ООО «БФТ»

129085, г. Москва, ул. Годовикова, д. 9, стр. 17

+7 (495) 784-70-00
ineed@bft.ru
bft.ru

Утвержден
Шифр документа – ЛУ

Комплекс программ «БФТ.Платформа». Функциональное описание

Версия 1.11

Листов 201

© 2026, ООО «БФТ-Холдинг»

АННОТАЦИЯ

Документ описывает назначение, функции и техническую архитектуру платформы быстрой разработки – комплекс программ «БФТ.Платформа», далее – платформа.

Документ предназначен для:

- разработчиков приложений на базе платформы;
- системных и бизнес-аналитиков;
- других специалистов в сфере ИТ, для которых необходимо общее представление о функциональности и структуре платформы.

СОДЕРЖАНИЕ

1. О платформе 10
   1.1. Назначение комплекса программ «БФТ.Платформа» 10
   1.2. Состав комплекса программ «БФТ.Платформа» 10
      1.2.1. Программный комплекс «Базовый состав платформы» 10
      1.2.2. Программный комплекс «Платформа высоконагруженных микросервисных приложений» 14
      1.2.3. Программный комплекс «Сбор, хранение данных и построение отчетности» 14
      1.2.4. Программный комплекс «Геоинформационные сервисы» 16
2. Термины и сокращения 17
3. Описание функциональности программного комплекса «Базовый состав платформы» 19
   3.1. Общие принципы работы 19
   3.2. Программный компонент «Ядро» 19
      3.2.1. Аутентификация пользователей 20
      3.2.2. Авторизация пользователей 20
      3.2.3. Поддержка условий безопасности паролей 21
      3.2.4. Управление учетными записями пользователей 22
      3.2.5. Настройка ролевого доступа 23
      3.2.6. Журналы 26
      3.2.7. Аудит действий пользователя в системе 28
      3.2.8. Настройка структур рубрикатора для пользователей 29
      3.2.9. Автоматическое выполнение функций серверных заданий 30
      3.2.10. Ведение настроечных параметров 31
      3.2.11. Подключение дополнительных баз данных 32
      3.2.12. Организационно-штатная структура (ОШС) 32
      3.2.13. Общие настройки системы 33
         3.2.13.1. Настройка интерфейса системы 33
         3.2.13.2. Ведение классифицирующих меток 33
         3.2.13.3. Консоль администратора 34
         3.2.13.4. Ведение метрик 34
         3.2.13.5. Просмотр свойств среды 34

3.2.13.6. Настройка статусных моделей и групп статусов 34
3.2.13.7. Настройка автоматического запуска задач по времени 34
3.2.13.8. Настройка и хранение наборов атрибутов для иерархических справочников 35
3.2.13.9. Многооконность 36
3.3. Программный компонент «Конфигуратор» 36
3.3.1. Управление моделями данных 36
3.3.1.1. Создание объекта приложения 38
3.3.1.2. Редактирование объекта приложения 56
3.3.1.3. Удаление объекта приложения 57
3.3.1.4. Ведение моделей данных объекта приложения 57
3.3.1.5. Ведение данных объекта приложения 60
3.3.2. Управление сценариями 64
3.3.3. Управление XML схемой 68
3.3.4. Конструктор REST API 68
3.3.5. Управление скриптами 69
3.3.6. Управление импортом и экспортом модели данных 70
3.3.7. Управления репозиторием 72
3.3.8. Самодиагностика 74
3.3.9. Ведение регламентов электронной подписи 75
3.3.10. Производственный календарь 76
3.3.11. Управление бизнес-метриками 76
3.3.12. Генерация последовательностей 76
3.4. Программный компонент «Отчеты» 76
3.4.1. Ведение шаблонов отчетов 77
3.4.2. Ведение отчетов 77
3.4.3. Журнал выполнений отчетов 78
3.5. Программный компонент «Уведомления» 79
3.6. Программный компонент «Бизнес-процессы (BPM)» 80
3.6.1. Управление бизнес-процессами 83
3.6.2. Конструирование схем и моделей данных бизнес-процессов 84
3.6.3. Исполнение бизнес-процессов 87
3.6.3.1. Назначение задач пользователям 87

3.6.4. Администрирование бизнес-процессов ... 88
3.6.4.1. Управление исполнением процессов ... 88
3.6.4.2. Просмотр перечня активных экземпляров процессов ... 89
3.6.4.3. Просмотр истории запуска процессов ... 89
3.6.4.4. Просмотр перечня активных экземпляров процесса при отсутствии исторических данных Camunda ... 89
3.6.4.5. Просмотр статистики всех опубликованных версий бизнес-процессов. ... 90
3.6.4.6. Управление инцидентами. ... 90
3.6.5. Применение диаграмм решений приложения для бизнес-процессов ... 90
3.6.5.1. Настройка описателей диаграмм ... 91
3.6.5.2. Создание диаграммы решений приложения ... 92
3.7. Программный компонент «Отчеты Стимулсофт» ... 92
3.8. Программный компонент «Отчеты BIRT» ... 93
3.9. Программный компонент «Виджеты» ... 93
3.10. Программный компонент «Многофункциональный компонент для работы с табличными данными» ... 95
3.11. Программный компонент «Импорт данных из файлов» ... 97
3.12. Программный компонент «Лицензирование» ... 99
3.13. Программный компонент «Интеграция взаимодействий с другими приложениями» ... 100
3.13.1. Основные понятия интеграции ... 101
3.14. Программный компонент «Управление интеграцией с MDM» ... 102
3.15. Программный компонент «Полнотекстовый поиск» ... 103
3.16. Программный компонент «Сервис электронной подписи» ... 104
3.17. Программный компонент «Машиночитаемые доверенности» ... 105
3.18. Программный компонент «Компонент электронных таблиц» ... 107
3.19. Программный компонент «Конструктор печатных форм» ... 108
3.20. Программный компонент «Обработка больших данных» ... 108
3.20.1. Ведение источников данных ... 109
3.20.2. Управление конвейерами данных ... 110
3.20.3. Управление процессами ... 112
3.20.4. Управление исполнением процессов ... 113

3.20.5. Просмотр перечня результатов валидации данных источников данных 113
3.21. Программный компонент «Мастер-данные» 115
3.22. Программный компонент «Дедупликация» 115
3.23. Программный компонент «Качество данных» 115
3.24. Программный компонент «Редактор документов» 116
3.25. Программный компонент «Хранилище вложений» 117
3.26. Программный компонент «Модуль интеграции взаимодействий с другими приложениями (Модуль интеграции «БФТ.ПИВ»)» 117
3.27. Программный компонент «Исполнение длительных задач» 118
3.28. Программный компонент «Реестр приложений» 119
3.29. Программный компонент «Создание и настройка ИИ-агентов» 119
3.29.1. Особенности реализации 120
3.29.2. Область применения 120
3.29.3. Архитектура 120
3.29.4. Технические особенности реализации 121
3.29.5. Функциональные возможности 121
3.29.6. Конфигурации, реализованные на базе компонента 122
3.29.6.1. Конфигурация ИИ-ассистент 122
3.29.6.2. Конфигурация ПИАФ 123
3.30. Программный компонент «Конфигуратор API» 125
3.31. Программный компонент «Сервер аутентификации по OAuth 2.0» 125
3.32. Возможности интерфейса пользователя 126
4. Описание функциональности программного комплекса «Платформа высоконагруженных микросервисных приложений» 127
4.1. Программный компонент «Базовые библиотеки прикладной разработки» 127
4.2. Программный компонент «Безопасность прикладного программного обеспечения» 130
4.2.1. Сервис authz-server 131
4.2.2. Сервис auth-cache-manager 132
4.2.3. Сервис idm-rest-service 133
4.2.4. Сервис oauth2-proxy 134
4.3. Программный компонент «Криптографические операции» 135

4.3.1. Сервис cert-manager 135
4.4. Программный компонент «Потоковая обработка данных» 136
4.4.1. Коннектор flink-connector-hbase 137
4.4.2. Коннектор flink-connector-jdbc 138
4.4.3. Коннектор flink-connector-opensearch 138
4.4.4. Сервис flink-admin-service 139
4.4.5. Сервис flink-event-processor 139
4.5. Программный компонент «Система управления распределенными базами данных (СУРБД)» 141
4.5.1. Сервис transact-manager 142
4.5.2. Сервис transact-scheduler 143
4.5.3. Сервис history-manager 144
4.5.4. Сервис lock-scheduler 145
4.6. Программный компонент «Шлюз для фронтенда» 145
4.6.1. Сервис graphql-mesh 146
4.6.2. Сервис ui-task-manager 147
4.7. Программный компонент «Сбор информации о сервисах» 148
4.7.1. Сервис noble-registry-service 149
4.8. Программный компонент «Печатные формы» 150
4.8.1. Сервис fo-json-document 151
4.8.2. Сервис fo-printing-apache-fop 151
4.8.3. Сервис fo-printing-form 152
4.8.4. Сервис fo-printing-formatalbum 153
4.8.5. Сервис fo-printing-grpc-client 154
4.8.6. Сервис fo-printing-jasper 154
4.8.7. Сервис fo-printing-weasyprint 154
4.8.8. Сервис fo-service-controller 155
4.9. Программный компонент «Аудит доступа к данным и история изменений» 156
4.9.1. Сервис entity-audit-service 156
4.10. Программный компонент «Компонент обеспечения миграции данных» 157
4.10.1. Библиотека ps-migration-base 157
4.10.2. Сервис ps-migration-data-router 158
4.10.3. Сервис ps-migration-export-metadata 158

4.11. Программный компонент «Фронтенд для веб-клиента» 159
5. Описание функциональности программного комплекса «Сбор, хранение данных и построение отчетности» 161
5.1. Программный компонент «Импорт и преобразование данных» 163
5.2. Программный компонент БФТ.Хранилище 164
5.3. Программный компонент «Администрирование» 166
5.4. Программный компонент «Конфигуратор объектов приложений» 166
5.5. Программный компонент «Сбор и консолидация произвольной информации» 168
5.6. Программный компонент «Формирование регламентированной отчетности (Стимулсофт)» 169
5.7. Программный компонент «Формирование регламентированной отчетности (JXLS)» 170
5.8. Программный компонент «Массовая рассылка отчетов» 171
5.9. Программный компонент «Распределенное выполнение регламентированных отчетов» 172
5.10. Программный компонент «Формирование универсального отчета» 173
5.11. Программный компонент «Предоставление данных внешним системам» 174
5.12. Программный компонент «Базовый набор аналитических отчетов о финансовом состоянии региона (муниципального образования) на основе данных продуктов семейства АЦК с применением Polymatica Analytics» .. 175
5.13. Программный компонент «Базовый набор аналитических отчетов о контрактной деятельности региона на основе данных продуктов семейства АЦК с применением Polymatica Analytics» 179
5.14. Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта Российской Федерации с применением Polymatica Analytics» 179
5.15. Программный компонент «Хранение данных регионального сегмента ЕРН» 182
5.16. Программный компонент «Предоставление данных регионального сегмента ЕРН по подписке и запросам» 183
5.17. Программный компонент «Формирование произвольных запросов к данным регионального сегмента ЕРН» 184
5.18. Программный комплекс «Решение бизнес-анализа Полиматика» 185
5.18.1. Программный компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)» 185

5.18.2. Программный компонент «Конструктор информационных панелей (Polymatica Dashboards)» 187
6. Описание функциональности программного комплекса «Геоинформационные сервисы» 189
6.1. Программный компонент «Картографическое ядро» 190
6.2. Программный компонент «Работа с векторными данными» 191
6.3. Программный компонент «Работа с растровыми данными» 191
6.4. Программный компонент «Сервис геоаналитики для формирования тепловых, кластерных и гексагональных слоев» 191
6.5. Программный компонент «Редактор объектов карты» 191
6.6. Программный компонент «Импорт и экспорт векторных данных» 192
6.7. Программный компонент «Подключение внешних тайловых сервисов» .. 192
6.8. Программный компонент «Кэширование тайловых слоев» 192
6.9. Программный компонент «Интерфейс администратора» 193
6.10. Программный компонент «Интерфейс пользователя» 193
6.11. Программный компонент «Поиск данных» 193
6.12. Программный компонент «Генератор электронных документов» 194
6.13. Программный компонент «Работа с карткомпонентом НСПД» 194
6.14. Программный компонент «Передача векторных данных в НСПД» 194
6.15. Программный компонент «Разбор данных из XML-выписок Росреестра» 194
7. Техническое описание платформы 196
7.1. Общая информация 196
7.2. Архитектура платформы 197
8. Лист регистрации изменений 201

1. О платформе

1.1. Назначение комплекса программ «БФТ.Платформа»

Комплекс программ «БФТ.Платформа» – это платформа быстрой разработки, которая предназначена для создания прикладных приложений средствами администрирования и визуального конструирования.

Платформа ориентирована на создание следующих типов систем:

- Транзакционные системы (OLTP):
  - Системы электронного документооборота.
  - Учетные, финансовые системы (ERP).
- Аналитические системы BI (Business Intelligence).

1.2. Состав комплекса программ «БФТ.Платформа»

Комплекс программ «БФТ.Платформа» состоит из программных комплексов:

- Базовый состав платформы.
- Платформа высоконагруженных микросервисных приложений.
- Сбор, хранение данных и построение отчетности.
- Геоинформационные сервисы (сокращенно «БФТ.Карта»).

1.2.1. Программный комплекс «Базовый состав платформы»

Программный комплекс «Базовый состав платформы» состоит из программных компонентов, Таблица 1.

