm v2 (создана на осно... | Скомпилирована | 14.12.2018 | 14.12.2018 | | <input type="checkbox"/> |
| 8180 | Кредитный конвейер cgm v2 (создана на осно... | Скомпилирована | 14.12.2018 | 14.12.2018 | | <input type="checkbox"/> |
| 8160 | Кредитный конвейер cgm v2 | Скомпилирована | 14.12.2018 | 14.12.2018 | | <input type="checkbox"/> |
| 8140 | Кредитный конвейер cgm v1 (создана на осно... | Скомпилирована | 01.11.2018 | 01.11.2018 | | <input type="checkbox"/> |
| 8120 | Кредитный конвейер cgm v1 (создана на осно... | Скомпилирована | 01.11.2018 | 01.11.2018 | | <input type="checkbox"/> |
| 8100 | Кредитный конвейер cgm v1 | Скомпилирована | 01.11.2018 | | | <input type="checkbox"/> |

Рисунок 5.3 Версионность BPMN-моделей

1.4.6 АРТ-лояльность

Данная компонента позволяет объединить все различные программы лояльности/купонов/скидок множества торговых организаций и брендов в единую систему, доступную для всех компаний и потребителей. Пример схемы бизнес-процессов, задействованных в реализации компоненты, указана на рисунке 6.1.

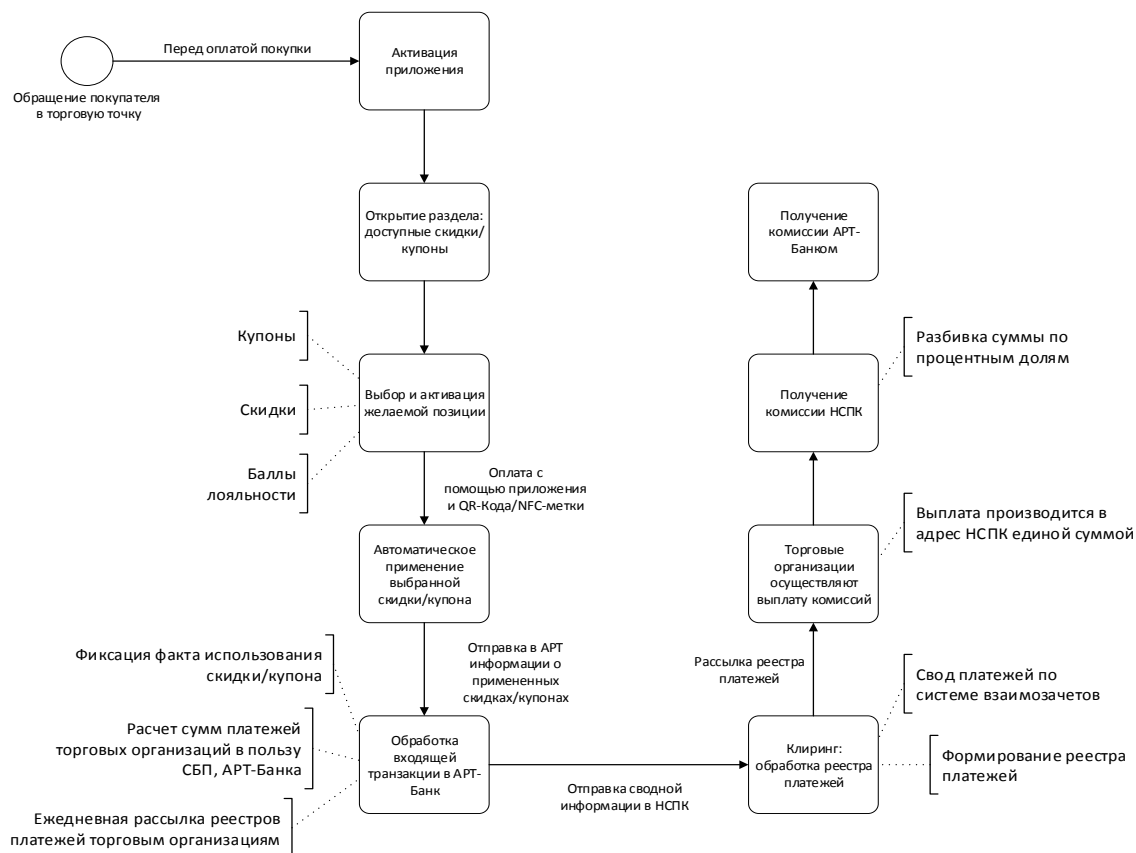


Рисунок 6.1 Схема бизнес-процессов АРТ-лояльность

Интеграция ART-Лояльность может строиться как в режиме ONLINE, так и в режиме OFFLINE, в зависимости от требований организации. Для реализации ONLINE-взаимодействия используется API и механизм очередей.

Для реализации схем лояльности из мастер-системы будут выгружаться данные по клиентам, от полноты данных зависит, насколько широкие возможности будут доступны для идентификации клиента, и таргетированный подход к его обслуживанию. Также из мастер-системы будут выгружаться все продукты, для которых будут применяться схемы лояльности и строиться рекламные кампании.

Основой всех расчетов служит информация по всем транзакциям клиента. Самым оптимальным вариантом является выгрузка всех транзакций из процессингового центра по картам клиентов непосредственно в АБС и в ART-Лояльность (рисунок 6.2).

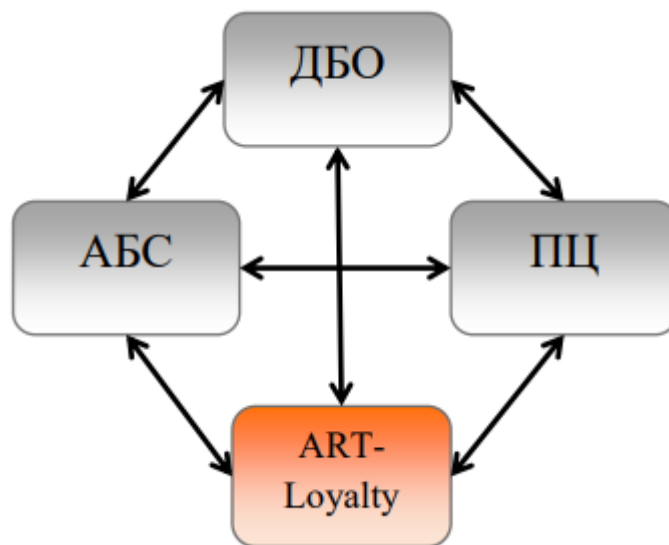


Рисунок 6.2 Схема интеграции ART-лояльность

1.4.7 Продуктовый каталог

Данная функциональная компонента включает в себя такие разделы, как:

- дерево бизнес-настроек правил работы продуктов (наличие грейс-периода, его длительность, условия подключения, пролонгация депозитов (с переводом на другой продукт или с сохранением текущего продукта) и т.д.);
- ведение тарифного сборника (ведение всех параметров и значений для корректного расчета тарифов, то есть весь продуктовый сборник банка);
- ведение правил расчета курсов конвертации (ведение типов и правил ведения курсов конвертации);
- ведение правил начисления процентов (виды процентных ставок, расчеты графиков погашения, их типы и т.д.);

Дерево бизнес-настроек представляет собой набор параметров из определенных справочников, к которым обращаются внешние системы. С помощью данных настроек, пользователь имеет возможность сопровождать все продукты Банка, такие как банковские карты, депозиты, различные валютные и кассовые продукты, все виды кредитования, банковские ячейки и так далее.

Эти параметры можно принимать через API или настраивать и модифицировать посредством интерфейса (рисунок 7.1).

| Наименование | Мнемокод | Тип | Значение по умолчанию | Значение текущее | Дата с ^ |
|---|-------------------------------|---------------------|-------------------------------|------------------|----------|
| Условия кредитного договора | CD | Метка использования | Включено | | 01.01.19 |
| Вести ЗКП | CD_ZKP | Метка использования | Включено | | 01.01.19 |
| Вести ЗКП1 | CD_ZKP1 | Метка использования | Включено | | 01.01.19 |
| Шаблон транзакции 1 для формирования ЗКП1 | CD_ZKP1_TRANSTEMPL1 | ID шаблона операции | Фиксация ЗКП1 (транзакция 1) | | 01.01.19 |
| Сумма остатков по семантике - Транзакции 1 для формиро... | CD_ZKP1_SEMANTIC1 | ID семантики | CREDIT_POSITION_1 - Судны... | | 01.01.19 |
| Биллинговая дата | CD_ZKP_BILLINGDATE | Метка использования | Включено | | 01.01.19 |
| На дату | CD_ZKP_BILLINGDATE_ONDATE | ID расписания | Последний рабочий день ме... | | 01.01.19 |
| Хранение биллинговой даты | CD_ZKP_SAVEMARK_BILLINGDATE | Метка использования | Включено | | 01.01.19 |
| Ведение льготного периода | CD_GRACE | Метка использования | Включено | | 01.01.19 |
| Биллинговая дата для льготного периода | CD_GRACE_BILLINGDATE | Метка использования | Включено | | 01.01.19 |
| На дату | CD_GRACE_BILLINGDATE_ONDATE | ID расписания | Последний рабочий день ме... | | 01.01.19 |
| Проверка условий ведения льготного периода | CD_GRACE_CHECK | Метка использования | Включено | | 01.01.19 |
| Продолжительность льготного периода | CD_GRACE_PERIOD | Метка использования | Включено | | 01.01.19 |
| На дату | CD_GRACE_PERIOD_ONDATE | ID расписания | Последний рабочий день ме... | | 01.01.19 |
| Сумма для проверки условий льготного кредитования | CD_GRACE_CHECK_SUM | Метка использования | Включено | | 01.01.19 |
| Нулевой остаток по сумме на конец дня (по семантике) | CD_GRACE_CHECK_SUM_SEMANTIC | ID семантики | CREDIT_POSITION_23 - Судны... | | 01.01.19 |
| Вести ЗКП льготного периода | CD_ZKP3 | Метка использования | Включено | | 01.01.19 |
| Шаблон транзакции 1 для формирования ЗКП льготного пер... | CD_ZKP3_TRANSTEMPL1 | ID шаблона операции | Фиксация ЗКП ГРЭЙС (транз... | | 01.01.19 |
| Сумма остатков по семантике - Транзакции 1 для формиро... | CD_ZKP3_SEMANTIC1 | ID семантики | CREDIT_POSITION_1 - Судны... | | 01.01.19 |
| Хранение биллинговой даты | CD_GRACE_SAVEMARK_BILLINGDATE | Метка использования | Включено | | 01.01.19 |
| Действия в случае невыполнения условий льготного периода | CD_GRACE_BREAK | Метка использования | Включено | | 01.01.19 |
| Вынос на просрочку | CD_DELAYINFO | Метка использования | Включено | | 01.01.19 |
| Вынос на просрочку 1 | CD_DELAY | Метка использования | Включено | | 01.01.19 |
| Судный день | CD_DELAY_LOANDATE | Метка использования | Включено | | 01.01.19 |
| На дату | CD_DELAY_LOANDATE_ONDATE | ID расписания | Последний рабочий день ме... | | 01.01.19 |

Рисунок 7.1 Дерево бизнес-настроек

Настройка продукта представляет собой указание ряда параметров, правил, условий и шаблонов (рисунок 7.1)

| Модель интерфейсов | | | Тип | Порядок |
|------------------------|---|--------------------|-----|---------|
| Потребительский кредит | ▼ | для заведения | ▼ | |
| Потребительский кредит | ▼ | для редактирования | ▼ | |

Рисунок 7.2 Параметры продукта

Компонента «Касса» предназначена для выполнения кассиром различных операций (рисунок 7.3), таких как заведение, редактирование или авторизация операций, открытие, выбор или печать реестра, выполни служебных операций в обменном пункте, а также просмотр отчетов по операциям по приему на инкассо и экспертизу денежных знаков.

Реестр | Служебная

Фильтр

Дата [] по []

Операция []

- Обработанные - Новые Поиск

| ID | Дата / Время | Отчет | Комментарий | Сумма | Валюта | |
|---------|---------------------|---|----------------------------|-----------|--------|----------------|
| | | Журнал регистрации курсов валют | | | | |
| | | Текст для дела | | | | |
| 3744377 | 01.12.2016 22:35:27 | | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3744377 | 01.12.2016 22:35:27 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3744377 | 01.12.2016 22:35:27 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |

Количество 94

| Дата/Время | Операция | Сумма | Валюта | Оператор | Комментарий |
|------------|----------|-------|--------|----------|-------------|
| | | | | | |

Обновить Просмотр Добавить... Удалить Авторизовать Выход

Курсы валют Печать Изменить...