Таблица 1 – Перечень программных компонентов «Базовый состав платформы»

№	Полное название программного компонента	Сокращенное название	Описание	Статус программного компонента
1	Администрирование	Ядро	Инструмент для управления доступом пользователей и аудит действий в системе.	Стабильный
2	Управление объектной моделью прило-	Конфигуратор	Инструмент для построения интерфейсов без программирования, для создания справочников, документов (со ста-	Стабильный

	жения и специализированные настроечные компоненты для обеспечения работы с объектными моделями приложений (Конфигуратор объектов приложений (ОП)).		тусной моделью) в конфигураторе объектов приложений. Инструмент для ведения данных объектных моделей приложений.	
3	Отчеты		Инструмент для создания отчетных форм с помощью шаблонов Word и Excel.	Стабильный
4	Уведомления		Инструмент для создания, хранения и отображения информационных сообщений для пользователей системы.	Стабильный
5	Управление бизнес-процессами	Бизнес-процессы (BPM)	Инструмент для настройки и исполнения бизнес-процессов, смоделированных в нотации BPMN 2.0 с помощью визуального редактора.	Стабильный
6	Отчеты Стимулсофт		Инструмент для создания отчетных форм с помощью визуального конструктора Стимулсофт.	Стабильный
7	Отчеты BIRT		Инструмент для создания отчетных форм с помощью визуального конструктора BIRT.	Стабильный
8	Представление аналитических данных	Виджеты	Инструмент для представления информации с использованием виджетов на основании сконфигурированной модели данных.	Стабильный
9	Многофункциональный компонент для работы с табличными данными		Инструмент для отображения информации в сетке данных AG Grid.	Стабильный
10	Импорт данных из файлов		Инструмент для выполнения импорта данных из файлов разных форматов.	Стабильный
11	Лицензирование		Инструмент, позволяет ограничивать или запрещать функционал прикладных приложений лицензий. Может существовать несколько экземпляров менеджеров лицензий на разных серверах приложения и каждый из них может генерировать валидные лицензии.	Стабильный
12	Интеграция взаимодей-		Инструмент, позволяет в low-code стиле настраивать обмен сообщениями через Kafka и Kafka Connect.	Стабильный

	ствий с другими приложениями		С его помощью можно описать форматы для входящих и исходящих сообщений, настроить в визуальном редакторе сценарии обработки сообщений.	
13	Управление интеграцией с МДМ		Инструмент для интеграции МДМ (БФТ.ЕНСИ) и Конфигуратора.	Стабильный
14	Полнотекстовый поиск		Инструмент, предназначенный для: - полной индексации справочников; - онлайн-индексации при внесении изменений; - поиска по справочникам; - поиска по содержанию файлов в распространенных форматах (PPT, XLS, DOC, PDF и т.п.).	Стабильный
15	Сервис электронной подписи		Инструмент для обработки запросов от клиентского приложения на формирование и проверку электронной подписи переданных данных.	Стабильный
16	Машиночитаемые доверенности		Инструмент для создания, обработки и хранения электронных машиночитаемых доверенностей, оформленных в структурированном формате.	Стабильный
17	Компонент электронных таблиц	Расчетные таблицы	Инструмент для расчета показателей в объектах приложения «БФТ.Платформы», а также для хранения рассчитанных показателей и применения в дальнейших расчетах.	Стабильный
18	Конструктор печатных форм		Инструмент для создания, редактирования шаблонов печатных форм объектов приложений «БФТ.Платформы».	Технологический предварительный просмотр
19	Обработка больших данных		Инструмент для массовой параллельной обработки данных в среде исполнения Apache Spark: от подключения внешних источников и графической сборки конвейеров загрузки-трансформации-выгрузки до автоматического запуска этих конвейеров в управляемых процессах.	Стабильный
20	Мастер-данные		Инструмент для управления изменениями и версиями записей объектов приложения, обеспечения историчности данных и единственности актуальной (эталонной) версии.	Стабильный
21	Дедупликация		Инструмент для поиска записей-дублей, анализа и обработки дублирующих данных по заранее заданным правилам.	Стабильный

22	Качество данных		Инструмент, обеспечивающий проверку качества данных – асинхронный запуск настроенных пользователем проверок и формирование отчета качества.	Стабильный
23	Редактор документов		Инструмент для интеграции с сервисом OnlyOffice и обеспечения редактирования файлов-вложений напрямую из интерфейса «БФТ.Платформы» без необходимости их скачивания, обработки в отдельных внешних приложениях, и закачивания обратно в платформу.	Стабильный
24	Хранилище вложений		Инструмент для поддержки внешних систем хранения вложений, таких как Архив документов, файлы, хранилище S3, базы данных.	Стабильный
25	Модуль интеграции взаимодействий с другими приложениями (Модуль интеграции «БФТ.ПИВ»)		Выведен из состава платформы в соответствии с приказом Генерального директора.	Устаревший
26	Исполнение длительных задач		Инструмент для управления исполнением задачами.	Стабильный
27	Реестр приложений		Инструмент для создания и настройки приложений на основе «БФТ.Платформы» и отображения связи между приложением и модулями, которыми управляет приложение, а также связи между модулем и источником данных, в котором хранится модуль.	Стабильный
28	Создание и настройка ИИ-агентов		Инструмент для создания, настройки и управления универсальными интеллектуальными агентами, реализующими обработку данных и выполнение прикладных задач на основе технологий обработки естественного языка.	Технологический предварительный просмотр
29	Конфигуратор API		Инструмент для быстрой выгрузки больших объемов записей справочников (созданных пользователем или системных).	Стабильный
30	Сервер аутентификации по OAuth 2.0		Инструмент, позволяющий приложению на основе «БФТ.Платформы» выступать в качестве сервера аутентификации по протоколу OAuth 2.0.	Стабильный

Программные компоненты программного комплекса «Базовый состав платформы» могут поставляться как отдельно, так и в любой комбинации. Состав каждой поставки программных компонент определяется разработчиком (интегратором) самостоятельно, в зависимости от требований и ограничений проекта.

1.2.2. Программный комплекс «Платформа высоконагруженных микросервисных приложений»

Программный комплекс «Платформа высоконагруженных микросервисных приложений» – это набор взаимосвязанных программных компонент, объединённых общей архитектурой и унифицированным API, который позволяет разрабатывать высоконагруженные микросервисные приложения, поддерживающие потоковую обработку данных и работу с комплексом отдельных баз данных, как с единым целым, обеспечивая при этом транзакционную целостность и согласованность данных между ними. Программный комплекс «Платформа высоконагруженных микросервисных приложений» включает следующие программные компоненты:

1. Базовые библиотеки прикладной разработки.
2. Безопасность прикладного программного обеспечения.
3. Криптографические операции.
4. Потоковая обработка данных.
5. Система управления распределенными базами данных.
6. Шлюз для фронтенда.
7. Сбор информации о сервисах.
8. Печатные формы.
9. Аудит доступа к данным и истории изменений.
10. Обеспечение миграции данных.
11. Фронтенд для веб-клиента.

Подробно в разделе 4. Описание функциональности программного комплекса «Платформа высоконагруженных микросервисных приложений».

1.2.3. Программный комплекс «Сбор, хранение данных и построение отчетности»

Программный комплекс «Сбор, хранение данных и построение отчетности» включает возможности организации импорта, преобразования и хранения данных, а также формирования аналитики и отчетности за счет использования различных self-service инструментов.

Программный комплекс «Сбор, хранение данных и построение отчетности» состоит из программного комплекса «Решение бизнес-анализа Полиматика» и программных компонентов:

1. Программный компонент «Импорт и преобразование данных».
2. Программный компонент «БФТ.Хранилище».
3. Программный компонент «Администрирование».
4. Программный компонент «Конфигуратор объектов приложений»
5. Программный компонент «Сбор и консолидация произвольной информации».
6. Компонент «Формирование регламентированной отчетности (СтимулСофт)».
7. Программный компонент «Формирование регламентированной отчетности (JXLS)».
8. Программный компонент «Массовая рассылка отчетов».
9. Программный компонент «Распределенное выполнение регламентированных отчетов».
10. Программный компонент «Формирование универсального отчета».
11. Программный компонент «Предоставление данных внешним системам».
12. Программный компонент «Базовый набор аналитических отчетов о финансовом состоянии региона (муниципального образования) на основе данных продуктов семейства АЦК с применением Polymatica Analytics».
13. Программный компонент «Базовый набор аналитических отчетов о контрактной деятельности региона на основе данных продуктов семейства АЦК с применением Polymatica Analytics».
14. Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта Российской Федерации с применением Polymatica Analytics».
15. Программный компонент «Хранение данных регионального сегмента ЕРН».
16. Программный компонент «Предоставление данных регионального сегмента ЕРН по подписке и запросам».
17. Программный компонент «Формирование произвольных запросов к данным регионального сегмента ЕРН».

Программный комплекс «Решение бизнес-анализа Полиматика» состоит из программных компонентов:

1. Программный компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)».

2. Программный компонент «Конструктор информационных панелей (Polymatica Dashboards)».

Подробно в разделе 5. Описание функциональности программного комплекса «Сбор, хранение данных и построение отчетности».

1.2.4. Программный комплекс «Геоинформационные сервисы»

Программный комплекс «Геоинформационные сервисы» – это набор взаимосвязанных программных компонент, объединённых общей архитектурой и унифицированным API, который позволяет решать задачи, связанные с хранением, обработкой и отображением пространственных данных.

Программный комплекс «Геоинформационные сервисы» включает следующие программные компоненты:

1. Картографическое ядро.
2. Работа с векторными данными.
3. Работа с растровыми данными.
4. Сервис геоаналитики для формирования тепловых, кластерных и гексагональных слоев.
5. Редактор объектов карты.
6. Импорт и экспорт векторных данных.
7. Подключение внешних тайловых сервисов.
8. Кэширование тайловых слоев.
9. Интерфейс администратора.
10. Интерфейс пользователя.
11. Поиск данных.
12. Генератор электронных документов.
13. Работа с карткомпонентом НСПД.
14. Передача векторных данных в НСПД.
15. Разбор данных из XML выписок Росреестра.

2. Термины и сокращения

Термин/сокращение	Описание
BPM	Компонент Business Process Management (управления бизнес-процессами).
CSV	(англ. Comma-Separated Values) – формат представления таблиц с помощью текстовой информации
GEOJSON	Открытый формат хранения географических структур данных, основанный на JSON
GPKG (GeoPackage)	Открытый, платформонезависимый формат для хранения и обмена геопространственными данными
JSON	(англ. JavaScript Object Notation – нотации объекта JavaScript) – текстовый формат обмена данными, основанный на JavaScript
MDM	Централизованная система ведения справочников (классификаторов) и реестров.
MVT	(англ. Model-View-Template) – архитектурный шаблон, который используют при создании веб-приложений
WMS	(англ. Web Map Service) – картографический веб-сервис
WMTS	(англ. Web Map Tile Service) – сервис тайлов веб-карты
XML	(англ. eXtensible Markup Language) – язык разметки
XSD	(англ. XML Schema Definition) – язык, который описывает структуру XML-документов
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
БД	База данных.
БФТ.Платформа	Платформа быстрой разработки.
Конфигуратор	Конфигуратор объектов приложения – интерфейс для управления объектной моделью приложений.
ФГИС ЕЦП НСПД, НСПД	Федеральная государственная географическая информационная система «Единая цифровая платформа «Национальная система пространственных данных»
ОП	Объект приложения.
СУБД	Система управления базой данных.

Тайл	(от англ. tile – плитка) – небольшое изображение фиксированного размера, служащее фрагментом большего изображения

3. Описание функциональности программного комплекса «Базовый состав платформы»

3.1. Общие принципы работы

Программный комплекс «Базовый состав платформы» обеспечивает следующие принципы работы:

- Автоматическое обновление данных на всех серверах приложений, работающих параллельно с единой базой данных, при изменении кешированных данных на одном из серверов приложений;
- Управление (включение/отключение) функцией снятия скриншота экрана пользователя при возникновении ошибки.

3.2. Программный компонент «Ядро»

Программный компонент «Ядро» позволяет выполнять следующие функции:

- Аутентификация пользователей.
- Авторизация пользователей.
- Поддержка условий безопасности паролей.
- Управление учетными записями пользователей.
- Настройка ролевого доступа.
- Аудит.
- Настройка структур рубрикатора для пользователей.
- Автоматическое выполнение серверных заданий.
- Сервис нотификаций.
- Ведение настроечных параметров.
- Настройка интерфейса системы.
- Ведение классифицирующих меток.
- Ведение индексов полнотекстового поиска.
- Консоль администратора.
- Просмотр свойств среды.
- Ведение метрик.
- Проверку наличия необходимых настроек при старте приложения.

3.2.1. Аутентификация пользователей

Аутентификация пользователя включает функции:

- Ввод пользователем логина и пароля.
- Запоминание и восстановления пароля.
- Аутентификация пользователя путем ввода логина и пароля осуществляется способами:
  - Штатной аутентификации:
    - Базовая через ввод пользователем Логина и Пароля своей учетной записи.
    - По сертификату через ввод пользователем номера сертификата электронной подписи.
      - Отображение только активных сертификатов при аутентификации.
    - С возможностью запоминания и восстановления пароля.
  - Через сервер безопасности:
    - Ввод пользователем логина учетной записи пользователя в системе сервера безопасности.
    - Ввод пользователем пароля для входа в систему сервера безопасности.
    - С возможностью запоминания и восстановления пароля.

3.2.2. Авторизация пользователей

Авторизация пользователя включает функции:

- Идентификация пользователя системой по логину и паролю:
  - Вывод логина заблокированного пользователя и IP-адреса, с которого пришел запрос на авторизацию при получении ошибки "401 Unauthorized".
- Восстановление пароля через почтовый адрес пользователя:
  - Если пользователь заблокирован в «БФТ.Платформе», то при восстановлении пароля выводится ошибка : «HTTP error: 403»
- Авторизация по LDAP путем синхронизации с Active Directory:
  - Аутентификация в Active Directory.
  - Создание нового пользователя, если ранее его не было.
  - Блокировка пользователя, если учетная запись заблокирована в домене (если учетная запись существовала в «БФТ.Платформе»).
  - Отказ создания пользователя, если учетная запись заблокирована в домене.

  - Авторизация в системе.
  - Авторизация с помощью свойства security.server.ldapUserPolicy с политикой безопасности по умолчанию. Если для свойства security.server.ldapUserPolicy не указана политика безопасности, то при авторизации По LDAP назначается политика безопасности DEFAULT.
  - Недоступность изменения данных в профиле пользователя: Фамилия, Имя, Отчество, почта, телефон.
- Проверка полномочий пользователя по работе с системой.
- Отображения окна предупреждения при вводе неверных данных.
- Выбор вкладки авторизации по умолчанию.
- Выполнение авторизации только через LDAP.

3.2.3. Поддержка условий безопасности паролей

Поддержка условий безопасности паролей включает функции:

- Настройка правил проверки паролей:
  - Срок действия пароля.
  - Наличие пароля (не «пустой»).
  - Количество неудачных попыток ввода пароля.
  - Срок временной блокировки, мин.
  - Наличие в пароле букв, цифр или спецсимволов.
  - Минимальная длина пароля (от 16 символов).
  - Блокировать пользователей, неактивных в течение, дн.
  - Контролировать количество уникальных паролей.
  - Несовпадение логина и пароля.
  - Использование «слепых» паролей:
    - При авторизации пользователей. При наборе пароля его символы заменяются на экране спецсимволами.
    - При отображении учетных записей пользователей в каталоге. Символы паролей заменяются спецсимволами, количество отображаемых символов не соответствует длине пароля.
  - Хранение паролей в зашифрованном виде.
  - Установка сроков действия паролей.
  - Установка двухфакторной аутентификации.
    - Количество неудачных попыток ввода одноразового кода.

- Установка списка политик безопасности в application.properties через настройку ice.security.allowed-policies, владельцам которых разрешен вход в «БФТ.Платформу».

3.2.4. Управление учетными записями пользователей

Управление учетными записями пользователей включает функции:

- Создание учетной записи.
- Просмотр учетной записи.
- Изменение учетной записи.
- Удаление учетной записи.
- Отображение организационно-штатной структуры учетной записи.
- Копирование учетной записи пользователя:
  - Копирование значений полей «Политика безопасности», «Разрешенные типы аутентификации», полей с вкладки «Настроечные параметры», «Роли».
  - Отсутствие копирования ролей, в свойствах которых в поле Действует по указаны дата и время ранее даты и времени при копировании;
  - Копирование значений настроечных параметров с включенным признаком Пользовательский, в том числе с включенным признаком Доступно для изменения только Администратору.
- Блокировка учетной записи:
  - Ручная, с указанием причины и типом блокировки:
    - Постоянная.
    - Временная.
  - Автоматическая, при исчерпании заданного количества попыток ввода пароля.
- Управление учетными записями пользователей включает также возможность указать следующие учетные данные пользователя:
  - Логин;
  - Фамилия;
  - Имя;
  - Отчество;
  - Эл. почта;
  - Получать электронные письма;
  - Телефон;
  - Получать смс;

- Политика безопасности;
- Пароль;
- Подтвердите пароль;
- Потребовать смену пароля;
- Установить контроль на количество уникальных паролей;
- Установить проверку на наличие в пароле букв разных регистров, цифр и спец-символов;
- Сертификат для логина;
- Разрешенные типы аутентификации;
- Роли:
  - Назначение ролей пользователю:
    - Действует с.
    - Действует по.
    - Описание.
    - Группа ролей.
  - Отзыв роли у пользователя (Установка до какой даты действует роль);
  - Отображение информации о назначенной роли в профиле пользователя.
- Примечание;
- Разрешена LDAP-аутентификация;
- Заблокирован;
- Причина блокировки;
- Добавление настроечного параметра для пользователей.

- Просмотр МЧД, связанных с пользователем.
- Просмотр и назначение ОШС, связанных с пользователем.
- Смена ОШС, к которой относится пользователь.

3.2.5. Настройка ролевого доступа

Настройка ролевого доступа позволяет осуществить:

- защиту информации от несанкционированного доступа;
- защиту от ошибочных действий пользователей;
- соблюдение регламентных процедур.

Настройка ролевого доступа обеспечивает выполнение следующих функций:

- ведение списка ролей пользователей;
- создание новых ролей пользователей;

- назначение ролей пользователям (в том числе на период);
- использование ранее настроенных ролей пользователей, как шаблонов для создания новых ролей;
- изменения ролей пользователей:
  - удаление привилегий;
  - присвоение новых привилегий;
  - изменение правил доступа к объектам данных.
- удаления ролей пользователей;
- поиска ролей в реестре ролей пользователей;
- выгрузка списка ролей для сервера безопасности;
- объединение ролей в одну группу.
- настройку условий отображения ролей по лицензионным флагам и модулям.

Ролевая модель предоставляет возможность назначения пользователям:

- разрешений на выполнение отдельных функций;
- ограничений доступа к информации.

Для выбранной роли осуществляется настройка прав доступа:

- к разделам меню (или объектам приложения) системы;
- к массиву записей (данным) объекта приложения;
- к полям записей объекта приложения (полям на форме редактирования и колонкам на форме списка);
- к процессам приложения, в том числе к просмотру процессов приложения без доступа к изменению, удалению, запуску бизнес-процессов приложения;
- к экземплярам процессов приложения с помощью сегментов доступа с разграничением по модулям и процессам приложения;
- к возможности конфигурирования объектов приложения, сценариев, скриптов.

Система позволяет ограничивать доступ к объектам приложения с помощью сегментов доступа, имеющих тип:

- по группе настроечного параметра;
- по записи;
- по модулю приложения;
- по группе объекта приложения;
- по группе сценария;

- по группе скрипта приложения;
- по процессу приложения;
- по модулю процесса приложения.

Система позволяет ограничивать доступ к функциям системы:

- создания объектов данных;
- редактирования объектов данных;
- удаления объектов данных;
- действия над объектами данных в соответствии с регламентными полномочиями.

В системе существуют системные роли:

- Супер пользователь: предназначена для отладки программного обеспечения. Даёт полный доступ ко всем объектам системы.
- Администратор системы: предназначена для администрирования системных сервисов: управление пользователями, ролями, системным расписанием и т.п.
- Администратор репозитория конфигураций – предназначена для работы с патчами конфигурации.
- Базовая – данная роль даёт базовые возможности пользователю в системе:
  - Вход в систему.
  - Доступ на чтение и изменение своих данных в Профиле пользователя.
  - Доступ на чтение раздела Рабочие панели.
  - Доступ к уведомлениям и вложениям в них:
    - Полный доступ к разделу **Журнал сообщений**, чтобы совершать действия с **Лентой уведомлений** – отмечать прочитанным, очищать.
  - Если одновременно с ролью **Базовая** назначить пользователю бизнес-роль, дающую доступ к **Конфигуратору** – **Чтение, Объекты приложения – Полный** (для конкретного модуля), то у пользователя появляются возможности для работы со справочниками:
    - Создание записи по шаблону – возможность создания записи из шаблона.
    - Работа с формами просмотра данных – возможность полного доступа на вспомогательные инструменты для работы с данными:
      - Настройка отображаемых колонок и их сохранение.
      - Настройка фильтров и их сохранение.
  - Работа с объектами приложения со статусной моделью – чтение всей настроек из подраздела Статусы и переходы, чтобы читать записи объекта приложения со статусной моделью.

- Доступ на чтение раздела МЧД: Код системы, Полномочия, Ограничения полномочий, Значения ограничений полномочий, Подпись – все записи; Реестр МЧД ограничен записями, где пользователь является представителем; Участники ограничен записями, где доверители выдали текущему пользователю МЧД или представители связаны с текущим пользователем; Доверитель ограничен записями, где доверитель выдал текущему пользователю МЧД; Представитель ограничен записями, где представитель связан с текущим пользователем.

- Доверитель: предназначена для работы с модулем МЧД (mchd-v2). Дополнительно роль Доверитель включает в себя настройки доступа из роли Базовая. Данная роль предоставляет пользователю доступ к разделу МЧД:
  - Полный доступ на объекты: Участник, Доверитель, Представитель, Подпись, Полномочия, Ограничения полномочий, Значения ограничений полномочий.
  - Полный доступ к разделу Реестр МЧД (ограничен записями, где организация пользователя является доверителем или сам пользователь является представителем).
  - Только на чтение раздел Код системы.
- Роль для импорта конфигурации предназначена для выполнения импорта конфигурации.

3.2.6. Журналы

Аудит системы обеспечивает выполнение следующих функций:

- Автоматическая фиксация изменения записей объектов приложения:
  - изменения значений полей с отображением старых и новых значений;
  - установка связи между объектами приложения (электронными документами) системы;
  - присоединение вложений;
  - изменения состояния (статуса) объектов приложения (электронных документов) системы.
- Просмотр перечня изменений:
  - для каждой записи объекта приложения (из его формы);
  - для всех объектов системы.
- Автоматическая фиксация действий и методов, выполняемых пользователем над объектами приложения:
  - стандартные: «Создание», «Чтение», «Просмотр», «Изменение», «Удаление»;
  - прочие, в зависимости от установленных функций для объекта приложения.

- Дополнительные условия:
  - фиксируется и успешное, и неуспешное выполнение действий;
  - не фиксируются действия закрытия форм объектов приложения;
  - возможность выключения сохранения действий пользователя для конфигурации объекта приложения, если на вкладке "Доп. свойства" указать расширенное свойство disabled_writing_timeline со значением true.
- История изменений элементов:
  - Выбирать поле конфигурации объекта приложения (вкладку конфигурации), по которому необходимо просмотреть изменения:
    - Раздел Общие:
      - Правила.
      - Валидация.
      - Форма редактирования.
      - Триггеры.
      - Действия.
      - Атрибуты.
      - Имя таблицы.
      - Представление.
      - Кэшируемый.
      - Аудит.
      - Наименование ОП.
      - Группа ОП.
      - Статусная модель.
      - Родительский ОП.
    - Раздел Отображаемое имя:
      - Поля отображаемого имени.
      - Скрипт отображаемого имени.
      - Поля подсказки.
    - Раздел Отчеты:
      - Шаблон ответа.
      - Форматы экспорта.
    - Раздел Форма списка:
      - Колонки формы списка.
      - Фильтры.
      - Вид отображения.

        - Вид списка.
        - Скрипт формы списка.
        - Разворачивать узлы фильтрации.
        - Ссылка на родителя.
    - Отображать дату и время, в которые внесены изменения.
    - Отображать ФИО пользователя, который внёс изменения.
    - Отображать список элементов выбранного поля (вкладки), которые претерпели изменения с цветовым обозначением:
        - Голубые элементы – изменены.
        - Зелёные элементы – добавлены.
        - Красные элементы – удалены.
    - Открывать элементы из списка для просмотра их старого и нового состояния.
    - Показывать во всплывающей подсказке Отображаемое имя и список изменен-ных свойств элемента при наведении на него курсора в окне История изменений элементов.
- Журнал перевода по статусам:
    - Регистрации действий пользователей по переводу записей справочников со статуса на статус. В Журнале перевода по статусам фиксируется информация:
        - Время и дата начала и окончания перехода по статусу;
        - Логин и ФИО пользователя, инициировавшего переход по статусу;
        - Наименование начального и конечного статусов;
        - Наименование перехода между статусами.
    - Скрывать или отображать записи со статусом с включенным признаком Ветвле-ние.
    - Настраивать сроки хранения записей в журнале.

3.2.7. Аудит действий пользователя в системе

Журнал событий:

- Автоматическая регистрация действий пользователей в системе;
- Возможность включения/выключения регистрации действий пользователей с помощью признака **Включить аудит** на вкладке **Свойства** объекта приложения.
- Просмотр информации о группе, относящейся к событию аудита

Типы событий аудита:

- Ведение действий пользователей в системе (типов событий);

- Возможность включать/отключать фиксацию типов событий в **Журнале событий**, с помощью признака **Журналируемый**;
- Возможность задавать параметры для фиксации их значений;
- Определение уровня критичности события аудита;
- Ведение групп событий действий пользователя в системе.

Группы событий:

- Ведение групп типов событий аудита (действий пользователя).

3.2.8. Настройка структур рубрикатора для пользователей

Настройка структур рубрикатора включает выполнение функций:

- Настройка групп элементов рубрикатора.
- Объединение групп элементов рубрикатора в строки.
- Настройка элементов, входящих в группы рубрикатора.
- Копирование групп элементов и элементов между рубрикаторами.
- Указание вызываемых объектов приложения для элементов рубрикатора.
- Указание значения параметров URL адреса для элемента рубрикатора с видом Ссылка.
- Настройка иконок для групп или элементов рубрикатора.
- Настройка цвета иконок для элементов рубрикатора.
- Настройка видимости отдельных пунктов меню (групп) или объектов приложения (элементов) в рамках роли.
- Настройка вида меню полный выпадающий список, только элементы верхнего уровня или полное меню с навигацией.
- Отображение иконок для пунктов меню.
- Назначение структур рубрикатора пользователям.
- Настройка пользовательского рубрикатора, создаваемого пользователем (настраиваются пункты меню группы разных уровней вложенности):
  - Настройка всплывающей подсказки с полным наименованием объекта приложения при наведении на него курсора.
- Возможность использования базового рубрикатора для работы с полным функционалом системы.
- Возможность регулирования ширины навигационной панели.

- Возможность скрытия и отображения навигационной панели на кнопке.
- Динамический поиск в навигационной панели. Результаты поиска обновляются при изменении введенных символов в строке поиска, нажатие клавиши "enter" не требуется.
- Выполнение поиска по системному или отображаемому имени рубрикатора.
- Выполнение сортировки записей в системном справочнике «Рубрикатор» по полям системного и отображаемого имени рубрикатора.
- Использование системных форм для отображения пунктов базового рубрикатора.
- Задавать CSS стили в меню рубрикатора.
- Создавать условия для скрытия пунктов меню рубрикатора через скрипт или редактор условий.
- Совершать поиск по элементам меню и групп.
- Автоматическое присваивание иконок с динамическим изменением для групп и элементов.
- Возможность вывода в пользовательский рубрикатор системных объектов приложения с настройкой параметров отображения формы.
- Возможность запоминать режим отображение данных в навигаторе, установленный пользователем.
- Возможность сформировать документацию по рубрикатору.
- Возможность создания кнопок в меню системных справочников.
- Генерация файла документации с конфигурацией пользовательского рубрикатора.
- Возможность добавлять группы пользовательского рубрикатора в раздел Избранное.

3.2.9. Автоматическое выполнение функций серверных заданий

Программный компонент обеспечивает автоматическое выполнение системных задач по заданному графику и хранение информации о правилах их выполнения:

- Ведение перечня задач.
- Настройка параметров выполнения задач.
- Автоматический запуск задач по графику.
- Отмена выполнения запущенной задачи.
- Просмотр информации по задачам:
  - Перечень экземпляров задач:

  - Всех выполняемых в текущий момент и выполненных автоматических задач.
  - Для каждой выполненной и выполняемой задачи.
- Перечень экземпляров задач содержит информацию:
  - Дата и время старта и завершения задачи;
  - Результат выполнения задачи.

- Выполнение системного задания по записи события «Вход в систему» в таблицу базы данных.
- Возможность включения и отключение нескольких записей планировщика заданий.
- Возможность запуска заданий планировщика на определенных серверах при горизонтальном масштабировании.

3.2.10. Ведение настроечных параметров

Ведение настроечных параметров необходимо для настройки пользовательского интерфейса, или для использования в скриптах, или для других целей. В связи с этим, настроечные параметры могут обладать разными признаками:

- пользовательские – значения параметров устанавливаются для каждого пользователя индивидуально;
- доступно для изменения только администратору – значение параметра устанавливается/редактируется только пользователем с ролью Администратор системы или Суперпользователь;
- контекстные – значения параметров устанавливаются для определенного контекста;
- темпорированный – значения параметров имеют период действия;
- шифрованный – значения параметров сохраняются в зашифрованном виде.