Рисунок 7.3 Кассовая компонента

На вкладке «Служебная» (рисунок 7.4) осуществляются все служебные операции в обменном пункте, а также здесь просматриваем отчеты по операциям по приему на инкассо и экспертизу денежных знаков.

Реестр | Служебная

Фильтр

Дата 01.12.2016 по 01.12.2016

Операция []

- Обработанные - Новые Поиск

| ID | Дата / Время | Отчет | Комментарий | Сумма | Валюта | |
|---------|---------------------|---|----------------------------|-----------|--------|----------------|
| | | Журнал регистрации курсов валют | | | | |
| | | Текст для дела | | | | |
| 3744377 | 01.12.2016 22:35:27 | | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3744377 | 01.12.2016 22:35:27 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3744377 | 01.12.2016 22:35:27 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 50 000.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |
| 3962578 | 01.12.2016 20:09:28 | Приходный кассовый ордер (0402008) (опе | Пополнение счета наличными | 1 150.00 | RUR | Пополнение сче |

Количество 94

| Дата/Время | Операция | Сумма | Валюта | Оператор | Комментарий |
|------------|----------|-------|--------|----------|-------------|
| | | | | | |

Обновить Просмотр Добавить... Удалить Авторизовать Выход

Курсы валют Печать Изменить...

Рисунок 7.4 Вкладка «Служебная» кассовой компоненты

Ячейки – компонента ПО, позволяющая оперировать с сейфами в банке, сдаваемыми клиентам в наём для хранения ценностей (денег, ценных бумаг, вещей и документов) с заключением договора аренды (рисунок 7.5).

Ячейки (клиент: неизвестный)

Тип ячейки: Фамилия: Дата начала с: 00.00.0000 по: 00.00.0000

Статус ячейки: Имя: Дата окончания с: 00.00.0000 по: 00.00.0000

Договор аренды: Отчество:

| Номер ячейки | Тип | Статус | Номер договора | Дата с | Дата по | Арендатор | Статус договора | ID | ID Договора аренды | ID Клиента |
|--------------|----------------|----------|----------------|--------|---------|-----------|-----------------|----|--------------------|------------|
| 3 | A(150x250X300) | Занята | | | | | | 37 | | |
| 39 | A(150x250X300) | Занята | | | | | | 39 | | |
| 41 | A(150x250X300) | Занята | | | | | | 41 | | |
| 43 | A(150x250X300) | Занята | | | | | | 43 | | |
| 45 | A(150x250X300) | Свободна | | | | | | 45 | | |
| 47 | A(150x250X300) | Свободна | | | | | | 47 | | |
| 49 | A(150x250X300) | Свободна | | | | | | 49 | | |
| 51 | A(150x250X300) | Свободна | | | | | | 51 | | |
| 53 | A(150x250X300) | Свободна | | | | | | 53 | | |
| 55 | A(150x250X300) | Свободна | | | | | | 55 | | |
| 57 | A(150x250X300) | Свободна | | | | | | 57 | | |
| 59 | A(150x250X300) | Свободна | | | | | | 59 | | |
| 61 | A(150x250X300) | Занята | | | | | | 61 | | |
| 63 | A(150x250X300) | Свободна | | | | | | 63 | | |
| 65 | A(150x250X300) | Свободна | | | | | | 65 | | |
| 67 | A(150x250X300) | Свободна | | | | | | 67 | | |
| 69 | A(150x250X300) | Свободна | | | | | | 69 | | |
| 71 | A(150x250X300) | Свободна | | | | | | 71 | | |
| 73 | A(30x250X300) | Занята | | | | | | 73 | | |
| 75 | A(30x250X300) | Свободна | | | | | | 75 | | |
| 77 | A(30x250X300) | Занята | | | | | | 77 | | |
| 79 | A(30x250X300) | Свободна | | | | | | 79 | | |
| 81 | A(30x250X300) | Свободна | | | | | | 81 | | |
| 83 | A(30x250X300) | Занята | | | | | | 83 | | |
| 85 | A(30x250X300) | Свободна | | | | | | 85 | | |

Рисунок 7.5 Ячейки

1.4.8 Система онлайн привлечения клиента (Sales Guru)

Данная функциональная компонента используется для:

- Автоматизации взаимодействия с каждым клиентом на сайте организации-пользователя;
- Сбора, фиксации и анализа ключевых показателей работы с клиентом (его запросы, интересы, пожелания, прогноз дохода);
- Фиксации и создания для клиента новых тарифов, пакетов продуктов, услуг, сервисов.

Поиск клиента

История торговли

Морозов Олег Менеджер

Карточка клиента

ЦПР 131
 Уникальный ID 1220
 ПМ **МОРОЗОВ ОЛЕГ ИВАНОВИЧ**

Пакеты клиента Льготные тарифы

Удачный выбор ▼

Тарифы:

Рисунок 8.1 Карточка клиента

| Торговаться | Объект торговли | История |
|-------------|--|--|
| | Внутрибанковские переводы в валюте РФ на счета физ.лиц | → |
| | Индивидуальный тариф | Обещания Перевод дочерних компаний из других банков Количество переведенных дочерних компаний в банк в мес. (Не проверено) |
| | Выдача наличных денежных средств в валюте РФ на прочие цели | → |
| | Внутрибанковские переводы в валюте РФ со счета ИП на собственный счет физ.лиц | |
| | Стандартный тариф Внутрибанковские переводы в валюте РФ со счета ИП не более 300000 RUR с комиссией 0% от 300000 до 1000000 RUR включительно с комиссией 1.25% свыше 1000000 RUR с комиссией 2.25% | |

Рисунок 8.2 Детализация карточки клиента

Морозов Олег
Менеджер

Выйти

| ID | ЦПР | Объект торговли | Дата начала | Дата окончания | Статус торговли | Менеджер | Отчет |
|------|-----|--|-------------|----------------|-----------------|---------------|-------|
| 1007 | | Валютные переводы | 29.04.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |
| 1220 | 131 | Переводы на счета ФЛ | 13.10.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |
| 1220 | 131 | Переводы ИП на собственный счет ФЛ | 29.04.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |
| 1220 | 131 | Выдача наличных через банкомат | 17.12.2021 | 19.04.2023 | Авторизовано | МОРОЗОВ О. И. | |
| 1220 | 131 | Валютный контроль (договор не требуется принимать на учёт, п4.3) | 13.10.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |
| 1822 | 430 | Валютный контроль (договор принят на учёт, п4.1) | 29.04.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |
| 2809 | 563 | Конвертация | 29.04.2021 | 19.04.2022 | Авторизовано | МОРОЗОВ О. И. | |

Рисунок 8.3 Журнал истории торговли

Смирнов Константин
Менеджер

Выйти

| ID | ЦПР | Пакет | Набор объектов торговли | Дата заведения | Менеджер | Действия |
|------|-----|-----------------|------------------------------------|----------------|---------------|----------|
| 1220 | 131 | 2 Удачный выбор | Внесение наличных через банкомат | 19.04.2021 | МОРОЗОВ О. И. | |
| 2440 | 421 | 2 Удачный выбор | Переводы ЮЛ на счета ЮЛ/ИП | 19.04.2021 | ВОЛКОВА К. А. | |
| 2440 | 421 | 2 Удачный выбор | Конвертация | 19.04.2021 | ВОЛКОВА К. А. | |
| 3903 | | 1 Все, что надо | Переводы ИП на собственный счет ФЛ | 19.04.2021 | ВОЛКОВА К. А. | |
| 2809 | 563 | 2 Удачный выбор | Внесение наличных через банкомат | 19.04.2021 | МОРОЗОВ О. И. | |
| 1241 | 302 | 2 Удачный выбор | Переводы ИП на собственный счет ФЛ | 19.04.2021 | ВОЛКОВА К. А. | |

Рисунок 8.4 Журнал подтверждений

1.4.9 Кредитный конвейер

Данная компонента разработана для выполнения следующих задач:

- Подбор оптимальной программы кредитования;
- Работа с партнерами в единой точке;
- Контроль и быстрота прохождения заявки на всех этапах;
- Первичную верификацию и надежный скоринг;
- Принятие решения автоматически по преднастроенным правилам либо уполномоченными сотрудниками;
- Формирование пакета документов;
- Выгрузка данных в АБС Банка;
- Фиксация действий пользователей в системе.

Карточка клиента и предзаявка могут быть как заполнены клиентом самостоятельно в интернете, так и специалистами организации непосредственно при обращении в автосалон, магазин, партнерскую точку, банк и так далее (рисунок 9.1).

The screenshot displays a client card for 'Калица Сергей Петрович'. It is divided into several sections:

- Персональные данные:** Includes phone number (124566), gender (Male), date of birth (1928-07-24), marital status (Married), and company information (ООО 'ПРОДУКТОВЫЙ ДОМ').
- Связанные лица:** Lists family members like 'ЛАШИН В.И.' and 'РОГОВ В.И.' with their roles.
- Группы:** Lists credit programs like 'СБКИ/СБКИ-С VISA GOLD Банк КБ'.
- Данные о контактах с клиентом:** A table with columns for ID, date, name, type, status, and initiator.
- Детализация кредитов:** A table showing credit products like 'VISA GOLD RUR 2 года' and 'Mastercard Standard RUR' with their terms and amounts.
- Персональные предложения:** A section for targeted offers.
- Действия:** A list of actions like 'Адрес клиента' and 'Выгрузка с клиентом в Банк'.
- Контакты:** Mobile phone numbers and an email address.

Рисунок 9.1 Карточка клиента

Более подробные данные по заявке отображаются в окне детализации заявки (рисунок 9.2).

The screenshot shows a detailed view of a credit application. It includes:

- Клиентская информация:** Fields for name (КАЛИЦА СЕРГЕЙ ПЕТРОВИЧ), phone, date of birth, gender, and address.
- Стратегия оценки: Стратегия оценки рисков по кредитной заявке:** A table with columns for 'Имя', 'Результат', 'Примечание', 'Имя пользователя', 'Дата', 'Статус', 'Действие', and 'Заявка'.

| Имя | Результат | Примечание | Имя пользователя | Дата | Статус | Действие | Заявка |
|---|-----------|------------|------------------|------|--------|-------------|--------|
| Идентификация | ● | | | | ● | Детализация | Заявка |
| Клиент не в базисной | ● | | | | ● | Детализация | Заявка |
| Клиент не в группе 1 или 2 группы | ● | | | | ● | Детализация | Заявка |
| Сумма предложения | ● | | | | ● | Детализация | Заявка |
| Отсутствует кредитная история | ● | | | | ● | Детализация | Заявка |
| Клиент кредитован в Банке | ● | | | | ● | Детализация | Заявка |
| Платежи по кред. | ● | | | | ● | Детализация | Заявка |
| Нет положительной кредитной истории в Банке | ● | | | | ● | Детализация | Заявка |
| Нет в черном списке | ● | | | | ● | Детализация | Заявка |
| Нет отрицательных результатов проверки кредитоспособности | ● | | | | ● | Детализация | Заявка |

Рисунок 9.2 Детализация заявки

Заведенные на рассмотрение кредитные заявки отображаются в журнале Кредитных заявок (рисунок 9.3).

| Лист ожидания | | | | | | | | | |
|---------------|-----------------|----------------|-------------------|---------------|--------------|-----|------------|---------------------------|---------------|
| Тип заявки | Номер заявления | Фазы решения | Фамилия заявителя | Имя заявителя | Отчество | РНИ | ИНН | Кредитный тариф | Сумма кредита |
| Кредитование | 18812165 | Анкета клиента | Калица | Сергей | Петрович | | | 1.5-10кк | |
| Кредитование | 18813165 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18815165 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18816167 | Анкета клиента | Защайников | Виктор | Спирidonович | | 6548974512 | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18818173 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18818174 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18818175 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18818176 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5000000 |
| Кредитование | 18821165 | Анкета клиента | Калица | Сергей | Петрович | | | Кредит наличными 1.5-10кк | 5700000 |

Рисунок 9.3 Журнал кредитных заявок

Одним из главных преимуществ АРТ-кредитный конвейер является факт, что логика его работы построена на принципах BPM-моделирования (рисунок 9.4).

Это позволяет оперативно настраивать формы ввода и отображения информации, состав параметров кредитной заявки, процесс ее обработки, стратегию принятия решения по предоставлению кредита. Для каждого кредитного продукта может быть настроен индивидуальный состав параметров кредитной заявки и индивидуальная стратегия рассмотрения заявки и принятия решения.