Ведение настроечных параметров включает функции:

- настройка подсказки параметра;
- настройка контекстных параметров;
- настройка множественного выбора значений параметра;
- настройка множественного выбора значений параметра из пользовательского справочника;
- настройка пользовательских параметров;
- настройка темпорированных параметров;
- настройка шифрованных параметров;

- настройка обязательных параметров;
- копирование параметров;
- настройка отображения параметров по лицензионным ограничениям.

Настройка параметров:

- одному пользователю;
- выбранной группе пользователей;
- одновременно всем пользователям системы;
- выдача доступа изменения системных настроечных параметров через ролевой доступ;
- использование хранилища значений для загрузки данных в настоечные параметры:
    - локальное хранилище браузера,
    - системные параметры,
    - свойства приложения.

3.2.11. Подключение дополнительных баз данных

Подключение дополнительных баз данных включает функции:

- Ведение перечня дополнительных источников данных.
- Включение и отключение доступа к источнику данных.
- Управление параметрами подключения к источнику данных:
  - Jdbc url.
  - Пользователь.
  - Пароль.
  - Класс провайдера данных.
- Управление приоритетами источников данных
- Проверка подключения источников данных

3.2.12. Организационно-штатная структура (ОШС)

Ведение организационно-штатной структуры организации или организаций, которая может быть применена для разграничения доступа по записям справочника. Ведение ОШС обеспечивает возможности:

- Создание организационно-штатной структуры.
- Разграничение доступа к записям справочников с использованием ОШС.
- Отображение информации о пользователях, для которых применяется ограничение доступа ОШС.

- Импорт типового файла, сформированного в 1С в формате xml в справочник Администрирование → ОШС.

3.2.13. Общие настройки системы

Для обеспечения функционирования системы в целом предусмотрены следующие настройки:

- Настройка интерфейса системы – оформление интерфейса системы в определенном стиле – цветовой гамме окон, таблиц, полей и их заголовков, размеров и видов шрифтов и других параметров. Применение оформления индивидуально к каждому пользователю или по умолчанию.
- Ведение классифицирующих меток.
- Возможность управления приложением в консоли администратора.
- Ведение метрик.
- Просмотр свойств среды.

3.2.13.1. Настройка интерфейса системы

Настройка интерфейса системы осуществляется с помощью ведения тем и включает функции:

- Создание, настройка и удаление темы оформления.
- Применение базовой и других тем.
- Изменение параметров интерфейса:
  - цвет главного меню;
  - цвет кнопок и ссылок;
  - цвет текста;
  - цвет главных разделителей;
  - радиус скругления границ;
  - цвет фона.
- Создание и настройка дополнительных параметров темы.
- Применение темы по умолчанию для всех пользователей системы.
- Применение темы индивидуально для выбранного пользователя через использования настроечных параметров.
- Применение встроенных цветовых палитр.

3.2.13.2. Ведение классифицирующих меток

Программный компонент обеспечивает ведение классифицирующих меток, позволяющих разделять атрибуты объектов приложений на классы, для осуществления некоторых действия только над атрибутами, принадлежащими к определенному классу.

3.2.13.3. Консоль администратора

Консоль администратора обеспечивает выполнение функций:

- создание и запуск скриптов из интерфейса системы;
- удаление результатов выполнения скрипта;
- сохранение результатов запроса в файле на диске.

3.2.13.4. Ведение метрик

Ведение метрик обеспечивает выполнение функций:

- получение численного значения некоторого свойства программного обеспечения или его спецификаций;
- просмотр применяющихся метрик: Название, Теги, Статистика, Значение.

3.2.13.5. Просмотр свойств среды

Просмотр свойств среды обеспечивает просмотр перечня свойств платформы из файла Application.properties. Параметры – Источник, Свойство, Значение.

3.2.13.6. Настройка статусных моделей и групп статусов

Настройка статусных моделей и групп статусов обеспечивает выполнение функций:

- создание и изменение групп статусов для цветового разделения;
- создание статусных моделей для перемещения записей по статусам;
- создание описания статусных моделей;
- создание и изменение переходов между статусами;
- включение запроса подтверждения у пользователя для выполнения перехода;
- аудит переходов записи по статусам;
- просмотр графического представления статусной модели;
- удаление групп статусов и статусных моделей;
- формирование документации для статусной модели;
- отображение свойств статусов и переходов в виде иконок-индикаторов.

3.2.13.7. Настройка автоматического запуска задач по времени

Настройка автоматического запуска задач по времени позволяет:

- Вести перечень задач для осуществления их автоматического запуска;
    - Создание и редактирование заданий планировщика.
    - Настройка параметров выполнения задания:
        - наименование и описание задачи;
        - признак активности (вкл/выкл);
        - настройка крон-выражения;
        - выбор задания из перечня доступных в системе;
            - настройка индивидуальных параметров, требующихся для каждого задания.
- Запускать задания на выполнение;
- Запуск и отключение нескольких выделенных заданий;
- Отменить выполнения запущенной задачи;
- Удалить задания планировщика;
- Выполнять просмотр истории запусков заданий планировщика;
- Выполнять детальный просмотр отдельного запуска:
    - время запуска;
    - время завершения;
    - продолжительность;
    - состояние задания;
    - используемый сервис;
    - используемый метод;
    - состояние последней задачи.

3.2.13.8. Настройка и хранение наборов атрибутов для иерархических справочников

- Создание и редактирование характеристики через справочник Характеристики.
    - При создании характеристики доступна нумерация полей.
- Создание и редактирование характеристики через форму редактирования записи иерархического справочника.
- Добавление характеристики в запись иерархического справочника.
- Удаление характеристики из записи иерархического справочника.
- Настройка полей характеристик.
- Работа с атрибутом характеристики.

3.2.13.9. Многооконность

Упрощение и ускорение работы с несколькими записями справочника и списками записей объектов приложений в рамках одной вкладки браузера.

Режим многооконности может быть включен только в пользовательском рубрикаторе и позволяет работать с несколькими записями объектов приложения одновременно в рамках одной вкладки браузера. Использование режима многооконности значительно ускоряет работу в сложных конфигурациях, избавляя от необходимости открывать множество вкладок браузера и сокращая время ожидания загрузки интерфейса.

3.3. Программный компонент «Конфигуратор»

3.3.1. Управление моделями данных

Управление объектными моделями приложений и данных выполняется в Конфигураторе объектов приложений (далее Конфигуратор) и заключается в обеспечении добавления, изменения, удаления и настройки структур объектов приложения (конфигураций), таких как, например, справочники, документы, реестры.

Управление объектами данных осуществляется через пользовательский интерфейс (Конфигуратор).

Конфигуратор является инструментом для проектирования систем с объектами приложений «на лету», без привлечения разработчиков и предназначен как для пользователей, так и для администраторов (аналитиков). Администраторы системы (аналитики) проектируют и настраивают объекты приложения системы, а пользователи работают с данными настроенных объектов приложения системы. Разработчики могут быть привлечены для решения редких и частных задач, которые невозможно выполнить с помощью методов и средств Конфигуратора.

Изменения, внесенные в объектную модель приложения через Конфигуратор, отражаются в системе без перезапуска сервера приложения.

Объекты данных, подлежащие управлению и ведению:

- Объекты, для которых не предусмотрен жизненный цикл (рабочий процесс, статусная модель), например, справочники;
- Объекты, для которых предусмотрен жизненный цикл, например, документы со статусной модель.

Ведение объектных моделей приложений и данных включает функции:

- Создание объекта приложения;
- Редактирование созданного объекта приложения;

- Удаление созданного объекта приложения;
- Загрузка модели данных объекта приложения из файла;
- Сохранение модели данных в файл;
- Добавление в Избранное моделей данных объектов приложения:
  - возможность отображения всех моделей данных;
  - возможность отображения только избранных моделей данных;
  - возможность удаления из избранных;
- Обновление перечня моделей данных объектов приложения;
- Создание копии модели данных объекта приложения;
- Построение ER-диаграммы модели данных объекта приложения:
  - возможность просмотра и скачивания ER-диаграммы модели данных объекта приложения, модулей и групп объектов приложений;
- Просмотр таблицы базы данных модели данных объекта приложения:
  - свойств таблицы базы данных;
  - отображение колонок и их свойств для таблицы базы данных;
  - отображение индексов колонок таблицы;
  - ведение политик RLS;
  - отображение объектов приложения;
- Просмотр связей атрибутов справочника:
  - со списочной формой справочника;
  - с отображаемым именем;
  - с формой редактирования справочника;
  - с правилами поведения элементов управления формы редактирования;
  - создание статических представлений модели данных объекта приложения;
  - просмотр истории изменений элементов:
  - выбор поля конфигурации объекта приложения, по которому необходимо просмотреть историю изменений;
  - отображение даты и времени, когда внесены изменения;
  - отображение ФИО пользователей, которые внесли изменения;
  - отображение списка элементов, которые претерпели изменения с цветовым обозначением;
  - просмотр старого и нового состояния для каждого элемента из списка элементов.
  - просмотр взаимосвязей элементов конфигурации:

  - отображать идентификатор связанного объекта конфигурации;
  - отображать наименование связанного объекта конфигурации;
  - отображать тип связанного объекта конфигурации;
  - отображать информацию о поле объекта конфигурации, связанного с текущим объектом приложения;
  - возможность открыть связанный объект в отдельной вкладке браузера.
- Автоматическая генерация документации модулей и объектов приложений:
  - полная документация;
  - описание логической модели;
  - описание статусной модели;
  - описание форм редактирования;
  - описание правил валидации;
  - проверка конфигурации объекта приложения;
  - просмотр истории изменений модели данных объектной модели приложения;
- Создание и настройка статусной модели для объектной модели приложения:
  - просмотр графического представления статусной модели;
- Добавление, изменение и удаление статуса в модели:
  - выбор группы для статуса;
  - поиск статусной модели по полю «Наименование», «Код»;
  - установление признаков Ветвление, Конечный, Начальный;
  - перемещение статусов между собой;
- Добавление перехода для статуса с возможностью указания:
  - Кода, Названия, Конечного состояния;
  - перехода по умолчанию;
  - ролей для обеспечения выполнения перехода;
  - видимости перехода;
  - условий видимости перехода с помощью скрипта;
- Действия для перехода:
  - с помощью скрипта или серверного задания;
  - использование системных настроечных параметров:
  - для автоматического предоставления доступа к справочнику для пользователя;
  - восстановления фильтров, настроенных по условиям и колонкам формы списка, при повторном открытии формы списка.

3.3.1.1. Создание объекта приложения

Для создания нового объекта приложения программный компонент обеспечивает следующие возможности:

- Описание основных свойств создаваемого объекта приложения.
- Описание модели данных создаваемого объекта приложения.
- Описание уникальных атрибутов создаваемого объекта приложения.
- Описание формы редактирования создаваемого объекта приложения.
- Описание формы списка создаваемого объекта приложения.
- Настройка рабочего процесса (статусной модели) для объекта приложения.
- Описание условий форматирования формы списка создаваемого объекта приложения.
- Описание фильтров формы списка создаваемого объекта приложения.
- Описание печатных форм создаваемого объекта приложения.
- Описание правил валидации модели данных создаваемого объекта приложения.
- Описание действий модели данных создаваемого объекта приложения.
- Описание триггеров модели данных создаваемого объекта приложения.
- Описание правил проведения элементов управления формы редактирования.
- Настройка подписываемых данных объектной модели данных.
- Просмотр json-представления модели данных создаваемого объекта приложения.

При сохранении пользователем объекта приложения в Конфигураторе система автоматически генерирует объект приложения в соответствии с заданным описанием модели данных, формы редактирования и списка. Также возможна генерация формы редактирования и формы списка объекта приложения на основании заданной модели данных в случае, если форма редактирования и списка не описаны через Конфигуратор.

При сохранении и публикации изменений в объектах приложения с типами Справочник, Реестр, Пользовательская форма, Отчет и Представление пользователям, у которых открыт измененный объект приложения в разделе Справочники на момент изменения конфигурации, отображается пуш-уведомление о внесении изменений и автоматически обновляется страница в браузере. По умолчанию перезагрузка страницы отключена. Включается в прикладном проекте в блоке инициализации основного объекта web-компонента приложения.

В соответствии с заданной моделью данных осуществляется форматно-логический контроль, а именно:

- Контроль типа и размерности поля (в зависимости от типа поля: максимальная и минимальная длина, точное значение длины, максимальное количество знаков, допустимое количество знаков до и после запятой и т.д.);
- Контроль обязательности заполнения и уникальности значений полей;

- Контроль на соответствие значения поля заданному регулярному выражению. Контроль на соответствие значения поля заданной маске ввода. Произвольный контроль, задаваемый с помощью скриптового языка для полей или объекта в целом.

3.3.1.1.1. Описание основных свойств объекта приложения

Программный компонент обеспечивает возможность описания основных свойств объекта приложения, а именно:

- Выбор типа объекта приложения из вариантов:
  - Справочник.
  - Реестр.
  - Пользовательская форма.
  - Отчет.
  - Представление.
  - Сообщение.
- Выбор статусной модели для объекта приложения.
  - Выбор типа статусной модели: управляющая состоянием или отображающая состояние.
- Указание имени таблицы данных объекта приложения, которая будет отображена в базе данных.
- Блокировка переименования имени таблицы базы данных после сохранения.
- Управление наследованием модели данных родительского объекта приложения:
  - Список атрибутов.
  - Отображаемое имя.
  - Настройки колонок формы списка.
  - Фильтры.
  - Настройки формы редактирования.
  - Печатные формы.
  - Правила валидации.
  - Триггеры.
  - Действия.
  - Правила поведения элементов формы редактирования.
  - Подписываемые данные
- Управление наследованием модели данных объекта приложения другими объектами.
- Чтение данных из представления таблицы базы данных.
- Создание базового объекта приложения.

- Управление изменениями данных объекта приложения.
- Управление кешированием данных объекта приложения.
- Управление версионностью данных объекта приложения.
- Управление видимостью объекта приложения для других пользователей.
- Копирование конфигурации объекта приложения.
- Управление созданием индекса полнотекстового поиска.
- Управление признаками объекта приложения:
    - Полнотекстовый поиск.
    - Представление БД.
    - Абстрактный.
    - Не наследуется.
    - Только чтение.
    - Скрыть в рубрикаторе.
    - Исторический.
    - Кешировать.
    - Агрессивное кеширование.
    - Включить аудит.