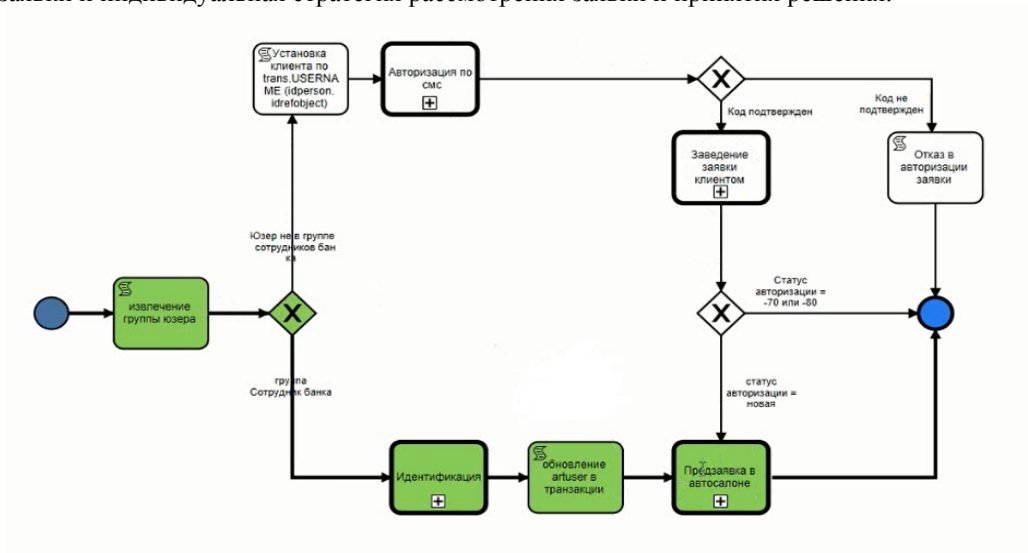


Рисунок 9.4 BPM-модель

1.4.10 Конструктор отчетов

Компонента позволяет создавать шаблоны для отчетов и указывать источники данных, на основе которых они формируются (рисунок 10.1).

Параметры отчета

Название отчета * Тип шаблона *

Вклады. Отчет по пролонгированным вкладам Слово

Формировать в фоне

Фильтр ✎ ✏ ☰ +

Наименование фильтра: **report_prolonged_deposits_filter**

SQL-сервис ✎ ✏ ☰ +

Наименование SQL-сервиса: **rep_dep_prolong**

Шаблон ✎ ✏ ☰ +

Наименование шаблона: **Вклады. Отчет по пролонгированным вкладам**

Сохранить
К списку отчетов
Предпросмотр

Рисунок 10.1 Параметры отчета

Одним из параметров отчета является SQL-сервисы (рисунок 10.2) и используемый для формирования отчета шаблон (рисунок 10.3).

SQL сервис **rep_dep_prolong** редактирование

Общие | Группы | Политики безопасности | Связанные объекты | Администрирование

| | | | |
|-------------------------|---------------|------------------------------------|---------------|
| Идентификатор sqlserv * | Тип sqlserv * | <input type="checkbox"/> Проверено | Дата проверки |
| rep_dep_prolong | SQL | | 09.03.2022 |

Описание

| | | |
|--------|----------------------------|---------------------------|
| Автор | Автор последнего изменения | Дата последнего изменения |
| ARTDBA | ARTDBA | 2021-02-10 19:22:51 |

Тип БД * Кэширование на браузере

Параметры

SQL текст *

```
select * from table( pckartrep_dep_prolong.dep_prolong(p_date_from => :p_date_from, p_date_to => :p_date_to, p_idriskgroup => :p_idriskgroup )) t
```

Рисунок 10.2 Параметры отчета

Шаблон ✎ ✏ ☰ +

Наименование шаблона: **Вклады. Отчет по пролонгированным вкладам**

Наименование шаблона *

Вклады. Отчет по пролонгированным вкладам

Создать новый шаблон
Загрузить шаблон
Редактировать
Отмена

Рисунок 10.3 Шаблон отчета

1.4.11 Рабочее место оператора

Компонента состоит из трех функциональных блоков:

- Клиент;
- Договоры;
- Операции.

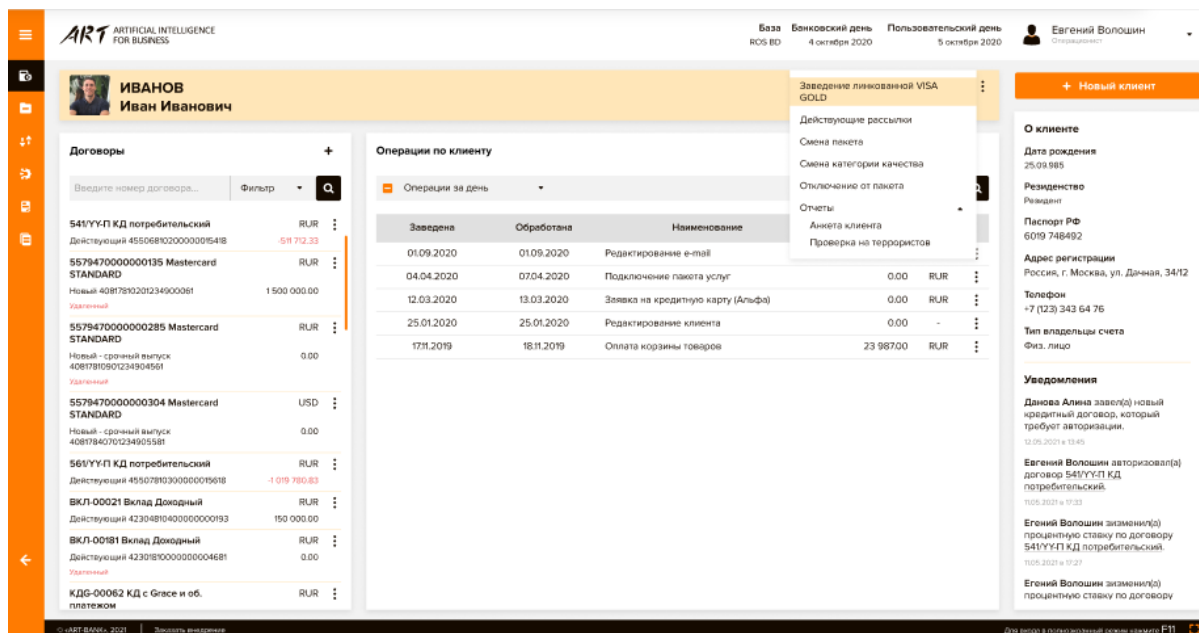


Рисунок 11.1 Рабочее место оператора

В блоке клиент реализован функционал заведения нового клиента, редактирования данных и документов по действующему клиенту, формирование ряда документов и так далее (рисунок 11.2).

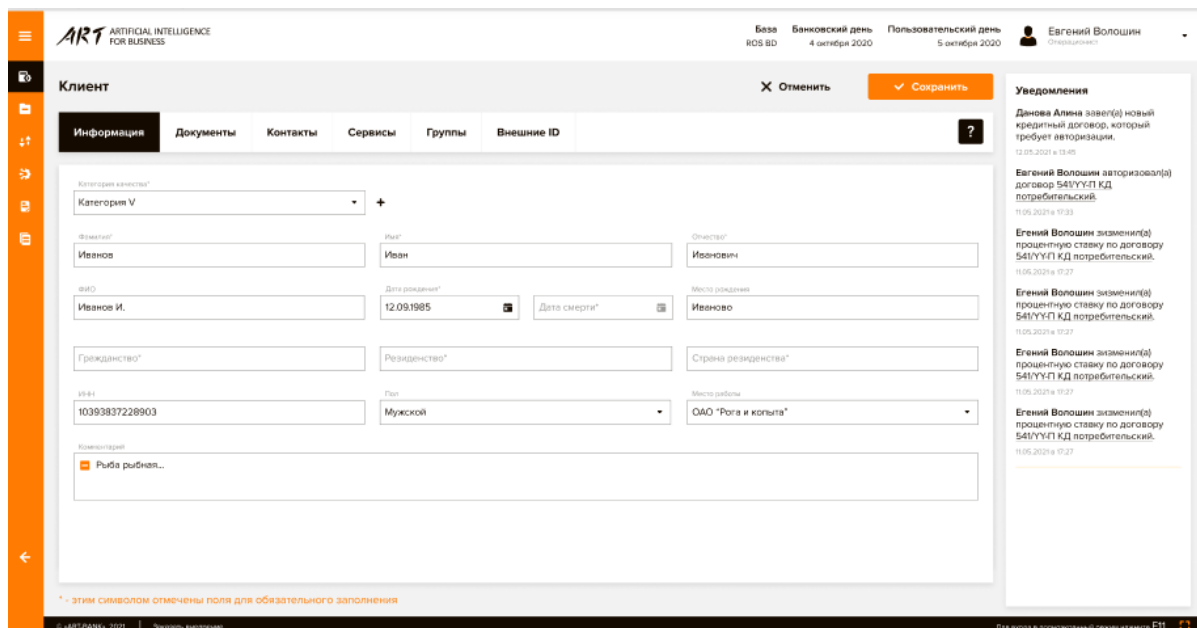


Рисунок 11.2 Блок «Клиент»

В блоке «Договоры» оператор может совершить необходимые операции по договорам клиента (рисунок 11.3).

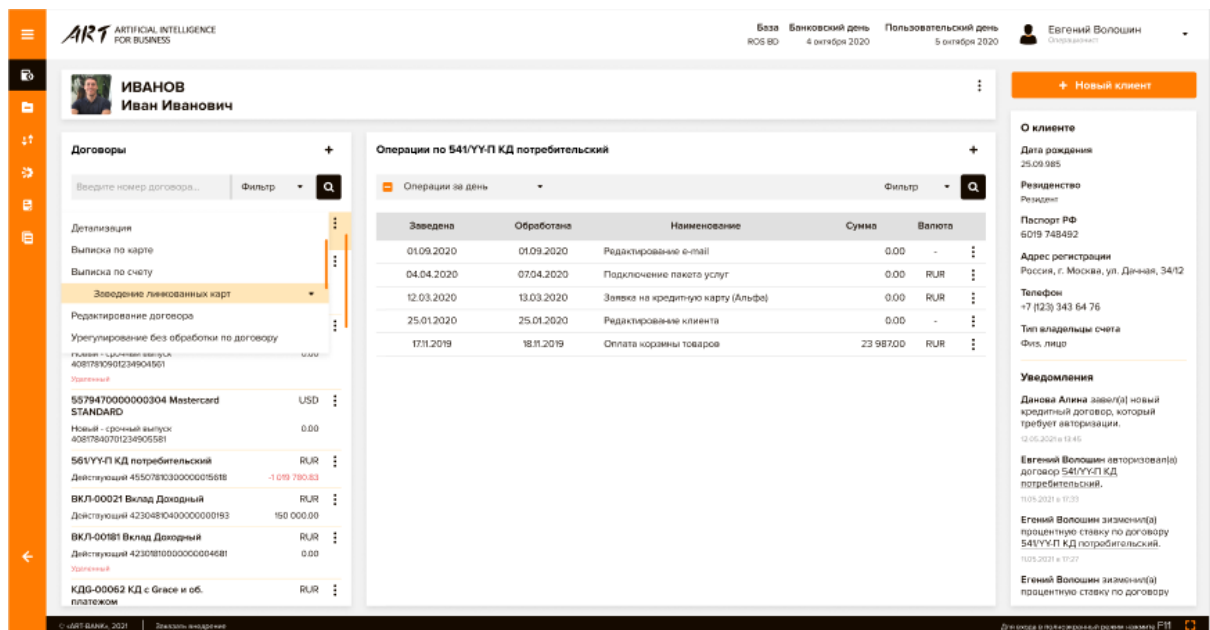


Рисунок 11.3 Блок «Договоры»

В блоке «Операции» доступен функционал заведения новых операций, получение информации о текущих операциях и действия по ним (рисунок 11.4).

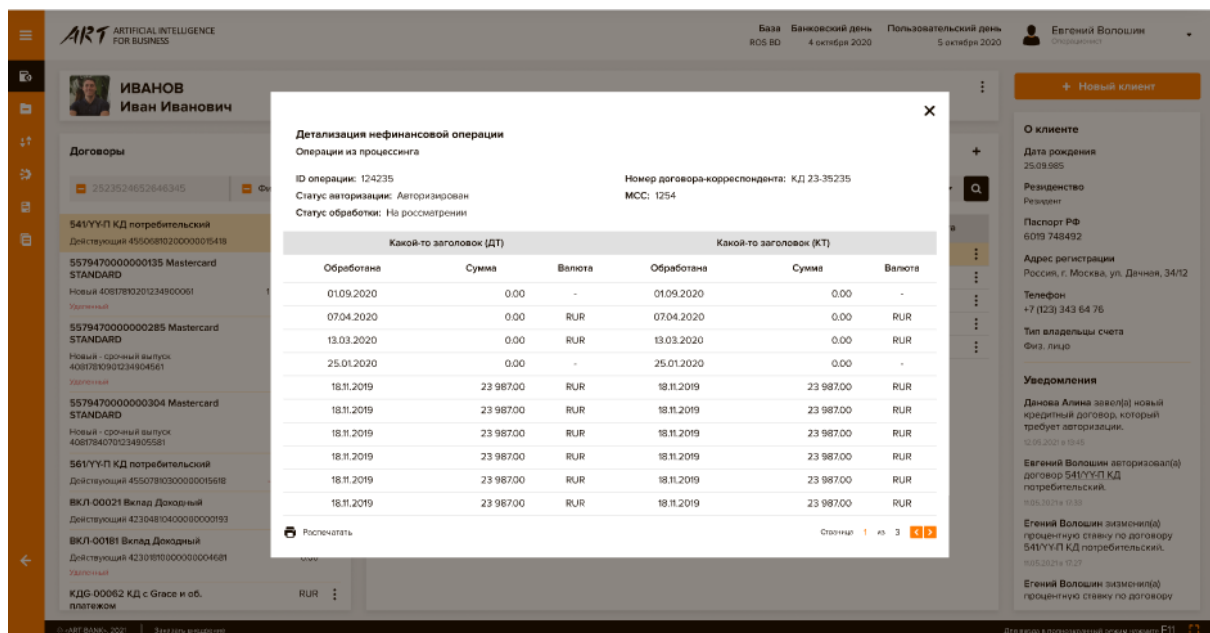


Рисунок 11.4 Блок «Операции»

1.4.12 Финансовый мониторинг

Данная функциональная компонента позволяет настраивать правила мониторинга транзакций клиентов как в соответствии с требованиями Федерального закона №115-ФЗ «О противодействии легализации (отмыванию) доходов, полученных преступным путём, и финансированию терроризма», так и в соответствии с внутренними регламентами и нормами самой финансовой организации (рисунки 12.1 и 12.2).

| ID | Код | Тип | Наименование | Расшифровка | Реестр типов операций для проверки по критерию | Реестр типов операций-исключений | Реестр КНП | Категория периода | Категория сумм |
|----|-----|---------------------|-------------------|--|--|----------------------------------|------------------------|-------------------|----------------|
| 1 | | Превышение суммы | Критерий 1 | Совершение сделки, не имеющей очевидного экономического смысла | | LIST_OF_TRANSTYPE_COMPLIANC | | | |
| 2 | | Превышение суммы | Критерий 2 | Совершение действий, направленных на уклонение от процедур финансового мониторинга | LIST_OF_TRANSTYPE_COMPLIANC | | | До 7 дней | От 500 000 |
| 3 | | Превышение суммы | Критерий 3 | Финансирование терроризма и/или экстремизма | LIST_OF_TRANSTYPE_COMPLIANC | | | | |
| 4 | | Превышение суммы | Критерий 4 | Обналичивание денег, полученных незаконным путем | LIST_OF_TRANSTYPE_COMPLIANC | | | До 7 дней | От 1 000 000 |
| 5 | | Превышение суммы | Критерий 5 | Операции, в которых получатель является нерезидентом | LIST_OF_TRANSTYPE_COMPLIANC | | | | |
| 6 | | Превышение суммы | Критерий 6 | Операции, одной из сторон которых являются лица, включенные в перечень организаций | LIST_OF_TRANSTYPE_COMPLIANC | | | | |
| 1 | | Накопительно внутри | Пороговая сумма 1 | Снятие или зачисление на сумму > 7 000 000 | LIST_OF_TRANSTYPE_COMPLIANC | | | До 7 дней | От 7 000 000 |
| 2 | | Накопительно внутри | Пороговая сумма 2 | Зачисление или перевод клиентом с регистрацией в оффшорной зоне на сумму > 2 000 000 | LIST_OF_TRANSTYPE_COMPLIANC | | | До 7 дней | 0 |
| 3 | | Накопительно внутри | Пороговая сумма 3 | Платежи и переводы денег, осуществляемые на безвозмездной основе, на сумму > 6 000 000 | LIST_OF_TRANSTYPE_COMPLIANC | | REGISTER_COMPLIANCEKNP | | От 6 000 000 |
| 4 | | Накопительно внутри | Пороговая сумма 4 | Операции, совершаемые юридическими лицами, с момента государственной регистрации | LIST_OF_TRANSTYPE_COMPLIANC | | | До 7 дней | От 7 000 000 |
| -1 | | Превышение суммы | Снято | Признак снят в ручном режиме | | | | | |

Рисунок 12.1 Раздел «Критерии подозрительности»

| ID | Реестр критериев | Вид операции | кодичн | Источник | Принадлежность источника к оффшору | Премник | Принадлежность приемника к оффшору | Знак операции | Реестр типа операции |
|----|------------------------------------|---|--------|-----------|------------------------------------|-----------|------------------------------------|---------------|---------------------------------|
| 17 | REGISTERS_COMPLIANCECRITERION_0111 | Получение выигрыша по результатам проведения пари | | | | | | | |
| 21 | REGISTERS_COMPLIANCECRITERION_0121 | Получение выигрыша по результатам проведения азартных игр | | Физ. лицо | | | | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 1 | REGISTERS_COMPLIANCECRITERION_0511 | Снятие с банковского счета клиента денег как разовая операция | 1 | | | | | -1 | |
| 2 | REGISTERS_COMPLIANCECRITERION_0512 | Снятие с банковского счета клиента денег как операция | 2 | | | | | -1 | |
| 3 | REGISTERS_COMPLIANCECRITERION_0521 | Зачисление на банковский счет клиента денег как разовая операция | 1 | | | | | 1 | |
| 4 | REGISTERS_COMPLIANCECRITERION_0522 | Зачисление на банковский счет клиента денег как операция | 2 | | | | | 1 | |
| 5 | REGISTERS_COMPLIANCECRITERION_0611 | Зачисление на банковский счет клиента денег, осуществленное физическим лицом | 1 | | | Физ. лицо | | 1 | |
| 6 | REGISTERS_COMPLIANCECRITERION_0612 | Зачисление на банковский счет клиента денег, осуществленное физическим лицом | 2 | | | Физ. лицо | | 1 | |
| 7 | REGISTERS_COMPLIANCECRITERION_0613 | Зачисление на банковский счет клиента денег, осуществленное юридическим лицом | 1 | | | Юр. лицо | | 1 | |
| 8 | REGISTERS_COMPLIANCECRITERION_0614 | Зачисление на банковский счет клиента денег, осуществленное юридическим лицом | 2 | | | Юр. лицо | | 1 | |
| 9 | REGISTERS_COMPLIANCECRITERION_0621 | Перевод на банковский счет клиента денег, осуществленный физическим лицом | 1 | Физ. лицо | ✓ | | | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 10 | REGISTERS_COMPLIANCECRITERION_0622 | Перевод на банковский счет клиента денег, осуществленный физическим лицом | 2 | Физ. лицо | ✓ | | | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 11 | REGISTERS_COMPLIANCECRITERION_0623 | Перевод на банковский счет клиента денег, осуществленный юридическим лицом | 1 | Юр. лицо | ✓ | | | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 12 | REGISTERS_COMPLIANCECRITERION_0624 | Перевод на банковский счет клиента денег, осуществленный юридическим лицом | 2 | Юр. лицо | ✓ | | | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 13 | REGISTERS_COMPLIANCECRITERION_0631 | Перевод денег клиентом в пользу физических лиц, инициированный физическим лицом | 1 | | | Физ. лицо | ✓ | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 14 | REGISTERS_COMPLIANCECRITERION_0632 | Перевод денег клиентом в пользу физических лиц, инициированный юридическим лицом | 2 | | | Физ. лицо | ✓ | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 15 | REGISTERS_COMPLIANCECRITERION_0633 | Перевод денег клиентом в пользу юридических лиц, инициированный юридическим лицом | 1 | | | Юр. лицо | ✓ | | REGISTER_COMPLIANCEOPERTRANSFEE |
| 16 | REGISTERS_COMPLIANCECRITERION_0634 | Перевод денег клиентом в пользу юридических лиц, инициированный юридическим лицом | 2 | | | Юр. лицо | ✓ | | REGISTER_COMPLIANCEOPERTRANSFEE |

Рисунок 12.2 Раздел «Справочник определения вида операции»

В Журнале финансового мониторинга отображаются операции, которые признаны подозрительными (рисунок 12.3).

| Журнал финансового мониторинга | | | | | | | | | |
|--------------------------------|-------------------------|----------------|---------------------|---------------------|-----------------|------------------|-----------------------|------------------------------------|------------------|
| выпуск подзор. | выпуск по пор. сумме | ID Операции | Дата операции | Дата операции в АРТ | Дата проведения | Статус обработки | Статус авторизации | Тип операции | Договор источник |
| | | 195770515 | 17.06.2013 14:38:16 | 08.08.2013 14:38:16 | 17.06.2013 | 0 Новая | Авторизовано | Пополнение счета безналичное | 413263*****1507 |
| | | 195770514 | 17.06.2013 14:37:55 | 08.08.2013 14:37:55 | 17.06.2013 | 0 Новая | Авторизовано | Пополнение счета наличными | 413263*****1507 |
| | | 195770513 | 17.06.2013 14:30:10 | 08.08.2013 14:30:10 | 17.06.2013 | 100 Обработана | Авторизовано | Пополнение счета наличными | 441327*****227 |
| | | 195770512 | 17.06.2013 14:30:00 | 08.08.2013 14:30:00 | 17.06.2013 | 100 Обработана | Авторизовано | Пополнение счета наличными | 441327*****227 |
| | | 195770509 | 17.06.2013 13:51:34 | 08.08.2013 13:51:34 | 17.06.2013 | 0 Новая | Удалено | Пополнение счета наличными | 413264*****7236 |
| | | 195770508 | 17.06.2013 13:50:17 | 08.08.2013 13:50:17 | 17.06.2013 | 0 Новая | Удалено | Пополнение счета наличными | 413264*****7236 |
| | | 195767908 | 31.05.2013 12:47:12 | 27.06.2013 12:47:12 | 31.05.2013 | 100 Обработана | Авторизовано | Пополнение счета наличными | 444444*****163 |
| | | 195755818 | 29.05.2013 14:05:07 | 29.05.2013 14:05:07 | 29.05.2013 | 100 Обработана | Авторизовано | Пополнение счета наличными | 441328*****2767 |
| | | 195753489 | 29.05.2013 12:39:34 | 29.05.2013 12:39:34 | 29.05.2013 | 100 Обработана | Авторизовано | Перечисление заработной платы | 441328*****3668 |
| | | 195735893 | 28.05.2013 09:53:03 | 10.05.2013 21:45:33 | 28.05.2013 | 100 Обработана | Авторизовано | Перевод на счет в АБС (внутренняя) | 441328*****6818 |
| | | 195734625 | 28.05.2013 10:37:53 | 29.05.2013 08:40:55 | 28.05.2013 | 100 Обработана | Авторизовано | Выдача наличных в АТМ | 413263*****9572 |
| | | 195734624 | 28.05.2013 10:36:37 | 29.05.2013 08:40:55 | 28.05.2013 | 100 Обработана | Авторизовано | Выдача наличных в АТМ | 413263*****9572 |
| | | 195734623 | 28.05.2013 10:35:42 | 29.05.2013 08:40:55 | 28.05.2013 | 100 Обработана | Авторизовано | Выдача наличных в АТМ | 413263*****9572 |
| | | 195734622 | 28.05.2013 10:34:47 | 29.05.2013 08:40:55 | 28.05.2013 | 100 Обработана | Авторизовано | Выдача наличных в АТМ | 413263*****9572 |

Рисунок 12.3 Журнал финансового мониторинга

1.4.13 Account processing

Данная компонента позволяет вести в онлайн режиме:

- актуальный доступный клиенту остаток по счетам (с учетом задолженности клиента, арестов и т.п.);
- хранилище данных - получать актуальную аналитическую информацию по договорам на все каналы (ДБО, терминалы, фронт офисы и так далее).

Настройка бизнес-логики авторизации (подтверждения) операций, взаимодействия с ДБО и АБС, правил холдирования средств, тарификации и иных разделов компоненты настраивается с помощью brm-моделей.

Компонента имеет механизм кросс-системной оркестровки (поддержка распределённых транзакций).

1.4.14 Мобильный банк

Данная компонента представляет собой клиентский мобильный банк, в котором клиент может управлять своими счетами/договорами, осуществлять различные операции, открывать новые счета/договора по предлагаемым банком продуктам.

С помощью данной компоненты пользователь может контролировать операции за определенный период времени (рисунки 14.1), совершать их (рисунки 14.2-14.3), просмотреть более детализированную информацию по конкретной транзакции (рисунки 14.4-14.5).