3.3.1.1.2. Описание модели данных объекта приложения

Для описания модели данных объекта приложения программный компонент «Конфигуратор» обеспечивает следующие возможности:

- Добавление новых полей следующих типов:
    - Целочисленный;
    - Длинный целочисленный;
    - Десятичный;
    - Длинный десятичный;
    - GUID;
    - Строковый;
    - Дата;
    - Дата со временем;
        - Дата и время в интерфейсе отображаются в локальном часовом поясе пользователя. В базе данных даты хранятся в серверном времени.
    - Булевский;
    - Ссылка;
    - Ссылка на системный объект;

  - Вложенный объект;
  - Вложенный список;
  - Вложенный список объектов;
  - Файл.
  - Характеристики.
- Группировка существующих полей.
- Редактирование существующих полей.
- Удаление существующих полей.
- Переименование полей, с учетом изменений связей с другими сущностями, с формированием отчета об измененных сущностях. Ограничение переименования – при импорте конфигурации.
- Задание настраиваемых типов данных на основе базовых (например, СНИЛС, ИНН) и использование их при создании полей в модели данных.
- Проверка для названий полей-идентификаторов (системных наименований, в объектах приложения, формах редактирования, атрибутах и т.д.) на несовпадение с зарезервированными словами.
- Указание для полей модели данных в зависимости от их типа следующих параметров и признаков:
  - Обязательность (кроме полей с типом вложенный объект, вложенный список, вложенный список объектов);
  - Уникальность (для всех типов полей);
  - Сообщение об ошибке – текст ошибки при валидации атрибутов, обязательность которых устанавливается с помощью признака "Обязательный" или с помощью правила;
  - Внешнего атрибута – для ввода наименования атрибута, которое будет отображаться в названии во внешней системе.
  - Фильтр (OQL) – для ввода условия фильтрации (для полей с типом Ссылка).
  - Условия фильтра – для ввода условия фильтрации с помощью конструктора условий (для полей с типом Ссылка).
  - Максимальную и минимальную длину (для строкового типа поля);
  - Максимальное количество знаков (для целочисленного типа поля);
  - Точную длину знаков (для строкового типа поля);
  - Точность числа и количество цифр после запятой (для десятичного и длинного десятичного типа);

- Минимальное и максимальное значение числа (для типов: целочисленный, длинный целочисленный, десятичный, длинный десятичный, дата, дата со временем);
- Значение по умолчанию (для всех типов полей, кроме вложенного объекта, вложенного списка, вложенного списка объектов и файла);
- Наименование атрибута во внешней системе;
- Маску ввода (для строковых полей и даты);
- Специальных символов для маски ввода (для строковых полей);
- Регулярное выражение (для строкового атрибута);
- Ссылку на объект приложения, как созданного через конфигуратор, так и разработанного на платформе (для полей с типом ссылка, ссылка на системный объект и вложенный список объектов);
- Подсказка (для всех типов полей);
- Вывод подсказки при просмотре длинных значений атрибутов при наведении курсора на длинное значение атрибута справочника;
- Вычисляемое значение (для всех типов полей). Значение задается с помощью скрипта;
- Скрытое значение поля (по умолчанию атрибут не отображается на панели фильтрации и на сгенерированных формах);
- Генерируемое значение поля (пересчитывается при изменении записи по заданному выражению);
- Основное значение поля (хранится в отдельной колонке таблицы БД, после удаления поля с включенным признаком "Основной" столбец таблицы БД, соответствующий удаленному атрибуту, не удаляется из базы данных, если в нем есть данные или другой объект приложения использует этот столбец);
- Режим загрузки данных таблицы (для полей с типом вложенный список объектов);
- Формат с разделителями (использование разделителя тысяч при отображении значения поля).
- Колонка связи в записи (для полей с типом ссылка, вложенный список объектов)
- Колонка связи в ссылке (для полей с типом ссылка, вложенный список объектов)
- Скрыть опции фильтра (для всех полей).

- Добавление меток для полей модели данных:
  - Добавление вложенного списка внутри другого вложенного списка;
  - Добавление списка значений.

- Указание полей, по которым будет строиться полнотекстовый индекс для объекта, что позволит использовать полнотекстовый поиск в ссылочных полях.
- Возможность указывать поля модели данных, значения которых не будут копироваться при копировании записи объекта приложения.
- Возможность указывать поля модели данных с типом Файл, вложения которых не будут подписываться.
- Возможность указать лицензионные флаги и модули для полей модели данных:
  - Значения флагов или модулей для атрибутов стоят выше в приоритете, чем флаги или модули на форме списка и остальных разделов, связанных с атрибутами.
- Вывод подсказки при просмотре длинных значений при наведении курсора на длинное значение атрибута справочника отображается полностью.
- Массовое перемещение атрибутов в списке.
- Массовое редактирование свойств атрибутов в списке.

3.3.1.1.3. Описание уникальных атрибутов объекта приложения

Программный компонент обеспечивает интеграцию объектов приложения Конфигуратора с объектами приложения продуктов БФТ.ЕНСИ (МДМ).

На вкладке Уникальные атрибуты конфигурации объекта приложения создаются комбинации из атрибутов объекта приложения, являющихся уникальными ключами для записей объектов приложения.

Уникальный атрибут представляет собой комбинацию атрибутов ОП, называемую также натуральным ключом записи объекта приложения. Это означает, что записи отличаются друг от друга значениями атрибутов объекта приложения, входящими в состав уникального атрибута. Можно создавать как один, так и несколько уникальных атрибутов для одного объекта приложения.

В модуле можно задать:

- Системное имя;
- Атрибуты – содержит перечень атрибутов, которые входят в состав уникального атрибута. Доступна возможность использовать атрибут в качестве первичного ключа.

3.3.1.1.4. Описание формы редактирования объекта приложения

Программный компонент обеспечивает возможность описания формы редактирования объекта приложения, а именно возможность задать:

- Параметры формы редактирования:
  - Расположение заголовков.

  - Ширина заголовка.
  - Ширина поля.
  - Ширина формы редактирования.
  - Высота формы редактирования.
  - URL для справки.
- Заголовок поля;
- Поле объекта;
- Порядок следования полей на форме;
- Наименование полей для отображения на форме на отличное от заданного в модели данных;
- Ширину и отступ заголовка;
- Ширину и отступ поля;
- Тип для строкового атрибута (текст, многострочный текст, редактор, расширенный редактор, пароль);
- Замещающий текст (отображается в поле ввода в отсутствии значения и указывает на ожидаемые данные).
- Условия фильтра (позволяет добавлять правила фильтрации в ссылочных компонентах по полям формы родительного объекта приложения).
- Форму редактирования ссылочных полей.
- Создать несколько форм редактирования с указанием:
  - Системного имени,
  - Отображаемого имени,
  - Описания,
  - Статусов,
  - Видов отображения,
  - Условий отображения.

Также существует возможность настройки CSS стилей для компонентов формы редактирования:

- CSS поля только для чтения – для настройки CSS-стилей для полей с активированным признаком **Только чтение**;
- CSS заголовка – для настройки CSS-стилей для заголовка поля вода.

Также существует возможность описать форму редактирования и форму списка для поля типа «Вложенный список», а именно:

- Указать поле объекта, заголовок и порядок их следования на форме редактирования вложенной таблицы;
- Изменить наименование полей для отображения на форме редактирования и списка на отличное от заданного в модели данных;
- Выполнения массового редактирования колонок формы списка для компонентов вида Список, Список ссылок, Связанный список и Список М-М.

Существует возможность описания формы редактирования объекта приложения с помощью визуального редактора. Этот редактор позволяет управлять компонентами формы, а именно:

- Описывать структуру формы (расположение компонентов на форме).
- Настроить отображение справки (поле «URL Справки»).
- Изменять свойства компонентов.
- Задавать связь компонента с полем модели данных.
- Описывать форму списка вложенного списка и вложенного списка объектов.
- Описывать форму редактирования вложенного списка (описывать структуру, изменять свойства компонентов и задавать связь компонентов с полями модели данных).
- Для вложенного списка есть возможности задать:
  - Признаки: свернут по умолчанию, доступен экспорт, только чтение, заблокирован, не показывать по умолчанию, не переносить строки;
  - Начальную высоту списка, в которой указывается оптимальное по умолчанию количество строк (для вложенных списков вида «Список», «Связанный список», «Список ссылок», «Список М-М», «Карта свойств»);
  - Высоту списка от/до;
  - Изменять размер списочных блоков путем их сжатия и растягивания на форме редактирования;
  - Поле для быстрой загрузки файлов (для вложенных списков вида «Список» с наличием дочернего атрибута типа «Файл»);
  - Отключение множественного выбора записей в компонентах «Список», «Связанный список», «Список ссылок», «Список М-М».

В визуальном редакторе доступны следующие визуальные компоненты:

- Контейнеры (Вкладки, Вкладка, Группа полей, Фрагмент, Строка, Колонка).
  - Для компонента Вкладки доступна возможность отображать все вкладки в выпадающем списке по кнопке меню вкладок.

- Элементы (Текст, Число, Чекбокс, Дата, Дата и время, Время, Ссылка, Ссылка или текст, Список ссылок, Гиперссылка, Фиксированная ссылка, Вложение, Код, Карта, Баннер, Кнопка, Разделитель, Набор ссылок, Перечисление, Переключатель, Фиксированная ссылка, Предпросмотр файла и другие).
- Компоненты Карточка, Разделы формы, Раздел формы, Отступ, Разделитель, Список и детали, Список карточек предназначенные для создание отдельных карточек с индивидуальным набором атрибутов, разделения формы редактирования на несколько панелей, создания боковой навигационной панели между разделами формы, а также для отображения связанных объектов в виде карточек.
- С помощью компонента Индикатор прогресса возможна настройка визуального отображения процесса выполнения какой-либо операции в виде линейной, круговой, приборной диаграммы.
- Для компонентов Ссылка, Ссылка или текст, Фиксированная ссылка, Список ссылок, Список М-М есть возможность настроить поведение выбора записей в справочниках для компонентов формы редактирования, через свойства компонента.

3.3.1.1.5. Описание формы списка объекта приложения

Программный компонент обеспечивает возможность описания стандартной формы списка объекта приложения, а именно возможности:

- Добавления, настройки и удаления колонок формы списка:
  - Вычисляемые колонки с возможностью задания формулы для условия с целью автоматического вычисления значений в колонке.
  - Тип данных колонки может быть выбран из стандартных типов данных:
    - Булевский.
    - Строковый.
    - Целочисленный.
    - Длинный целочисленный.
    - Десятичный.
    - Длинный десятичный.
    - Дата.
    - Дата со временем.
    - GUID.
    - Ссылка.
    - Ссылка на системный объект.
- Задать отображаемые колонки на форме списка.

- Выбор поля (с типом «Строковый» и признаком «Обязательный») для отображения из объекта приложения, на который ссылается колонка с типом Ссылка.
- Совершать массовое добавление колонок на форму списка.
- Совершать массовое редактирование колонок формы списка.
- Задать порядок следования колонок на форме списка.
- Задать наименование полей для отображения на форме на отличное от заданного в модели данных.
- Задать отображаемое имя заголовка группировки столбцов более чем по одному уровню вложенности.
- Задать отображаемые поля в подсказке на форме списка записей.
- Задать расположение фильтра (в панели фильтрации или заголовке таблицы).
- Задать поля, используемые для фильтрации.
- Задать режим фильтрации для строкового поля (по полному соответствию, по началу, по вхождению).
- Задать вид отображения (табличный, подробный или гибридный).
- Задать вид списка (древовидный для иерархического списка и табличный).
- Включить признак каскадного удаления.
- Включить признак редактирования данных в табличном виде отображения.
- Включить признак переноса данных на новую строку.
- Включить признак Составной фильтр.
    - Выбор объекта приложения, атрибут которого используется в качестве маски составного фильтра.
    - Выбор атрибута связанного объекта приложения, значения которого используются для фильтрации.
- Настроить выравнивание колонкам (по левому краю, по центру, по правому краю, по верхнему краю, по нижнему краю).
- Настроить начальную ширину колонки (ширина по умолчанию).
- Настроить минимальное и максимальное значение ширины колонки.
- Задать порядок по умолчанию (по возрастанию или по убыванию).
- Множественная сортировка (при удержании клавиши **shift** или **ctrl**).
- Указать значения, которые можно рассчитать по столбцу и вывести результат на форму записи с текстовым отображением:

  - Сумма;
  - Минимальное значение;
  - Максимальное значение;
  - Количество строк в колонке;
  - Среднее арифметическое по всем значениям колонки;
  - Количество уникальных значений;
  - Минимально значимое;
  - Сумма положительных значений;
  - Сумма отрицательных значений.
- Использование лицензионных флагов и модулей.
- Установки сортировки по умолчанию по нескольким колонкам с указанием порядка их применения.
- Возможность сохранять и отображать последние 5 введенных значений для колонок с типом Строковый в выпадающем списке в поле для ввода условия фильтрации.
- Возможность динамического управления редактированием ячеек на форме списка.
- Возможность выполнять сортировку по числовому значению.
- Возможность изменения порядка отображения колонок формы списка через действия.
- Включить признак Скрыть опции фильтра.
- Настройка автоматического скрытия кнопок на панели инструментов в каждом режиме отображения формы списка.

3.3.1.1.6. Настройка рабочего процесса (статусной модели) для объекта приложения

Программный компонент включает в себя встроенный механизм создания рабочих процессов (статусных моделей) для объектов приложения, а именно возможность ведения статусных моделей объектов приложения и задание:

- Статусов жизненного цикла объекта приложения;
- Действий и переходов для каждого статуса;
- Контролей для каждого действия;
- Обработчиков, описывающие логику выполнения действий и контролей.

Для регистрации действий пользователей по переводу записей справочников со статуса на статус предусмотрен Журнал перевода по статусам. В Журнале перевода по статусам фиксируется информация:

- Время и дата начала и окончания перехода по статусу;

- Логин и ФИО пользователя, инициировавшего переход по статусу;
- Наименование начального и конечного статусов;
- Наименование перехода между статусами.

3.3.1.1.7. Описание условий форматирования формы списка объектной модели данных

Программный компонент включает в себя настройку внешнего вида наименований колонок на форме списка, в том числе, обеспечивает следующие возможности:

- Добавления, изменения и удаления условий форматирования колонок формы списка.
- Изменения цвета, шрифта, подчеркивания и других параметров ко всей строке (шапке формы списка) или к отдельной колонке.
- Применения условия форматирования к конкретной колонке.
- Выбора полей для выборки, которые будут использованы для задания условий форматирования с помощью скрипта или редактора условий.
- Задания условия форматирования с помощью скрипта или в редакторе условий.
- Перемещение по списку записей и открытие записей через клавиатуру.