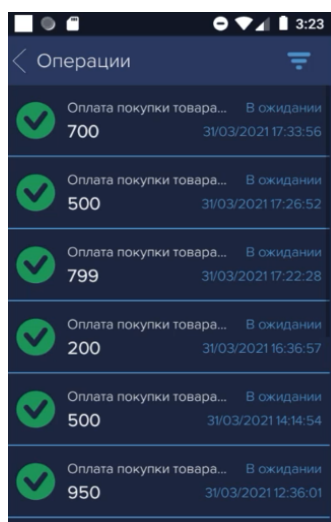


Рисунок 14.1 Операции за период

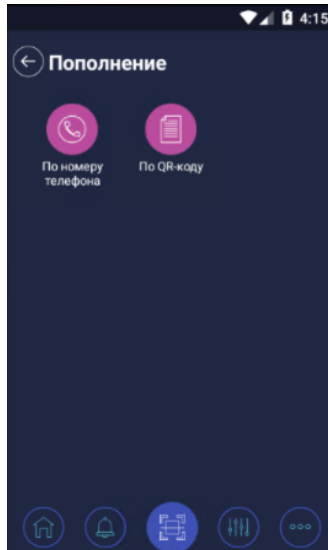


Рисунок 14.2 Раздел «Пополнение»

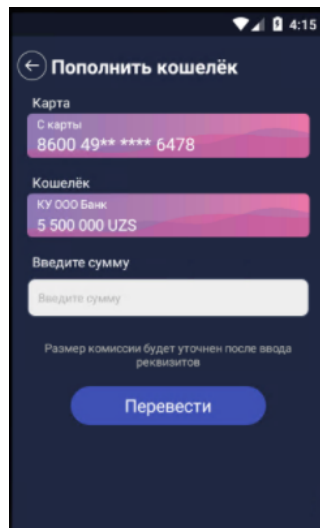


Рисунок 14.3 Операция по переводу денежных средств

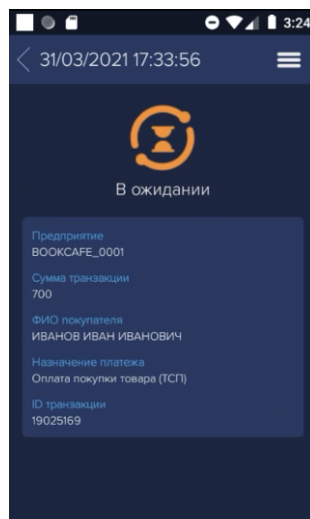


Рисунок 14.4 Детализация транзакции

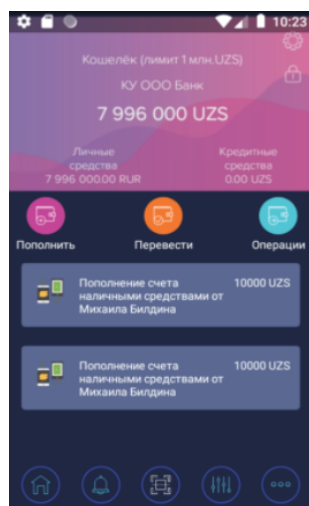


Рисунок 14.5 Детализация движения денежных средств

Так же с помощью данной функциональной компоненты имеется возможность осуществлять платежи (рисунок 14.3) и оформлять заказы на доставку товаров от разных торговых точек (рисунок 14.4) как с помощью партнерских служб доставки, так и посредством пунктов выдачи товаров (рисунок 14.5).

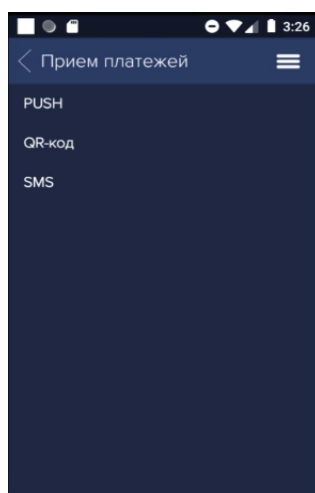


Рисунок 14.3 раздел «Платежи»

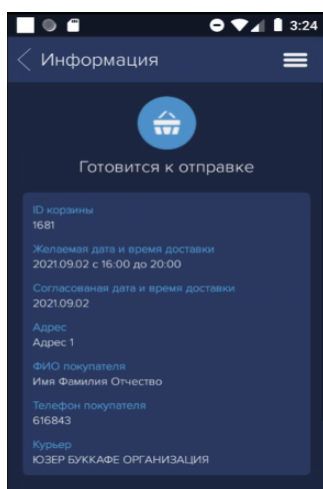


Рисунок 14.4 раздел «Заказы»

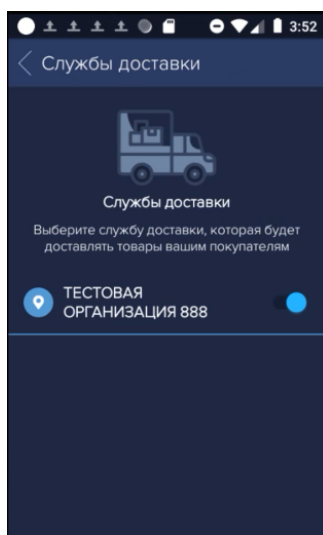


Рисунок 14.5 раздел настроек службы доставки

1.4.15 Интернет-банк

Данная компонента представляет собой браузерное приложение для дистанционного банковского обслуживания клиента. В котором клиент может управлять своими счетами/договорами, осуществлять различные операции, открывать новые счета/договора по предлагаемым банком продуктам.

Посредством данного решения пользователь может осуществлять переводы между своими счетами, отправлять платежи в Банк и получать выписки по счетам (информацию о движениях средств на счёте) из Банка.

7. Установка ПО

7.1 Требования к системе

1. Для установки ПО «АРТ-Финтех» необходимо:
 - Операционная система: Ubuntu v.16+ Linux;
 - Наличие 10 гигабайт свободного пространства в хранилище данных;
 - 4 GB оперативной памяти;
 - Postgresql 13;
2. Для мобильных приложений:
 - Операционная система «Android» версии не ниже 7.
 - Операционная система «iOS» версии не ниже 10.
3. Для приложений, использующих браузеры:
 - Safari 11 (macOS 10.12 Sierra, OS X 10.11 El Capitan, macOS 10.13 High Sierra)
 - Safari 12 (macOS 10.14 Mojave)
 - Google Chrome - версии, начиная с 35.0.1916.114
 - Microsoft Edge - версии, начиная с 42.17134 (windows 10.0.17134)
 - Mozilla Firefox - версии, начиная с 31 (windows 10.0.17134)
 - Opera - версии, начиная с 36.0.2130.32
 - Yandex - версии, начиная с 18.11.1

7.2 Установка PostgreSQL 13

1. Обновление пакетов и перезагрузка операционной системы:
 - `sudo yum update`
 - `sudo reboot`
2. Установка репозитория:

```
yum install -y https://download.postgresql.org/pub/repos/yum/repopms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```
3. Установка Postgresql
 - `yum install -y postgresql13-server`
 - `yum install -y postgresql13-server postgresql13-contrib`

Для демонстрационной версии дополнительных настроек PostgreSQL не требуется.

7.3 Установка инфраструктуры и сервисов среднего слоя, фронтального и мобильных приложений

7.3.1 Конфигурация серверов

Рекомендуемая минимальная конфигурация серверов:

| Код | Назначение | CPU | Мем, Гб | HDD, Гб |
|-----|--|-----|---------|---------|
| S1 | Хост 1 кластера среднего слоя | 2 | 4 | 40 |
| S2 | Хост 2 кластера среднего слоя | 2 | 4 | 40 |
| S3 | Хост 3 кластера среднего слоя | 2 | 4 | 40 |
| L1 | Хост 1 инфраструктуры (логи, мониторинг) | 4 | 8 | 200 |
| F1 | Веб-сервер Nginx (хостинг фронтального приложения) | 4 | 4 | 40 |

Для некоторых этапов установки будет использоваться средство автоматизации Ansible (<https://www.ansible.com/>).

Его требуется предварительно развернуть либо на хосте управления (если такой предусматривается), либо на одном из целевых хостов (например, L1).

Пример установки (Ubuntu):

```
# apt update
# apt install software-properties-common
# add-apt-repository --yes --update ppa:ansible/ansible
# apt install ansible
```

Основная часть компонентов ПО устанавливается на кластер из 3-х (желательно) узлов - компьютеров под управлением операционной системы Debian v.10+ / Ubuntu v.16+ Linux.

Для организации кластера используются штатные средства ПО Docker (Docker Swarm).

7.3.2 Порядок установки

1. Подготовка Docker
 - 1.1 Подготовка (обновление) пакетов, перезагрузка

```
# apt update && apt -y upgrade
```

- ```
reboot
```
- 1.2 Создание рабочего пользователя (здесь - с именем "art")
- ```
# adduser -U art
```
- 1.3 Установка Docker
- ```
apt-get update
apt-get install ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/debian \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

apt-get update
apt-get install docker-ce docker-ce-cli containerd.io
```
- 1.4 Создание группы "docker", добавление пользователя art в группу docker, вход под пользователем art
- ```
# usermod -aG docker art
# exit
```

Пункты 1.1-1.4 выполняются на хостах S1, S2, S3.

- 1.5 Инициализация Docker Swarm (на хосте S1):
- ```
$ docker swarm init --advertise-addr <ip>
```
- где <ip> - IP-адрес внешнего интерфейса на S1
- 1.6 Получение токена приглашения вступления в swarm-кластер в качестве manager-ноды
- ```
$ docker swarm join-token -q manager
```
- Все 3 узла будут в роли manager.
- 1.7 Добавление остальных 2-х хостов в swarm
- ```
$ docker swarm join --token <manager_token> <ip_1>:2377
```
- <manager\_token> - полученный на шаге 6 токен  
<ip\_1> - адрес 1-го хоста (S1).
2. Инфраструктурные компоненты
- 2.1 Подсистема сбора логов - EFK-стэк (Elasticsearch, Fluentd/Fluent-bit, Kibana).  
В случае высоконагруженной системы с плотным потоком лог-сообщений следует рассмотреть выделение дополнительных ресурсов для организации полноценного Elasticsearch-кластера.  
Здесь рассматривается упрощенный вариант - установка на единственный инфраструктурный хост (L1).
- 2.1.1 Установка Elasticsearch.  
Используем ansible-роль <https://github.com/elastic/ansible-elasticsearch.git>  
Пример ansible-плейбука для запуска установки:
- ```
- name: ES install
tags: elasticsearch
hosts: log
become: yes
roles:
  - role: elasticsearch
vars:
  es_data_dirs:
    - "/home/art/elasticsearch/data"
  es_config:
    node.name: "artlog"
    cluster.name: "artlog-cluster"
```



```
discovery.type: "single-node"
node.data: true
node.master: true
bootstrap.memory_lock: true
http.host: 0.0.0.0
http.port: 9200
transport.host: 0.0.0.0
transport.port: 9300
xpack.security.enabled: false
es_heap_size: 2g
es_api_host: 0.0.0.0
es_api_port: 9200
```

Запуск установки (выполнения плейбука):

```
$ ansible-playbook install.yml -i hosts.ini --become -K
```

hosts.ini - inventory-файл, где указываются целевые хосты установки (IP, доступ).

Пример inventory-файла:

```
log-1 ansible_host=192.168.22.111
```

```
[log]
```

```
log-1
```

```
[log:vars]
```

```
ansible_connection=ssh
```

```
ansible_port=22
```

```
ansible_user=art
```

2.1.2 Установка средства просмотра логов Kibana.

Пример ansible-плейбука для запуска установки:

```
- name: Kibana install
```

```
tags: kibana
```

```
hosts: log
```

```
become: yes
```

```
tasks:
```

```
- name: Ensure dependencies are installed.
```

```
apt:
```

```
name:
```

```
- apt-transport-https
```

```
- gnupg2
```

```
state: present
```

```
- name: Add Elasticsearch apt key.
```

```
apt_key:
```

```
url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

```
state: present
```

```
- name: Add Kibana repository.
```

```
apt_repository:
```

```
repo: 'deb https://artifacts.elastic.co/packages/7.15.1/apt stable main'
```

```
state: present
```

```
update_cache: true
```

```
- name: Updating the config file to restrict outside access
```

```
lineinfile:
```

```
destfile: /etc/kibana/kibana.yml
```

```
regexp: 'server.host:'
```

line: 'server.host: localhost'

- name: Updating the config file to disable xpack.security
lineinfile:

destfile: /etc/kibana/kibana.yml

regexp: 'xpack.security.enabled:'

line: 'xpack.security.enabled: false'

- name: Enabling Kibana service

systemd:

name: kibana

enabled: yes

daemon_reload: yes

- name: Starting Kibana service

systemd:

name: kibana

state: started

2.1.3 Установка сборщика логов Fluentd.

Пример ansible-плейбука для запуска установки:

- name: Fluentd collector

tags: fluentd

hosts: log

become: yes

tasks:

- name: fluentd | install package key

apt_key:

url: <https://packages.treasuredata.com/GPG-KEY-td-agent>

state: present

register: result

- name: fluentd | install repo

apt_repository:

repo: deb <http://packages.treasuredata.com/4/ubuntu/bionic/> bionic contrib

state: present

register: result

when: result|changed

- name: fluentd | update repo cache

apt:

update_cache: yes

when: result|changed

- package:

name: td-agent

state: latest

- copy:

dest: /etc/td-agent/td-agent.conf

mode: 0644

content: |

<source>

@type forward

bind 0.0.0.0

```

    port 24224
</source>

<filter *.*>
  @type parser
  key_name log
  reserve_data true
  remove_key_name_field true
  emit_invalid_record_to_error false
  <parse>
    @type json
  </parse>
</filter>

<match fluent.*>
  @type stdout
</match>

<match *.*>
  @type elasticsearch
  @log_level warn
  include_tag_key true
  host 127.0.0.1
  port 9200
  scheme http
  ssl_verify false
  logstash_format true
  logstash_prefix ${tag}
  reload_connections false
  reconnect_on_error true
  reload_on_failure true
  <buffer>
    timekey 10s
    flush_thread_count 5
    flush_interval 5s
    chunk_limit_size 8m
    queue_limit_length 96
    flush_mode interval
    retry_max_interval 30
    retry_type exponential_backoff
    retry_forever
    flush_at_shutdown true
    overflow_action block
  </buffer>
</match>

```

```

- name: Start fluentd
  systemd:
    name: td-agent
    state: started
    enabled: yes

```

Fluentd здесь настраивается как сервис, слушающий на порту 24224, куда лог-сообщения отправляются агентами Fluent-bit, установленными на хосты

S1/S2/S3. Результат отправляется в сервис Elasticsearch, запущенный на том же L1-хосте.

2.2 Подсистема мониторинга (Prometheus, Grafana, Exporters)

Подсистема мониторинга состоит из следующих компонент:

- ПО сбора метрик и ведения TSDB (time series database) - Prometheus;
- ПО рассылки сообщений тревоги - alertmanager;
- средства сбора метрик – exporters; здесь рассматривается node-exporter – метрики состояния хоста (критическая нагрузка, риск нехватки памяти и т.д.);
- ПО визуализации данных из TSDB Prometheus – Grafana.

2.2.1 Установка Prometheus

Используется ansible-роль <https://github.com/cloudalchemy/ansible-prometheus.git>

```
- name: Prometheus install
```

```
tags: prometheus
```

```
hosts: log
```

```
become: yes
```

```
roles:
```

```
- role: ansible-prometheus
```

```
vars:
```

```
prometheus_db_dir: "/home/art/prometheus/data"
```

```
prometheus_storage_retention: "30d"
```

```
prometheus_global:
```

```
scrape_interval: "15s"
```

```
scrape_timeout: "15s"
```

```
evaluation_interval: "10s"
```

```
prometheus_alertmanager_config:
```

```
- scheme: http
```

```
static_configs:
```

```
- targets: ["127.0.0.1:9093"]
```

```
prometheus_scrape_configs:
```

```
- job_name: "node"
```

```
metrics_path: /metrics
```

```
static_configs:
```

```
- targets:
```

```
- "<IP_F1>:9100"
```

```
- "<IP_L1>:9100"
```

```
- "<IP_S1>:9100"
```

```
- "<IP_S2>:9100"
```

```
- "<IP_S3>:9100"
```

```
prometheus_alert_rules:
```

```
- alert: Watchdog
```

```
expr: vector(1)
```

```
for: 10m
```

```
labels:
```

```
severity: warning
```

```
annotations:
```

```
description: "This is an alert meant to ensure that the entire alerting pipeline is functional.\nThis alert is always firing, therefore it should always be firing in Alertmanager\nand always fire against a receiver. There are integrations with various notification\nmechanisms that send a notification when this alert is not firing. For example the\n\"DeadMansSnitch\" integration in PagerDuty."
```

```

summary: 'Ensure entire alerting pipeline is functional'
- alert: InstanceDown
  expr: 'up == 0'
  for: 5m
  labels:
    severity: critical
  annotations:
    description: '{% raw %}Хост {{ $labels.instance }} (задание {{ $labels.job
}}) недоступен в течении 5 минут. {% endraw %}'
    summary: '{% raw %}Хост {{ $labels.instance }} недоступен. {% endraw %}'
- alert: NodeFilesystemAlmostOutOfSpace
  annotations:
    description: '{% raw %}На файловой системе {{ $labels.device }} хост {{
$labels.instance }} осталось доступно только {{ printf "%.2f" $value }}%
свободного места. {% endraw %}'
    summary: 'Осталось менее 5% свободного места на ФС.'
    expr: "(n      node_filesystem_avail_bytes{job=\"node\",fstype!=\"\"} /
node_filesystem_size_bytes{job=\"node\",fstype!=\"\"} * 100 < 5\nand\n
node_filesystem_readonly{job=\"node\",fstype!=\"\"} == 0\n)\n"
    for: 1h
    labels:
      severity: warning
- alert: NodeFilesystemAlmostOutOfSpace
  annotations:
    description: '{% raw %}На файловой системе {{ $labels.device }} хост {{
$labels.instance }} осталось доступно только {{ printf "%.2f" $value }}%
свободного места. {% endraw %}'
    summary: 'Осталось менее 3% свободного места на ФС.'
    expr: "(n      node_filesystem_avail_bytes{job=\"node\",fstype!=\"\"} /
node_filesystem_size_bytes{job=\"node\",fstype!=\"\"} * 100 < 3\nand\n
node_filesystem_readonly{job=\"node\",fstype!=\"\"} == 0\n)\n"
    for: 1h
    labels:
      severity: critical
- alert: NodeFilesystemAlmostOutOfFiles
  annotations:
    description: '{% raw %}На файловой системе {{ $labels.device }} хост {{
$labels.instance }} осталось доступно только {{ printf "%.2f" $value }}%
свободных дескрипторов (inodes). {% endraw %}'
    summary: 'Осталось менее 5% свободных дескрипторов (inodes) на ФС.'
    expr: "(n      node_filesystem_files_free{job=\"node\",fstype!=\"\"} /
node_filesystem_files{job=\"node\",fstype!=\"\"} * 100 < 5\nand\n
node_filesystem_readonly{job=\"node\",fstype!=\"\"} == 0\n)\n"
    for: 1h
    labels:
      severity: warning
- alert: NodeFilesystemAlmostOutOfFiles
  annotations:
    description: '{% raw %}На файловой системе {{ $labels.device }} хост {{
$labels.instance }} осталось доступно только {{ printf "%.2f" $value }}%
свободных дескрипторов (inodes). {% endraw %}'
    summary: 'Осталось менее 3% свободных дескрипторов (inodes) на ФС.'

```

```

    expr:  "\n          node_filesystem_files_free{job=\"node\",fstype!=\"\"} /
node_filesystem_files{job=\"node\",fstype!=\"\"} * 100 < 3\nand\n
node_filesystem_readonly{job=\"node\",fstype!=\"\"} == 0\n)\n"
    for: 1h
    labels:
    severity: critical

```

В этом плейбуке описано задание сборщика node-exporter, а также настройка взаимодействия с ПО рассылки алертов alertmanager и несколько правил с условиями формирования алертов. Список правил далеко не исчерпывающий, приведен для примера.

2.2.2 Установка ПО рассылки сообщений тревоги – alertmanager

Используется `ansible-роль` <https://github.com/cloudalchemy/ansible-alertmanager.git>

```

- name: alertmanager install
  tags: alertmanager
  hosts: log
  become: yes
  roles:
    - role: ansible-alertmanager
  vars:
    alertmanager_web_listen_address: 127.0.0.1:9093
    alertmanager_route:
      receiver: 'email'
      group_by: ['alertname']
      group_wait: 10s
      group_interval: 30s
      repeat_interval: 5m
    routes:
      - receiver: "empty"
        match:
          alertname: Watchdog
    alertmanager_receivers:
      - name: "empty"
      - name: "email"
    email_configs:
      - to: 'alert@company.ru'
        from: 'noreply@company.ru'
        smarthost: "smtp.provider.ru:465"
        auth_username: 'noreply@company.ru'
        auth_identity: 'noreply@company.ru'
        auth_password: 'secret'
        require_tls: no
        tls_config:
          server_name: smtp.provider.ru

```

Для примера приведена настройка рассылки через канал электронной почты.

2.2.3 Установка средства сбора метрик с хоста - node-exporter

Используется `ansible-роль` <https://github.com/cloudalchemy/ansible-node-exporter.git>

```

- name: node-exporter install
  tags: node_exporter

```

```
hosts: node_exp
become: yes
roles:
  - role: ansible-node-exporter
vars:
  node_exporter_web_listen_address: "0.0.0.0:9100"
```

node_exp – группа хостов из inventory-файла hosts.ini. В эту группу входят все рассматриваемые хосты – S1, S2, S3, F1, L1. Т.о. все они попадают под наблюдение.

2.2.4 Установка ПО визуализации данных из TSDB Prometheus – Grafana

Используется ansible-роль <https://github.com/cloudalchemy/ansible-node-exporter.git>

```
- name: node-exporter install
tags: node_exporter
hosts: node_exp
become: yes
roles:
  - role: ansible-node-exporter
vars:
  node_exporter_web_listen_address: "0.0.0.0:9100"
```

node_exp – группа хостов из inventory-файла hosts.ini. В эту группу входят все рассматриваемые хосты – S1, S2, S3, F1, L1. Т.о. все они попадают под наблюдение.

Используется ansible-роль <https://github.com/cloudalchemy/ansible-grafana.git>

```
- name: grafana install
tags: grafana
hosts: log
become: yes
roles:
  - role: ansible-grafana
vars:
  grafana_data_dir: "/home/art/grafana/data"
  grafana_datasources:
    - name: prometheus
      type: prometheus
      access: proxy
      url: 'http://127.0.0.1:9090'
      basicAuth: false
      isDefault: true
      version: 1
      editable: true
```

При установке сразу заводится источник данных prometheus.

2.3 Подсистема кэширования (Redis)

Redis разворачивается в виде кластера из 6 экземпляров, по 2 на каждый из хостов S1/S2/S3.

Схема размещения (каждый из экземпляров обозначен номером публикуемого порта: 6381, 6382, ...):

S1: 6381 6384

S2: 6382 6385

S3: 6383 6386

Схема отношений master-node/slave-node в кластере:

6381(M) / 6385(S)

6382(M) / 6386(S)

6383(M) / 6384(S)

Конфигурация каждого узла (redis.conf):

(для примера – узла 6381 на S1)

port 6381

appendonly yes

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

save 900 1

save 300 10

save 60 10000

maxmemory 200mb

maxmemory-policy allkeys-lfu

maxmemory-samples 5

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

cluster-announce-ip <IP_S1>

cluster-announce-port 6381

Назначение меток нодам docker-swarm кластера для распределения узлов redis-кластера:

Выполняется на S1:

```
$ docker node update --label-add redis_node_6381=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add redis_node_6384=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Выполняется на S2:

```
$ docker node update --label-add redis_node_6382=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add redis_node_6385=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Выполняется на S3:

```
$ docker node update --label-add redis_node_6383=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add redis_node_6386=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Конфигурация разворачивания docker-сервиса узла redis-кластера (в составе файла конфигурации деплоя stack.yml):

(пример одного из узлов - 6381)

version: '3.3'

services:

...

...