3.3.1.1.8. Описание фильтров формы списка объектной модели данных

Программный компонент включает в себя настройку фильтров формы списка объектной модели данных, в том числе, обеспечивает следующие возможности:

- Добавления, изменения и удаления фильтров для формы списка;
- Задания условий для фильтрации с помощью скрипта, в редакторе OQL, редакторе условий с применением операторов:
  - =/!=;
  - >/<;
  - >=/<=;
  - IN/NOT IN;
  - содержит/не содержит;
  - начинается на/заканчивается на;
  - соответствует/не соответствует;
  - заполнено/не заполнено;
  - кроме.
- Выбора поля объекта приложения, к которому необходимо применить фильтр.

- Выбор поля объекта приложения, содержащего маску ввода, к которому необходимо применить фильтр.
- Возможность автоматически применять фильтр при открытии справочника.
- Возможность запретить сбрасывать фильтр на форме списка записей.
- Настройка условий отображения по лицензионным флагам и модулям.
- Возможность использования параметризированных фильтров.
- Возможность запрета сброса параметризированных фильтров.
- Возможность закрепить фильтр по умолчанию на форме списка справочника.
- Возможность использования фильтра для ЭП.
- Возможность скрыть опции фильтра.
- Быстрая очистка созданного цветового форматирования.

3.3.1.1.9. Описание печатной формы

Программный компонент включает в себя настройку вызова печатной формы из формы редактирования записи объекта приложения по кнопке **Печать.** Настройка позволяет:

- Указывать системное имя печатной формы.
- Указывать отображаемое имя печатной формы.
- Указывать шаблоны печатных форм (п. 3.5.1).
- Указывать объекты приложения, которые будут использоваться для представления формы ввода параметров печатной формы.
- Указывать форматы экспорта печатной формы:
  - pdf;
  - docx;
  - odt;
  - xlsx;
  - xls;
  - ods;
  - html;
  - rtf;
  - xml;
  - csv.
- Указывать варианты заполнения набора параметров отчета, которые должны быть заполнены перед его формированием.

- Задавать условия доступа к формированию печатной формы для:
  - разрешенных ролей пользователей;
  - лицензионных флагов;
  - лицензионных модулей.
- Скрывать возможность формирования печатной формы.
- Устанавливать признак массовой печати документов из формы списка.
- Управлять положением кнопки «Печать» на форме редактирования записи объекта приложения.

3.3.1.1.10. Описание правил валидации объектной модели данных

Программный компонент включает в себя настройку условий, при выполнении которых происходит или не происходит то или иное действие с записью объекта приложения или на форме редактирования записи. Настройка позволяет:

- Указывать системное и отображаемое имя правила валидации.
- Указывать поле объекта приложения, которое обрамляется красной рамкой при срабатывании правила валидации.
- Определять условие валидации в виде скрипта или с применением редактора правил, включающего операторы:
  - =/!=;
  - >/<;
  - >=/<=;
  - IN/NOT IN;
  - содержит/не содержит;
  - начинается на/заканчивается на;
  - соответствует/не соответствует;
  - заполнено/не заполнено;
- Определять уровень серьезности ошибки и сообщение об ошибке.
- Указывать статус и действие для срабатывания правила валидации.
- Формировать сообщение об ошибке скриптом.
- Выбирать поля для выборки, которые будут использованы для написания условия валидации с помощью скрипта или редактора условий.
- Копировать правило валидации в пределах одной объектной модели приложения.
- Переходить по гиперссылке в окне ошибки валидации на форме записи к проверяемому полю.

- Выполнять повторную валидацию при обработке из основного ОП.
- Выполнять правила валидации из вложенного ОП.
- Формировать правило валидации сценарием с указанием переменной сценария, в которой содержится результат выполнения валидации.
- Формировать правило валидации и сообщение об ошибке в одном поле «Скрипт».
- Выбор среды выполнения скрипта валидации:
    - Kotlin.
    - Kotlin (JS).
    - Kotlin (JVM).

3.3.1.1.11. Описание действий объектной модели данных

Программный компонент включает в себя настройку индивидуального меню с действиями на форме редактирования или форме списка объектной модели приложения с возможностями:

- Создания, изменения, удаления группы действий.
- Свойств действия:
    - Выбора группы действия, скрытия/отображения группы действия.
    - Заголовка, иконки, подсказки для действия.
    - Роли для возможности выполнять действие только пользователям с выбранной ролью.
    - Размещение и расположение кнопки действия.
    - Задать системные имена действий перед и после которого требуется разместить новое действие.
    - Настроить тип действия.
    - Задавать значения параметрам для действий, выполняемым скриптом приложения.
    - Задавать системные имена группы действия для группировки действий и отображения в контекстном меню формы списка вида AG Grid.
    - Выбирать объект конфигурации, указанного в сценарии, для действий, выполняемым сценарием.
    - Выбирать форму редактирования объекта конфигурации, указанного в сценарии, для действий, выполняемым сценарием.
    - Указывать варианты заполнения набора параметров сценария, которые заполняются при выполнении сценария, для действия, выполняемого сценарием.
    - Настройка условий отображения по лицензионным флагам и модулям.

3.3.1.1.12. Описание триггеров объектной модели данных

Программный компонент включает в себя создание, изменение, удаление и настройку триггеров объектной модели приложения с возможностями определения:

- События для срабатывания триггеров;
- Скрипта или сценария для срабатывания триггера.

3.3.1.1.13. Описание правил поведения элементов управления формы редактирования объектной модели данных

Программный компонент включает в себя создание, изменение, удаление и настройку правил поведения элементов интерфейса на форме редактирования записи с данными объектной модели данных, в том числе:

- Изменение порядка расположения правила в списке.
- Выбор или выделение одного правила в списке.
- Копирование выбранного правила.
- Включение одного или нескольких правил в списке.
- Включение всех правил в списке.
- Выключение всех правил в списке.
- Удаление всех выключенных правил в списке.
- Создание правил с использованием операторов:
  - =/!=;
  - >/<;
  - >=/<=;
  - IN/NOT IN (доступен множественный выбор значений через операции Справочник (список), Системный справочник (список), Список значений), Параметр ОП
  - содержит/не содержит;
  - начинается на/заканчивается на;
  - соответствует/не соответствует;
  - заполнено/не заполнено;
  - состоит в/не состоит в;
  - свёрнута/развёрнута по умолчанию.
- Удаление операторов в конструкторе правил:
  - НЕ
  - И
  - ИЛИ

- Применение разных функций в правилах, например:
  - Получение Логина текущего пользователя;
  - Получение полного имени текущего пользователя;
  - Суммирование строк списка по указанному полю;
  - Определение количества строк в списке;
  - и другие функции.
- Управление поведением компонентов формы редактирования с помощью действий:
  - Показать/Скрыть.
  - Сделать обязательным/Сделать не обязательным.
  - Сделать редактируемым/Сделать не редактируемым.
  - Задать маску ввода.
  - Присвоить значение.
  - Установить заголовок поля.
  - Свернуть/Развернуть (для табличных компонентов формы редактирования и компонента «Группа полей»).
  - Сфокусироваться.
- Поиск атрибута по системному наименованию или отображаемому имени при формировании условия.
- Отладка правил.

3.3.1.1.14. Настройка подписываемых данных объектной модели данных

Программный компонент позволяет настраивать системное наименование и отображаемый заголовок дайджеста модели данных, а также:

- Добавлять новую версию дайджеста;
- Порядковый номер версии дайджеста;
- Тип заполнения подписываемых полей;
- Формат, в котором генерируются подписываемые данные, в соответствии с настройкой полей, указанной в данной версии дайджеста;
- Признак версии дайджеста, по умолчанию используемой при подписании;
- Алгоритм канонизации json;
- Дата начала действия версии дайджеста;
- Массовое добавление подписываемых полей;
- Настройка регламента ЭП.

3.3.1.1.15. Просмотр json-представления объектной модели данных

Программный компонент позволяет просматривать json-представления объектной модели данных. Описание в формате *JSON* обновляется автоматически по мере изменения конфигурации объекта приложения.

3.3.1.1.16. Настройка скрипта формы списка

Программный компонент позволяет управлять элементами на форме списка объекта приложения при помощи скриптом на языке Kotlin или JavaScript:

- Настраивать гиперссылки на форме списка.
- Настраивать иконки для шапки и ячейки колонок на форме списка.

3.3.1.1.17. Настройка отображаемого имени

Программный компонент обеспечивает возможность описания формы списка объекта приложения в подробном режиме просмотра, а именно возможности:

- Выбор полей значений, которых должны отображаться в списке данных объекта приложения.
- Настройка скрипта для отображения данных в списке объекта приложения.

3.3.1.2. Редактирование объекта приложения[1]

Программный компонент обеспечивает возможность редактирования созданного через Конфигуратор объекта приложения, а именно:

- Изменение модели данных объекта приложения:
  - Добавление новых полей.
  - Редактирование существующих полей.
  - Удаление существующих полей.
  - Добавление уникальности по группе полей.
  - Редактирование существующей уникальности по группе полей.
  - Удаление существующей уникальности по группе полей.
- Изменение описания формы редактирования:
  - В простом режиме:
    - Изменение порядка следования полей.
    - Добавление отображения поля на форме.
    - Удаление отображения поля на форме.

[1] Функция удаления невозможна, если удаляемые данные используются в экземплярах объекта приложения.

        - Изменения наименования поля для отображения на форме.
    - В режиме управления компонентами формы:
        - Изменение структуры формы (расположения компонентов на форме).
        - Добавление новых компонентов на форму.
        - Удаление компонентов с формы.
        - Изменение свойств компонентов.

- Изменение описания формы списка:
    - Изменение порядка следования полей.
    - Добавление отображения поля на форме.
    - Удаление отображения поля на форме.
    - Изменение наименования поля для отображения на форме.
    - Изменение расположения фильтра.
    - Изменение набора полей, отображаемых в качестве фильтров.
    - Изменение режима фильтрации для строкового поля.
- Изменение описания рабочего процесса:
    - Назначение объекту приложения нового рабочего процесса.
    - Изменение существующего рабочего процесса:
        - Добавление, удаление статусов и изменение их наименований.
        - Добавление, удаление действий и изменение их логики и наименований.
        - Добавление, удаление контролей и изменение их логики и наименований.

При сохранении сделанных изменений в системе автоматически изменяется вся инфраструктура, необходимая для работы с этим объектом приложения. То есть автоматически изменяется состав и порядок следования реквизитов на форме экземпляра и форме списка, перечень действий и наименования статусов, логика выполнения действия и контролей – в соответствии с новой структурой объекта приложения.

Данные изменения отражаются в системе без перезапуска сервера приложения.

3.3.1.3. Удаление объекта приложения

Программный компонент обеспечивает возможность удаления созданных через конфигуратор объектов приложения.

При удалении пользовательского объекта приложения в системе автоматически физически удаляется инфраструктура, которая использовалась для работы с объектом приложения: форма экземпляра, форма списка.

3.3.1.4. Ведение моделей данных объекта приложения

Программный компонент обеспечивает следующие действия с моделями данных объектов приложения:

- Загрузка модели данных объекта приложения из файла.
- Сохранение модели данных в файл.
- Добавление в избранные моделей данных объектов приложения.
- Добавление в закладки моделей данных объектов приложения
- Обновление перечня моделей данных объектов приложения.
- Создание копии модели данных объекта приложения
- Построение ER-диаграммы модели данных объекта приложения.
- Просмотр истории изменений элементов конфигурации.
- Просмотр истории взаимосвязей элементов конфигурации.
- Просмотр таблицы базы данных модели данных объекта приложения.
- Создание статических представлений модели данных объекта приложения/
- Просмотр истории изменений модели данных объекта приложения.
- Поиск моделей объектов приложения и объектов приложения с данными.
- Обновление данных в таблице БД материализованных представлений.
- Генерация SQL-запросов выполняющих отбор записей представления.

3.3.1.4.1. Загрузка модели данных объекта приложения из файла

Программный компонент обеспечивает загрузку из файла zip-архива в Конфигуратор всех объектных моделей приложений со статусными моделями.

3.3.1.4.2. Сохранение модели данных в файл

Программный компонент обеспечивает выгрузку в файл zip-архива из Конфигуратора всех объектных моделей приложений со статусными моделями.

3.3.1.4.3. Добавление в избранное моделей данных объектов приложения

Программный компонент обеспечивает добавление выбранной группы или одной объектной модели приложений в избранное, с возможностью переключения просмотра только избранных моделей или всех моделей объектов приложения.

3.3.1.4.4. Добавление в закладки моделей данных объектов приложения

Функционал обеспечивает добавление выбранной объектной модели приложения в закладки, с возможностью переключения просмотра только недавно просмотренных или моделей объектов приложения, добавленных в закладки. Доступно добавление:

- Добавление объектов приложения в закладки.
- Добавление записей объектов приложения в закладки.
- Возможность перетаскивания объектов приложения и записей объектов приложения в любое место навигационной панели, а также перетаскивания между группами. Группу можно перетащить в любое место списка, а также поместить в другую группу при помощи перетаскивания с помощью мыши.
- Просмотр недавно открытых объектов приложения или записей в группе **Недавно открытые**.

3.3.1.4.5. Обновление перечня моделей данных объектов приложения

Программный компонент обеспечивает возможность автоматического обновления перечня объектных моделей приложений.

3.3.1.4.6. Создание копии модели данных объекта приложения

Программный компонент обеспечивает возможность создания копии модели данных объекта приложения.

3.3.1.4.7. Построение ER-диаграммы модели данных объекта приложения

Программный компонент обеспечивает возможность просмотра ER-диаграммы – связей между объектными моделями приложений (в том числе системными) внутри группы или модуля.

3.3.1.4.8. Просмотр истории изменений элементов конфигурации

Программный компонент обеспечивает возможность просмотра изменений по свойствам или компонентам, описывающим структуру конфигурации, а также просмотр сравнений json-представлений форм редактирования.

3.3.1.4.9. Просмотр взаимосвязей элементов конфигурации

Программный компонент обеспечивает возможность просмотра и открытия объектов конфигураций, связанных с выбранном объектом приложения.

3.3.1.4.10. Просмотр таблицы базы данных модели данных объекта приложения

Программный компонент обеспечивает просмотр таблицы базы данных модели данных приложения с возможностью:

- Отображения названия, типа таблицы, перечня колонок таблицы и свойств каждой выбранной колонки;
- Ведения индексов колонок таблицы;
- Ведения политики RLS (разграничение доступа по строкам таблиц);
- Отображения объектов приложения, в которых используется таблица.

3.3.1.4.11. Создание статических представлений модели данных объекта приложения

Программный компонент обеспечивает создание статистических представлений модели данных объекта приложения. Особенности использования представления в Конфигураторе:

- Представление используется как «виртуальная» таблица базы данных;
- Представление не хранит данных, а извлекает данные из таблиц в момент обращения к представлению;
- Представления могут быть использованы при разработке отчетов;
- Редактирование и добавление записей невозможно, доступно только чтение.

3.3.1.4.12. Просмотр истории изменений модели данных объекта приложения

Программный компонент обеспечивает возможность просмотра истории изменений модели данных приложений.

3.3.1.4.13. Поиск моделей объектов приложения и объектов приложения с данными

Программный компонент обеспечивает поиск моделей объектов приложения и объектов приложения с данными:

- По русским и латинским названиям сущностей (названий атрибутов, таблиц, фильтров, правил и др.);
- По json-описанию объекта приложения;
- По названию объектов приложения или конфигураций;
- По метаданным моделей объектов приложений.

3.3.1.5. Ведение данных объекта приложения

Управление данными объекта приложения осуществляется через пользовательский интерфейс.

Возможны следующие функции по ведению записей с данными:

- Добавление, редактирование, копирование, удаление выделенных записей.
- Просмотр истории изменения записи.
- Ведение шаблона для записи.
- Наложение электронной подписи для одной или нескольких записей.
- Изменение статуса записи.
- Изменение нескольких выбранных записей.
- Загрузка данных из Excel.

Для формы списка справочника реализованы возможности:

- Отображения времени загрузки списочной формы на форме списка объекта приложения;
- Переключения между страницами формы списка;
- Отображение количества выбранных и общего количества записей на странице формы списка.

Для формы редактирования записи справочника реализована возможность:

- Быстрой прокрутки к началу формы редактирования по комбинации клавиш ctrl + home.

3.3.1.5.1. Перечень действий с записями

Имеются возможности выполнять следующие действия с записями:

- Добавление, редактирование, копирование, удаление выделенных записей, выделение всех записей на странице, выделение всех записей в объекте приложения независимо от количества страниц;
- Сравнение различий двух выделенных записей, включая вложенные списки;
- Для вложенной таблицы (таблицы на форме редактирования):
  - копирование с заменой (при выполнении действия открывается окно создания новой записи с уже заполненными данными, выбранной ранее записи);
- Настройка возможности переноса слов в колонке табличной формы списка;
- Фильтрация, сортировка;
- Фильтрация по свойствам объекта приложения через раздел «Дескриптор справочника» – «Модель данных»;

- Фильтрация по колонкам с поиском записей, в которых определенные поля заполнены или пустые, с помощью признаков **Заполнено** и **Кроме**;
    - Отображение записей с пустым значение в колонке при фильтрации с условием **Не содержит** и включенными **Кроме**.
- Возможность скрыть кнопку фильтрации по условиям на форме списка справочника;
- Копирование отфильтрованных записей;
- Раскрытие вложенных записей для иерархических структур данных;
- Подсчет итогов по графам списка;
- Открытие записи объекта приложения на отдельной вкладке браузера (удерживая клавишу ctrl нажать на интересующую запись);
- Отмена выделения в ветке для иерархических структур;
- Адаптация размера таблицы в модальных окнах в соответствии с изменением размера модального окна;
- Копирование ссылки записи;
- Открыть запись в отдельной вкладке браузера;
- Массовое выделение дочерних записей у нескольких родительских записей в иерархических формах (также для формы списка вида AG Grid).
- Добавление записи в закладки в пользовательском рубрикаторе.
- Поиск полей на форме редактирования.
- Изменение высоты таблицы на форме редактирования.
- Создание json-представления записи объекта приложения.
- Изменение режима отображения записей с сохранением выбранного состояния в локальное хранилище браузера.
- Копирование ссылки на отфильтрованные записи справочника.

3.3.1.5.2. Просмотр истории изменения записи

Программный компонент обеспечивает возможность просмотра истории изменений записей объектов данных приложений.

3.3.1.5.3. Ведение шаблона для записи

Программный компонент обеспечивает сохранение записи как шаблона с возможностями:

- Создания, изменения, удаления шаблона записи;
- Указания шаблона, как общедоступного или собственного;

- Создания записи из шаблона.

3.3.1.5.4. Наложение электронной подписи для одной или нескольких записей

Программный компонент обеспечивает возможности:

- Подписывать одну или несколько записей;
- Устанавливать роль для подписания записи;
- Выбирать и скачивать сертификат подписи;
- Просматривать сведения о сертификате;
- Устанавливать формат подписи.

3.3.1.5.5. Изменение статуса записи

Программный компонент обеспечивает возможность изменения статуса записи при наличии статусной модели, привязанной к объектной модели приложения.

3.3.1.5.6. Изменение нескольких выбранных записей

Программный компонент обеспечивает возможность одновременного изменения в каком-либо поле нескольких записей объекта приложения.

3.3.1.5.7. Контекстное меню

Открытие контекстного меню нажатием правой кнопки мыши по записи в объекте приложения в режиме просмотра **Подробный вид**:

- Удаление выделенных записей;
- Копирование;
- Сохранение запиши как шаблон;
- Изменение статуса записи;
- Создание json-представления записи.
- Подсчет итогов.

Открытие контекстного меню нажатием правой кнопки мыши по записи в объекте приложения в режиме просмотра **Табличный вид**:

- Экспорт выделенных строк;
- Удаление выделенных записей;
- Копирование записи;
- Экспорт (настраиваемый) выделенных строк;
- Сохранение записи как шаблон;

- Изменение статуса записи;
- Создание json-представления записи.
- Подсчет итогов.

3.3.1.5.8. Выгрузка записей объектов приложения

- Выгрузка записей из справочников в формате xlsx или ods. в синхронном режиме или в асинхронном режиме.
- Экспорт в XLSX (учитывается разница во времени между временем пользователя и временем сервера).
- Проверка допустимого количества записей для экспорта через параметр ice.report.load.maxRowCount.
- Экспорт записей иерархического справочника с сохранением древовидной структуры в XLSX.
- Экспорт в ODS.
- Экспорт выделенных строк.
- Настройки экспорта.
- Экспорт (настраиваемый).
- Экспорт (настраиваемый) выделенных строк.

3.3.2. Управление сценариями

Управление сценариями представляет собой инструмент для автоматизации действий системы с помощью применения функциональных блоков, выполняющих определенные действия.

Сценарии представляют собой графический интерфейс, позволяющий автоматизировать необходимый процесс в проектируемой системе без привлечения разработчика.

Например, сценарии можно использовать для:

- автоматического добавления/редактирования/удаления записей объекта приложения;
- автоматического создания отчета;
- отправки сообщения и множества других функций, которые надо автоматизировать;
- автоматического тестирования работоспособности других сценариев.

Один сценарий может включать в себя как одно действие, так и их последовательность, а также другие сценарии или скрипты, что позволяет автоматизировать процессы разных уровней сложности.

Управление сценариями включает выполнение следующих функций:

- Поиск сценариев.
- Создание, изменение, удаление одного или нескольких сценариев, группы сценариев.
- Обновление списка сценариев.
- Копирование записи сценария.
- Проверка корректности настройки элементов сценария с помощью проверок самодиагностики.
- Добавление выбранных сценариев в избранное.
- Применение сценариев в действиях объектов приложения.
- Привязка сценария к триггерам объекта приложения.
- Настройка сценариев, включающая функции:
  - привязка сценария к статусной модели объекта приложения, чтобы автоматически выполнять требуемые действия при изменении статуса записи с данными объекта приложения;
  - привязки сценария к триггерам, действиям объекта приложения;
  - применения сценариев в действиях объектов приложения;
  - вызов сценария из скрипта формы редактирования объекта приложения;
  - добавление переменных в сценарии;
  - добавление в переменные сценария типов значений;
  - копирование одной или сразу нескольких переменных в сценарии;
  - добавление условий отбора, позволяющих осуществить выборку записей объекта приложения;
  - добавление и настройка элементов диаграммы сценария:
    - Старт.
    - Выход.
    - Ошибка.
    - Комментарий.
    - Условия.
      - Скрипт.
      - Условие.
    - Скрипт приложения.
    - Скрипт.
    - Метод сервиса.
    - Сценарий.

        - Возможность открытия дочернего сценария по гиперссылки в отдельной вкладке браузера.
- Создание записи.
- Редактирование записи.
- Удаление записи.
- Найти записи.
- Редактирование переменных.
- Сообщение.
    - Использование параметров для отправки сообщений (Каналы, Роли адресатов, Пользователи адресатов, Emeil'ы адресатов, телефоны адресатов, файл вложения, активно до)
- Отчет.
- Sql.
- http-запрос (возможность отправки файлов в http-запросе).
    - GET.
    - POST.
    - PUT.
    - PATCH.
    - HEAD.
    - DELETE.
    - Выбор фабрики ssl-контекста.
- json.
    - Сохранение json-запроса в переменные и поля с типом Файл.
- Xml.
    - Сохранение xml-схемы в переменные и поля с типом Файл.
- Валидация – для проверки выполнения заданных условий и порождение ошибки в случае невыполнения условий.
- Ошибка – для вывода пользователю сформированного сообщения ошибки на шаге сценария.
- Запуск процесса – для запуска бизнес-процесса.
- Задача процесса – для выполнения пользовательской задачи BPM-процесса.
- Сообщение процесса – для отправки сообщения для BPM-процесса.
- Поиск процесса – для поиска экземпляров BPM-процесса по заданным критериям:

- Процесс приложения.
- Идентификатор экземпляра процесса.
- Идентификатор экземпляра родительского процесса.
- Идентификатор экземпляра подпроцесса процесса.
- Запущен после.
- Запущен до.
- Завершен после.
- Завершен до.
- Состояние.

- Также Поиск процесса используется для сохранения результатов в:

  Поле – Поле, в которое сохраняется результат поиска. Значение поля выбирается из выпадающего списка атрибутов объекта приложения, который указан в свойствах сценария.

  - Переменную – Переменная, в которую сохраняется результат поиска. Значение поля выбирается из выпадающего списка переменных сценария.

- Сообщение интеграции.

- Выделение и перемещение сразу нескольких элементов диаграммы сценария;
- Отладка сценария;
- Поиск элементов, используемых в сценариях;
- Выделение элемента сценария на диаграмме;
- Просмотр предыдущих и следующих шагов для выбранного элемента, используемого в сценарии.

- Возможность запуска, повторного выполнения и хранения результатов выполнения сценариев автоматического тестирования в Журнале выполнения автотестов с целью периодической проверки работоспособности любых сценариев, которые могут вызывать другие сценарии, процессы приложения или отдельные задачи процесса, сервисы нотификации (отправка/получение уведомлений) и сервисы REST API;
- Отладка сценариев для выявления ошибок или проверки корректности работы;
- Работа с консолью диаграммы сценария при отладке;
- Вызов контекстного меню в консоль сценария:
  - очистка консоли;
  - копирование данных консоли;
  - скачивание данных консоли в виде log-файла;
  - поиск по данным в консоли.

- Возможность выделять несколько действий в сценарии.
- Возможность копировать несколько выделенных действий в сценарии.
- Возможность перемещать несколько выделенных действий в сценарии.
- Возможность устанавливать связь для выделенных действий сценария.

3.3.3. Управление XML схемой

Управление XML схемой представляет собой инструмент, обеспечивающий работу с XML схемами в форматах *.xsd и *.zip и позволяет выполнять следующие функции:

- Добавление файлов XML схем;
- Хранение файлов XML схем с возможностью использования в элементах диаграммы сценариев;
- Изменение файлов XML схем;
- Удаление файлов XML схем;
- Просматривать список сценариев, для которых используется схема;
- Переходить по ссылке на сценарий, в котором используется схема;
- Контролировать удаление схем, которые используются в сценариях.

Хранение файлов XML схем позволяет использовать их в элементе диаграммы сценария XML.

3.3.4. Конструктор REST API

Конструктор REST API является инструментом для быстрой разработки API на основе HTTP методов без привлечения разработчиков и позволяет выполнять следующие функции:

- Отображение перечня запросов между «БФТ.Платформой» и сторонними сервисами/приложениями;
- Добавление, изменение, удаление, настройка запросов (сервисов);
- Возможности указания:
  - Коллекции запроса (сервиса);
  - Объекта приложения, к которому привязан сервис;
  - Шаблона пути с параметрами и методом:
    - GET.
    - POST.

        - PUT.
        - PATCH.
        - DELETE.
        - HEAD.
        - OPTIONS.
    - Идентификатора операции;
    - Отображения имени;
    - Описания сервиса;
- Возможности:
    - Добавления и настройки параметров с указанием места их расположения;
    - Указания в ответе:
        - Системное имя;
        - Описание;
        - Имя поля ответа;
        - Объект приложения;
        - Тип результата (запись, список записей, группировка, значение);
        - Обязательных полей;
        - С заданием условия фильтрации, дополнительных полей;
        - С заданием условий его формирования.
    - Указания в запросе:
        - Методов работы с полями ОП, к которому привязан сервис;
        - Обязательных полей;
    - Определения правил валидации;
    - Использования сценариев;
    - Выбора ролей пользователя для авторизации.
    - Возврат данных, заполненных в сценарии.

3.3.5. Управление скриптами

Управление скриптами является инструментом для хранения и обновления наборов скриптов для применения в объектных моделях данных.

Управление скриптами:

- обеспечивает выполнение следующих функций:
    - Автоматического добавления/редактирования/удаления записей;
    - Нахождения учетных записей пользователей по параметру;
    - Определения статуса записи по сообщению шины и др.

- обеспечивает следующие возможности по настройке скриптов:
  - Поиск скриптов;
  - Создание, изменение, удаление одного или нескольких скриптов, группы скриптов;
  - Обновление списка скриптов;
  - Добавление выбранных скриптов в избранное;
  - Компиляция скриптов;
  - Использование системных скриптов приложения;
  - Просмотр истории изменений в скриптах.

3.3.6. Управление импортом и экспортом модели данных

Управление импортом и экспортом модели данных является инструментом для:

- выгрузки всех или выбранных объектных моделей в файл zip-архива;
- загрузки в Конфигуратор из файла zip-архива (предварительно выгруженная из него же) всех или выбранных объектных моделей и статусных моделей.
- Поддержка настройки security_invoker для представлений в версиях PostgreSQL от 15 и выше, позволяющая учитывать защиту на уровне строк (RLS) при загрузке конфигурации из файла или выгрузке конфигурации в файл.