```

redis_6381:
  image: "docker.art-bank.ru/redis:1"
  volumes:
    - /etc/localtime:/etc/localtime:ro
    - /home/art/infra/data/redis/6381:/data
    - /home/art/infra/config/redis/6381/redis.conf:/usr/local/etc/redis/redis.conf
  ports:
    - "6381:6381"
    - "16381:16381"
  networks:
    - ilocal
    - pnet_mirlp
  command: ["redis-server","/usr/local/etc/redis/redis.conf"]
  deploy:
    mode: replicated
    replicas: 1
    placement:
      constraints:
        - node.labels.redis_node_6381 == yes
  ...
  ...
networks:
  ilocal:
    driver: overlay

```

При этом в /home/art/infra/config/redis/6381/redis.conf находится ранее подготовленный файл конфигурации соответствующего узла.

Запуск конфигурации деплоя stack.yml:

```
docker stack deploy --prune --compose-file stack.yml infra
```

Инициализация redis-кластера:

```

$ redis-cli --cluster create <IP_S1>:6381 <IP_S2>:6382 <IP_S3>:6383
$ redis-cli --cluster add-node <IP_S2>:6385 <IP_S1>:6381 --cluster-slave
$ redis-cli --cluster add-node <IP_S3>:6386 <IP_S2>:6382 --cluster-slave
$ redis-cli --cluster add-node <IP_S1>:6384 <IP_S3>:6383 --cluster-slave

```

2.4 Подсистема обмена сообщениями (Nats)

Конфигурация разворачивания (в составе файла конфигурации деплоя stack.yml):

```

nats-1:
  image: "docker.art-bank.ru/nats:2.2.0"
  networks:
    - ilocal
  volumes:
    - /etc/localtime:/etc/localtime:ro
    - /home/art/infra/config/nats/nats.conf:/etc/nats/nats.conf:ro
  command:
    - "--cluster"
    - "nats://0.0.0.0:6222"
    - "--http_port"
    - "8222"
    - "--port"
    - "4222"

```

```
- "--config"
- "/etc/nats/nats.conf"
ports:
- "14222:4222"
- "18222:8222"
deploy:
mode: replicated
replicas: 1
```

/home/art/infra/config/nats/nats.conf – конфигурационный файл Nats:

```
max_payload: 100Mb
```

2.5 NoSQL БД (MongoDB)

MongoDB разворачивается в виде кластера в рамках docker-swarm. Состоит из:

- 3-х MongoDB узлов данных (mongodata1, mongodata2, mongodata3);
- 3-х MongoDB узлов конфигурации (mongocfg1, mongocfg2, mongocfg3);
- 2-х MongoDB узлов маршрутизации (mongos1, mongos2).

Назначение меток нодам docker-swarm кластера для распределения узлов MongoDB-кластера:

Выполняется на S1:

```
$ docker node update --label-add mongo_data_1=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add mongo_cfg_1=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add mongo_s_1=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Выполняется на S2:

```
$ docker node update --label-add mongo_data_2=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add mongo_cfg_2=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add mongo_s_2=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Выполняется на S3:

```
$ docker node update --label-add mongo_data_3=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

```
$ docker node update --label-add mongo_cfg_3=yes $(docker system info 2> /dev/null | grep NodeID | awk '{ print $2 }')
```

Конфигурация разворачивания docker-сервисов кластера MongoDB (в составе файла конфигурации деплоя stack.yml):

```
mongodata1:
  image: "docker.art-bank.ru/mongodb:1"
  networks:
  - ilocal
  volumes:
  - /etc/localtime:/etc/localtime:ro
  - /home/art/infra/data/mongo/db:/data/db
  command: mongod --shardsvr --replSet datars --dbpath /data/db --bind_ip 0.0.0.0 --port
27017 --noauth
  deploy:
```

mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_data_1 == yes

mongodata2:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/infra/data/mongo/db:/data/db
command: mongod --shardsvr --replSet datars --dbpath /data/db --bind_ip 0.0.0.0 --port
27017 --noauth
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_data_2 == yes

mongodata3:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/infra/data/mongo/db:/data/db
command: mongod --shardsvr --replSet datars --dbpath /data/db --bind_ip 0.0.0.0 --port
27017 --noauth
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_data_3 == yes

mongocfg1:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/infra/data/mongo/cfg:/data/cfg
command: mongod --configsvr --replSet cfgrs --dbpath /data/cfg --bind_ip 0.0.0.0 --port
27017 --noauth
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_cfg_1 == yes

mongocfg2:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/infra/data/mongo/cfg:/data/cfg
command: mongod --configsvr --replSet cfgrs --dbpath /data/cfg --bind_ip 0.0.0.0 --port
27017 --noauth
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_cfg_2 == yes

mongocfg3:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/infra/data/mongo/cfg:/data/cfg
command: mongod --configsvr --replSet cfgrs --dbpath /data/cfg --bind_ip 0.0.0.0 --port
27017 --noauth
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_cfg_3 == yes

mongos1:
image: "docker.art-bank.ru/mongodb:1"
networks:
- ilocal
- pnet_mirlp
volumes:
- /etc/localtime:/etc/localtime:ro
command: mongos --configdb
cfgrs/mongocfg1:27017,mongocfg2:27017,mongocfg3:27017 --bind_ip 0.0.0.0 --port 27017 -
-noauth
ports:
- "27017:27017"
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_s_1 == yes

mongos2:
image: "docker.art-bank.ru/mongodb:1"
networks:

```

- ilocal
- pnet_mirlp
volumes:
- /etc/localtime:/etc/localtime:ro
command: mongos --configdb
cfgrs/mongocfg1:27017,mongocfg2:27017,mongocfg3:27017 --bind_ip 0.0.0.0 --port 27017 -
-noauth
ports:
- "27018:27017"
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.labels.mongo_s_2 == yes

```

Дополнительно, для инициализации MongoDB-кластера запускается сервис mongo-bootstrap:

```

mongo-bootstrap:
image: "docker.art-bank.ru/mongodb-bootstrap:1"
volumes:
- /etc/localtime:/etc/localtime:ro
networks:
- ilocal
command:
- '-dataSet=datars/mongodata1:27017,mongodata2:27017,mongodata3:27017'
- '-configSet=cfgrs/mongocfg1:27017,mongocfg2:27017,mongocfg3:27017'
- '-mongos=mongos1:27017,mongos2:27017'
deploy:
mode: replicated
replicas: 1

```

3. Сервисы среднего слоя

Средний слой комплектуется сервисами в зависимости от конфигурации решения. Каждый сервис представляет собой node.js приложение. Поставляется на площадку разворачивания в виде docker-образов (image), предварительно собранных по Dockerfile-сценарию поставщиком решения. Способ загрузки docker-образов зависит от доступности контура разворачивания. При наличии доступа из контура в интернет возможна загрузка из docker-репозитория компании-поставщика:

```

$ docker login docker.art-bank.ru
$ docker pull docker.art-bank.ru/<image_name>:<image_tag>

```

Примерная конфигурация разворачивания docker-сервиса приложения среднего слоя (в составе файла конфигурации деплоя stack.yml):

```

<svc_id>:
image: "docker.art-bank.ru/<app_id>:<app_tag>"
extra_hosts:
- "elasticsearch:<IP_S1>"
- "nats:<IP_S1>"
- "redis:<IP_S1>"
- "postgres:<IP_DB>"
- "mongodb:<IP_S1>"
command:

```

```

- "node"
- "index.js"
- "-n"
- "msg"
- "-r"
- "mks.2"
- "-f"
- "/home/node/app"
- "-p"
- "/run/secrets/<prj_id>_<app_id>.json"
volumes:
- /etc/localtime:/etc/localtime:ro
- /home/art/<prj_id>/settings/<prj_id>_<app_id>:/home/node/settings/<prj_id>_<app_id>
environment:
- "PROJECT_NAME=<prj_id>"
- "APP_NAME=<app_id>"
- "REDIS_CLUSTER=<IP_S1>:6381;<IP_S2>:6382;<IP_S3>:6383;"
- "NATS_CLUSTER=nats://nats:14222;"
- "ELASTICSEARCH_CLUSTER=http://elasticsearch:9200"
...
networks:
- pnet_<prj_id>
deploy:
  endpoint_mode: dnsrr
  mode: replicated
  replicas: 1
  update_config:
    parallelism: 1
    delay: 10s
    order: start-first
  restart_policy:
    condition: any
    delay: 10s
secrets:
- <prj_id>_<app_id>.json
logging:
  driver: "json-file"
  options:
    max-size: "500k"
    max-file: "5"
    tag: <prj_id>.<app_id>

```

Дополнительно к набору сервисов среднего слоя в рамках его стэка запускается ingress-сервис на основе конфигурации Nginx, через который организуется взаимодействие с API сервиса извне docker-стэка (он же – балансировщик нагрузки):

```

lbmw<prj_id>:
  image: "docker.art-bank.ru/nginx:1"
  ports:
    - "8000:8000"
  command: ["nginx", "-g", "daemon off;"]
  volumes:
    - /etc/localtime:/etc/localtime:ro
    - /home/art/<prj_id>/config/lbmw/nginx.conf:/etc/nginx/nginx.conf:ro

```

```
networks:
  - pnet_<prj_id>
deploy:
  mode: replicated
  replicas: 1
logging:
  driver: "json-file"
  options:
    max-size: "500k"
    max-file: "5"
    tag: <prj_id>.lbmw
```

Пример конфигурации (nginx.conf) ingress-сервиса:

```
user nginx;
worker_processes auto;

events {
  worker_connections 4096;
}

stream {
  access_log off;
  error_log /dev/stdout info;
  resolver 127.0.0.11 valid=10s ipv6=off;

  map $server_port $up_<app_id>_20001 {
    20001 "<app_id>:20001";
  }
  server {
    listen 20001;
    proxy_pass $up_<app_id>_20001;
  }
}

http {
  include mime.types;
  default_type application/octet-stream;
  charset UTF-8;
  resolver 127.0.0.11 valid=10s ipv6=off;

  access_log off;
  error_log /dev/stdout info;

  client_max_body_size 50m;

  server {
    listen 8080;
    client_max_body_size 0;
    proxy_connect_timeout 6000s;
    proxy_send_timeout 6000s;
    proxy_read_timeout 6000s;
    send_timeout 6000s;
```

```

set $up_<app_id>_def http://<app_id>:8080;

location /api/v1/<app_id> {
    proxy_pass $up_<app_id>_def;
    proxy_buffering off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_connect_timeout 6000s;
    proxy_send_timeout 6000s;
    proxy_read_timeout 6000s;
    send_timeout 6000s;
    client_max_body_size 0;

    if ($request_method = OPTIONS ) {
        add_header 'Access-Control-Allow-Origin' *;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
        add_header 'Access-Control-Allow-Headers' 'Origin, X-Requested-With, Content-Type,
Accept, Authorization';
        add_header 'Access-Control-Expose-Headers' 'Origin, X-Requested-With, Content-Type,
Accept, Authorization';
        add_header 'Access-Control-Max-Age' 1728000;
        add_header 'Content-Type' 'text/plain charset=UTF-8';
        add_header 'Content-Length' 0;
        return 204;
    }
}
...
}
}

```

Конфигурационные параметры в сервисы среднего слоя передаются либо через системные переменные (environments) – раздел environment конфигурации деплоя stack.yml, либо через docker-секреты, если параметры чувствительны к раскрытию (пароли, параметры подключения, ключи, сертификаты, и т.п.). В последнем случае такой «секрет» сначала необходимо сформировать в рамках docker-swarm:

```
$ cat <secret_source_file> | docker secret create <prj_id>_<app_id>.json -
```

Далее, имя секрета (<prj_id>_<app_id>.json) указывается в разделе «secrets» конфигурации деплоя.

Диагностические сообщения (логи) сервис выводит в стандартный вывод (stdout), этот канал захватывается лог-драйвером docker (driver: "json-file") и пишется в стандартные файлы хранения лог-сообщений docker-контейнеров (/var/lib/docker/containers/*/*-json.log). Для сбора данных сообщений и передачи на хост L1 используется Fluent-bit агент, который устанавливается на каждую из swarm-нод (S1/S2/S3).

Ansible-плейбук установки Fluent-bit агента с конфигурацией сбора лог-сообщений docker-контейнеров:

```

---
- name: Fluent-bit agent
  hosts: artlog
  become: yes
  tasks:
    - name: fluentbit | install package key
      apt_key:

```


url: http://packages.fluentbit.io/fluentbit.key
state: present
register: result

- name: fluentbit | install repo

apt_repository:
 repo: deb http://packages.fluentbit.io/ubuntu xenial main
 state: present
register: result
when: result|changed

- name: fluentbit | update repo cache

apt:
 update_cache: yes
when: result|changed

- package:

 name: td-agent-bit
 state: latest

- copy:

 dest: /etc/td-agent-bit/td-agent-bit.conf
 mode: 0644

 content: |

 [SERVICE]

 flush 3
 daemon Off
 log_level warning
 Parsers_File parsers.conf
 http_server Off

 [INPUT]

 Name tail
 Tag docker
 Path /var/lib/docker/containers/*/*-json.log
 DB /var/log/flb-docker.db
 multiline.parser docker
 Buffer_Max_Size 64KB
 Mem_Buf_Limit 256MB
 Skip_Long_Lines On
 Refresh_Interval 10

 [OUTPUT]

 Name forward
 Match *
 Host <IP_L1>
 Port 24224

- name: Start fluentd

systemd:
 name: td-agent-bit
 state: restarted
 enabled: yes

4. Веб-сервер Nginx (хостинг статики фронтального приложения, реверс-прокси для доступа с фронтального приложения к API среднего слоя)

4.1 Настройка системного firewall, минимально необходимый набор правил.

Пример настроек iptables:

```
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp --icmp-type echo-request -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m multiport --dports 80,443 -j ACCEPT
-A INPUT -j REJECT
-A FORWARD -j REJECT
```

4.2 Установка Nginx, сборка из исходников, конфигурирование.

Ansible-плейбук установки Nginx:

```
---
- name: Nginx install
  hosts: gate
  become: yes
  gather_facts: yes
  vars:
    nginx_version: nginx-1.20.2
    nginx_tarball_url: "http://nginx.org/download/{{ nginx_version }}.tar.gz"
    nginx_install_dir: "/tmp/{{ nginx_version }}"
    nginx_sbin_path: "/usr/sbin/nginx"
    nginx_conf_path: "/etc/nginx/nginx.conf"
    nginx_custom_modules: >
      --with-http_ssl_module
      --with-stream
      --with-stream_ssl_preread_module
      --with-http_v2_module
      --with-http_realip_module
      --with-cpu-opt=CPU
      --with-pcre
      --with-pcre-jit
      --with-md5-asm
      --with-sha1-asm
      --with-zlib-asm=CPU
      --with-http_auth_request_module
      --with-http_dav_module
      --with-http_stub_status_module
      --with-ipv6
  tasks:
    - name: Installing NGINX Dependencies
      become: yes
      apt:
        name:
          - gcc
          - libssl-dev
          - make
          - libpcre3-dev
          - zlib1g-dev
```

```
- libxml2-dev
- libxslt-dev
- libgd-dev
- libgeoip-dev
- libperl-dev
update_cache: yes

- name: Downloading NGINX sources
get_url:
  url: "{{ nginx_tarball_url }}"
  dest: "/tmp/{{ nginx_version }}.tar.gz"
register: nginx_source

- name: Unpacking NGINX
unarchive:
  copy: no
  dest: /tmp/
  src: "{{ nginx_source.dest }}"
when: nginx_source.changed
register: nginx_source_unpack

- file:
  state: absent
  path: /etc/nginx/conf.d/
become: yes

- name: Create required Nginx dirs
become: yes
file:
  path: "{{ item }}"
  state: directory
  owner: root
  mode: 0755
loop:
  - /etc/nginx
  - /etc/nginx/conf.d
  - /etc/certs
  - /var/log/nginx

- name: Create a nginx group
become: yes
group:
  name: nginx
  state: present

- name: Create a nginx user
become: yes
user:
  name: nginx
  group: nginx
  state: present
  system: yes
  shell: /sbin/nologin
  createhome: yes
```

```

    home: /var/cache/nginx

- name: Configuring NGINX source with custom modules
  become: yes
  command: "./configure --user=nginx --group=nginx --sbin-path={{ nginx_sbin_path }} --
conf-path={{ nginx_conf_path }} {{ nginx_custom_modules }}"
  args:
    chdir: "{{ nginx_install_dir }}"
    when: nginx_source_unpack is changed
    register: nginx_configure

- name: Installing NGINX
  become: yes
  shell: make && make install
  args:
    chdir: "{{ nginx_install_dir }}"
    when: nginx_configure is changed

- copy:
  dest: /etc/systemd/system/nginx.service
  owner: root
  mode: 0755
  content: |
    [Unit]
    Description=nginx - high performance web server
    Documentation=https://nginx.org/en/docs/
    After=network-online.target remote-fs.target nss-lookup.target
    Wants=network-online.target

    [Service]
    Type=forking
    PIDFile=/var/run/nginx.pid
    ExecStartPre=/usr/sbin/nginx -t -c /etc/nginx/nginx.conf
    ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
    ExecReload=/bin/kill -s HUP $MAINPID
    ExecStop=/bin/kill -s TERM $MAINPID

    [Install]
    WantedBy=multi-user.target
  become: yes

- name: Systemctl daemon-reload
  shell: systemctl daemon-reload

- systemd:
  name: nginx.service
  state: started
  enabled: yes

```

Основная конфигурация /etc/nginx/nginx.conf

```

user nginx;
worker_processes auto;
pid /run/nginx.pid;

```

```

events {
    worker_connections 4096;
    multi_accept on;
}
http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    charset UTF-8;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    log_format timed '$remote_addr $remote_user [$time_local] '
        '"$request" $status $body_bytes_sent '
        '"$http_referer" "$http_user_agent" "$http_x_forwarded_for" '
        '$request_time $upstream_response_time';

    access_log /var/log/nginx/access.log timed;
    error_log /var/log/nginx/error.log info;

    client_max_body_size 50m;

    open_file_cache max=50000 inactive=20s;
    open_file_cache_valid 30s;
    open_file_cache_min_uses 2;
    open_file_cache_errors on;

    proxy_cache_path /tmp/cache levels=1:2 keys_zone=pxycache:16m inactive=2h
max_size=1G;

    # for ws stream
    map $http_upgrade $connection_upgrade {
        default upgrade;
        "" close;
    }

    include nginx-prj-*.conf;
}

```

Конфигурация веб-сервера nginx-prj-<prj_id>.conf:

```

server {
    listen 443 ssl;
    server_name www.company.ru;

    ssl_certificate /etc/nginx/certs/www.company.ru/fullchain.pem;
    ssl_certificate_key /etc/nginx/certs/www.company.ru/privkey.pem;
    ssl_trusted_certificate /etc/nginx/certs/www.company.ru/fullchain.pem;
}

```

```

    ssl_protocols TLSv1.2;
    ssl_ciphers
'kEECDH+ECDSA+AES128:kEECDH+ECDSA+AES256:kEECDH+AES128:kEECDH+A
ES256:+SHA:!aNULL:!eNULL:!LOW:!MD5:!EXP:!DSS:!PSK:!SRP:!kECDH:!CAMELLI
A:!RC4:!SEED';
    ssl_prefer_server_ciphers on;
    ssl_dhparam /etc/nginx/dhparams.pem;
    ssl_session_timeout 1d;
    ssl_session_cache shared:SSL:50m;

    add_header X-Frame-Options ALLOWALL;
    add_header X-Content-Type-Options nosniff;
    add_header Strict-Transport-Security max-age=15768000;
    add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'
'unsafe-eval' https://cdnjs.cloudflare.com https://www.googletagmanager.com
https://www.google-analytics.com; style-src 'self' 'unsafe-inline'; font-src 'self' data:
fonts.gstatic.com; img-src 'self' data: https://www.google-analytics.com;";

    proxy_redirect      off;
    proxy_buffering     off;
    proxy_set_header    Host $host;
    proxy_set_header    X-Url-Scheme $scheme;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    Proxy-host $proxy_host;
    proxy_read_timeout  180s;

    if ($msec ~ "(.*)\.(.*)") {
        set $epoch_millis "$1$2000000";
    }

    location /favicon.ico {
        return 204;
        access_log off;
        log_not_found off;
    }

    if ( $request_method !~ ^(GET|POST|HEAD|OPTIONS)$ ) {
        return 405;
    }
    error_page 405 @error405;
    location @error405 {
        add_header Allow "GET, POST, HEAD, OPTIONS" always;
    }

    error_page 503 /temporarily_unavailable.html;
    location = /temporarily_unavailable.html {
        root /etc/nginx/html;
        internal;
    }
    location = /empty.body {
        return 200 "";
        internal;
    }

```

```

location = /temporarily_unavailable.json {
    root /etc/nginx/html/;
    internal;
}

include inc/<prj_id>_main_*.conf;
}

```

Конфигурация эндпоинтов.

Реверс-прокси API сервисов среднего слоя, inc/<prj_id>_main_front_api.conf:

```

upstream lbmw_api_<prj_id>_main {
    server <IP_S1>:8080;
    server <IP_S2>:8080;
    server <IP_S3>:8080;
}

location ~ ^/api/v1/(.*) {
    proxy_pass lbmw_api_<prj_id>_main/api/v1/$1$is_args$args;
}

location ~ ^/fs/(.*) {
    proxy_cache      pxycache;
    proxy_cache_key  <prj_id>_main$uri$is_args$args;
    proxy_cache_valid 200 301 302 10d;
    proxy_cache_methods GET;
    proxy_ignore_headers Cache-Control;
    proxy_buffering  on;
    proxy_pass lbmw_api_<prj_id>_main/api/v1/fs/$1$is_args$args;
}

location ~ ^/fsupload/(.*) {
    proxy_cache      pxycache;
    proxy_cache_key  <prj_id>_main$uri$is_args$args;
    proxy_cache_valid 200 301 302 10d;
    proxy_cache_methods GET;
    proxy_ignore_headers Cache-Control;
    proxy_buffering  on;
    proxy_pass lbmw_api_<prj_id>_main/api/v1/fs/upload$1$is_args$args;
}

```

Хостинг статики фронтového приложения, inc/<prj_id>_main_front_static.conf

```

location / {
    gzip on;
    gzip_min_length 500;
    gzip_proxied any;
    gzip_comp_level 4;
    gzip_types      text/css text/javascript text/xml text/plain text/x-component
application/javascript application/json application/xml application/rss+xml font/truetype
font/opentype application/vnd.ms-fontobject image/svg+xml;
    gzip_vary on;
    gzip_disable "msie6";
}

```

```
index index.html;
root /etc/nginx/html/<prj_id>/main;
try_files $uri $uri/ /index.html =404;
}
```

Websocket-прокси, inc/<prj_id>_main_ws.conf

```
upstream artesb_<prj_id>_ws {
    server <IP_S1>:20001;
    server <IP_S2>:20001;
    server <IP_S3>:20001;
}

location ~ ^/ws(.*) {
    proxy_pass artesb_<prj_id>_ws$1$is_args$args;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;
    proxy_read_timeout 6000s;
}
```

4.3 Разворачивание кода фронтového приложения

Дистрибуция фронтového приложения выполняется в виде файла-архива с кодом и ресурсами этого приложения, заранее подготовленными (обфускация, минификация, оптимизация) поставщиком ПО.

Полученный архив («статик-пак») необходимо развернуть в целевой каталог, согласно настройке веб-сервера (/etc/nginx/html/<prj_id>/main):

```
$ cd /etc/nginx/html/<prj_id>/main
$ tar -xzf static_pack.tar.gz
```

Приложения

Скрипты сборки docker-образов инфраструктурных компонент из описания:

Следующие сборки производятся с целью фиксации версии и включения результирующих образов в локальный docker-репозиторий.

```
docker.art-bank.ru/redis:1
```

```
FROM redis:5.0.7
```

```
ENV BUILD_VERSION 1
```

```
docker.art-bank.ru/nats:2.2.0
```

```
FROM nats:2.2.0-alpine3.13
```

```
ENV BUILD_VERSION 2.2.0
```

```
docker.art-bank.ru/mongodb:1
```

```
FROM mongo:4.2
```

```
ENV BUILD_VERSION 1
```

```
docker.art-bank.ru/mongodb-bootstrap:1
```

```
FROM stefanprodan/mongo-bootstrap:latest
```

```
ENV BUILD_VERSION 1
```


docker.art-bank.ru/nginx:1

FROM nginx:1.15

ENV BUILD_VERSION 1

7.3.3 Приложение для операционной системы «Android»

Требования к системе:

Операционная система «Android» версии не ниже 7.

1. Установка приложения с помощью «Google Playmarket»:
 - Открыть приложение «Playmarket»;
 - Найти нужное приложение;
 - Нажать "Install".
2. Установка приложения с использованием исполняемого файла с расширением «apk»:
 - Скопировать файл «android.apk» на мобильное устройство, например, в каталог «Downloads»;
 - С помощью файлового менеджера («Файлы», «Проводник», и тому подобные) найти файл и, нажав на него, запустить процесс установки;
 - Ознакомиться со списком запрашиваемых приложением разрешений и подтвердить своё согласие;
 - Запустить установленное приложение.

7.3.4 Приложение для операционной системы «iOS»

Требования к системе:

Операционная система «iOS» версии не ниже 10.

1. Установка приложения с помощью «App Store»:
 - Открыть приложение «App Store»;
 - Найти нужное приложение;
 - Нажать "Install".
2. Установка приложения с использованием архивного файла с расширением «ipa»:
 - Произвести скачивание и инсталляцию последней актуальной версии iTunes;
 - Выполнить скачивание приложения. Формат файла должен быть *.ipa;
 - Дважды кликнув по файлу, открыть его в iTunes. После этого произойдёт добавление в список приложений iTunes;
 - После этого, подключаем устройство к персональному компьютеру и заходим по следующему пути: iTunes — iPhone — «Программы»;
 - Отмечаем нужное приложение и нажимаем «Синхронизировать»;
 - В результате произойдёт успешная установка нужного приложения на устройство;
 - Запустить установленное приложение.