Инструмент обеспечивает выполнение следующих функций:

- Загрузка конфигурации из файла:
  - Выбор файла с конфигурацией из Проводника.
  - Выбор другого файла конфигурации для импорта (кнопка Выбрать другой файл).
  - Выбор версии продукта, в который осуществляется последующий импорт конфигурации.
  - Формат файла zip.
  - Загрузка конфигурации из файла с отображением информации:
    - Имя конфигурации;
    - Имя продукта;
    - Версия продукта;
    - Подключение к БД;
    - Список загружаемых объектных моделей.
  - Настройка параметров выполнение импорта:

        - Выбор режима объединения записей при наличии записей в конфигурации (пропускать существующие, заменять из файла, прервать выполнение, выполнить только изменение).
        - Возможность пропуска ошибок сохранения отдельных объектных моделей без остановки импорта остальных ОП.
        - Выбор и выполнение проверок конфигурации.
    - Внесение изменений в структуру БД (публикация объектных моделей).
    - Включение в Журнал выполнения отладочной информации.
    - Отмена импорта.
    - Возможность скачивания информации по загрузке из файла формата log.
    - Возможность перекомпиляции измененных скриптов.
- Выгрузка конфигурации в файл:
    - Свойства выгружаемой конфигурации:
        - Наименование.
        - Описание.
        - Целевая версия продукта.
    - Выбор полных или частичных объектных моделей и их записей(системных и пользовательских) для выгрузки конфигурации:
        - Хранимые процедуры.
        - Представления БД.
        - Индексы БД.
        - Политики RLS.
        - Группы настроечных параметров.
        - Связанные сущности.
    - Фильтрация по сущностям платформы:
        - По автору;
        - По времени.
    - Поиск и выбор связанных сущностей для выбранных объектов из перечня:
        - Процессы приложения.
        - Объект приложения.
        - Сценарий.
        - XML схемы.
        - Скрипты.
        - Источники данных.
        - Статусные модели.

        - Шаблоны отчетов.
        - Роли.
        - Настроечные параметры.
        - Шаблоны сообщений.
        - Записи справочников из ссылочных полей.
    - Возможность отмены выгрузки.
    - Создание и сохранение шаблона для выгрузки объектных моделей:
        - Наименование шаблона.
        - Целевая версия продукта.
        - Автор.
        - Дата создания.
    - Выбор шаблона из списка шаблонов.
    - Просмотр в списке и применение шаблона.
    - Выгрузка и сохранение конфигурации в файле формата zip.
    - Журнал выполнения выгрузки объектных моделей:
        - Отображение времени выгрузки;
        - Пошаговое отображение выгрузки.
    - Возможность скачивания информации по выгрузке в файл формата log.
- Исключения конфигурации:
    - Защита полей объектов конфигурации от изменений.
    - Создание исключений конфигурации.
    - Автоматическое создание исключений конфигурации.
    - Комментирование изменений и простор истории их внесения.
    - Открытие записей по прямой ссылке.
- Объекты конфигурации:
    - Хранение записей конфигурации.
    - Выгрузка записей конфигурации в файл при экспорте.
- События обновлений конфигураций:
    - Хранение записей конфликтов, возникших при применении патчей и снепшотов, а также при импорте конфигурации.
    - Простор только невыполненных изменений.

3.3.7. Управления репозиторием

Управление репозиторием является инструментом для ведения каталога обновлений объектных моделей данных в виде файловой системы (снепшоты и патчи).

Инструмент обеспечивает выполнение следующих функций:

- создание патчей обновления конфигураций и данных справочников в формате json и подготовка патча для отправки в серверную папку для хранения;
- возможность автоматизированного создания патча сразу после обновления стенда.
- создание снепшотов;
- исключение из снепшотов представлений баз данных;
- настройка сущностей для патчей;
- уведомления пользователя об удалении представления базы данных, если в нём были произведены изменения до создания патча;
- очистка от дефектных элементов (удаление значений по умолчанию для атрибутов, переменных сценариев, параметров REST API и настроечных параметров, если в качестве значения по умолчанию указана ссылка и тип значения по умолчанию не равен «Константа») перед применением патча или снепшота;
- выгрузка полного состояния конфигурации на момент времени;
- возможность поиска патчей на форме подготовки;
- обновление списка репозиториев;
- возможность выбора ветки репозитория, при отправке патча;
- удаление неактивных патчей/снепшотов при отправке в GIT:
  - логическое удаление записи с UI,
  - удаление самого файла в базе данных,
  - удаление файла из GIT;
- фильтрация изменений по выбранному пользователю или времени для отдельных элементов конфигурации;
- сохранение выделенных элементов в дереве изменений при очистке поля фильтрации по пользователю;
- возможность просмотра и последующего выбора связанных объектов, необходимых для формирования патча (при формировании патча автоматически присваивается наименование в формате YYYY.MM.DD HH-MM-SS-MS);
- возможность просмотра и последующего выбора связанных объектов, рекомендованных для формирования патча (при формировании патча автоматически присваивается наименование в формате YYYY.MM.DD HH-MM-SS-MS);

- возможность применения патчей, входящих в сборку вместе с снепшотом;
- возможность применения дублирующихся патчей с изменением номера версии;
- сравнение текущего состояния конфигурации и последней загруженной конфигурации;
- удаление неактуальных патчей/снепшотов конфигурации;
- удаление из веток репозитория, отличных от выбранных веток;
- выгрузка списка репозиториев конфигураций в файл формата xls;
- выгрузка списка репозиториев конфигураций в файл формата ods;
- выполнение патчей в одной транзакции (доступно при включении настройки application.properties ice.conf.patches.use-single-transaction) Для включения настройки ice.conf.patches.use-single-transaction. необходимо выключить настройку ice.conf.patches.allow-second-pass, так как одновременная работа данных настроек не поддерживается;
- возможность просмотра информации об авторах изменений;
- объединение нескольких патчей в единый образ;
- возможность быстро находить взаимосвязанные патчи;
- генерация системного неизменяемого идентификатора в json-представлении объекта приложения;
- переключение между патчами/снепшотами;
- сброс конфигурации до выбранного патча/снепшота;
- сохранение непропатченной конфигурации при откате на патч/снепшот;
- выполнение снепшота объекта для выгрузки частичного состояния конфигурации на момент времени;
- просмотр прогресса установки патчей/снепшотов;
- частичное применение патча/снепшота.

3.3.8. Самодиагностика

Самодиагностика является инструментом для хранения и запуска проверок объектных моделей приложений и реализует:

- проверку целостности и непротиворечивости текущей объектной модели приложения;
- выдачу сообщений об обнаруженных проблемах;
- подготовку импортов и переменных для перекомпиляции скриптов форм редактирования;

- перекомпиляцию скриптов;
- сохранение выбранных проверок в профиль фильтров, а также применение профиля фильтров для выполнения выбранных проверок;
- сохранение порядка и ширины колонок в профиль фильтра;
- запрет на повторную перекомпиляцию скриптом для одной и той же конфигурации;
- формирование zip-архива с результатов проверки конфигурации в json-формате;
- отправка пуш-уведомлений о начале и окончании проверки;
- экспорт журнала перекомпиляции скриптов в формате xlsx;
- экспорт результатов перекомпиляции в файл формата xlsx;
- проверку:
  - наличия таблицы для объектной модели приложения;
  - наличия ссылки на несуществующие объектные модели приложения в атрибутах, компонентах формы редактирования;
  - не валидных скриптов;
  - скриптов, заданных в значении переменных сценариев;
  - наличия атрибутов в объектных моделях приложения;
  - соответствия имён таблица в дескрипторе объекта приложения в филдсете;
  - наличия в объектных моделях приложения атрибутов, используемых в списочных формах и формах редактирования;
  - наличия дубликатов в системном имени атрибутов;
  - наличия дубликатов компонентов формы редактирования объекта приложения;
  - наличия в атрибуте ссылки на абстрактный объект приложения;
  - наличия неиспользуемых переменных в сценариях;
  - наличия отключенных правил в объекте приложения;
  - наличия вычисляемых полей на форме списка;
  - наличие полей, указанных на форме списка компонента, отсутствующих в связанном ОП;
  - корректности условия для вложенных списков в кубике с типом "Условие" и др.

3.3.9. Ведение регламентов электронной подписи

Ведение регламентов электронной подписи является инструментом для настройки и подписания экземпляров объектных моделей данных.

Инструмент обеспечивает ведение и настройку правил подписания пользователями экземпляров объекта приложения и их вложений, выбор сценария для выполнения после

подписания экземпляра объекта приложения, а также правил автоматической проверки наличия и валидности ЭП пользователей под экземплярами объекта приложения и их вложениями со стороны системы. Доступно отключение валидации электронной подписи при переводе записи по статусу или статусному переходу.

Выполнение проверок при наложении подписи:

- Обязательность подписания данных при переводе записи по статусу.
- Проверка валидности подписи при переводе записи по статусу.

3.3.10. Производственный календарь

Производственный календарь является инструментом для настройки информации о количестве рабочих, выходных и праздничных дней, а также норме рабочего времени за указанный период.

3.3.11. Управление бизнес-метриками

Инструмент обеспечивает создание метрик для временного хранения значений с целью формирования каких-либо статистических данных по разным параметрам прикладного приложения.

Бизнес-метрики, созданные пользователями, применяются, например, для определения количества изменений за заданный период какого-либо поля в записи объекта приложения, или получения данных по изменению статуса записи.

3.3.12. Генерация последовательностей

Генерация последовательностей является инструментом, обеспечивающим создание и настройку последовательной генерации номера для сущности системы согласно заданному алгоритму и с текстовым сопровождением, что включает в себя:

- создание серии с указанием названия, стартового значения, шага и периода сброса;
- создание номера с указанием скрипта для номера последовательности;
- возможность вызова последовательности из сценария, правила и другой сущности.

3.4. Программный компонент «Отчеты»

Программный компонент обеспечивает возможности создания отчетных форм для приложений, создаваемых на базе «БФТ.Платформы».

Отчетные формы проектируются на основе шаблонов Word, Excel.

Программный компонент позволяет осуществлять:

- Ведение шаблонов отчетов.
- Ведение отчетов в формате: pdf, docx, doc, odt, xlsx, xls, ods, html, rtf, xml, csv.
- Автоматическое формирование записей в Журнале выполнения отчетов.
- Экспорт отчетов в различных форматах с использованием конвертера документов (JodConverter):
  - JXLS: xls и ods – на основе xlsx.
  - XDocReport: pdf и odt – на основе docx.
- Постобработку сформированного отчётным движком документа с использованием скрипта (kotlin).
- Запрещать доступ к встроенному дизайнеру отчетов Stimulsoft через настройку ice.report.stimulsoft.enabled.

3.4.1. Ведение шаблонов отчетов

Ведение шаблонов отчетов включает следующие функции:

- Создание, изменение, удаление шаблона отчета.
- Создание копии шаблона отчета.
- Создание и настройка шаблонов отчетов при помощи визуального конструктора:
  - Использование шаблонов отчетов, подготовленных в следующих редакторах: Stimulsoft, Word, Excel, BIRT (опционально).
    - Настройка диалогового окна параметров отчета.
    - Настройка источников данных для отчета.
    - Настройка формы шаблона отчета.
    - Настройка выгрузки отчетных форм в форматах XLS, XLSX, PDF, DOCX, DOC ODT, ODS, RTF, HTML, CSV и XML.
    - Настройка родителя для отчетной формы.
    - Настройка постобработки документа с использование скрипта Kotlin.
    - Задание шаблона отчета с помощью скрипта.
  - Назначение прав доступа к шаблону отчета:
    - Владелец шаблона отчета.
    - Установление доступа для всех пользователей.

3.4.2. Ведение отчетов

Ведение отчетов включает следующие функции:

- Задание параметров отчета.
- Возможность предварительной настройки параметров отчета, в том числе:
- Заполнение параметров отчета;
- Настройка профиля параметров отчета для автоматического заполнения значений параметров отчета при выборе профиля отчета;
- Одновременное очищение значений параметров отчета и профиля отчета.
- Указание комментария перед формированием отчета.
- Предварительный просмотр отчета.
- Отключение функции предварительного просмотра отчета.
- Запуск на выполнение отчета в преднастроенном формате.
- Выгрузка отчетных форм в форматах XLS, XLSX, PDF, DOCX, DOC, ODT, ODS.
- Уведомление о результате выполнения отчета.
- Печать экранных форм прикладных приложений:
  - печать визуальных форм документов и карточек справочников;
  - печать списков документов и справочников.

3.4.3. Журнал выполнений отчетов

Программный компонент «Отчеты» обеспечивает автоматическую фиксацию действий, выполняемых пользователями в платформе, в Журнале выполнения отчетов. Фиксируется следующая информация об отчетах:

- Реквизиты отчетов.
- Файл с шаблоном отчета.
- Файл с отчетом.
- Логин пользователя, который выполнил отчет.
- Журнал выполнений отчетов позволяет:
  - сохранить отчет на диске;
  - просмотреть состояние выполнения отчета: «Завершён», «Ошибка»;
  - осуществить повторную отправку отчетов;
  - осуществить мониторинг результата отправки;
  - осуществить отправку отчетов по нескольким e-mail адресам;
  - выгрузить несколько выбранных отчетов в файл формата zip-архив;
  - открыть файл отчета в формате pdf в текущей вкладке браузера.

3.5. Программный компонент «Уведомления»

Программный компонент «Уведомления» обеспечивает создание, хранение и отображение информационных сообщений для пользователей системы.

Программный компонент обеспечивает автоматическим информированием пользователя о наличии у него непрочитанных сообщений. У пользователя существует возможность быстрого доступа к непрочитанным сообщения.

Программный компонент «Уведомления» включает выполнение следующих функций:

- Формирование и отправка информационных сообщений выбранным или всем пользователям.
- Хранение информационных сообщений.
- Просмотр информационных сообщений:
  - перечень каналов, по которым распространилось уведомление;
  - время просмотра сообщений по каждому каналу;
  - текст сообщения;
  - приоритет;
  - дата.
- Отправка сообщений:
  - одному пользователю;
  - выбранной группе пользователей;
  - одновременно всем пользователям системы.
- Просмотр состояния сообщений:
  - прочитано;
  - не прочитано.
- Выбор каналов отправки сообщений: Лента, почтовое сообщение, Диалоговое окно, Push-уведомления, СМС, Очередь AMQP.
- Работа с лентой сообщений:
  - отмечать все сообщения как прочитанные;
  - удаление всех или выделенного сообщения;
  - сортировка сообщений по дате создания;
  - фильтрация сообщений по приоритету.
- Создание, изменение, удаление шаблонов сообщений.
- Возможность отправки сообщения от администратора системы с прикреплением файлов внутри платформы.

- Возможность отправки сообщений, настроенных с использованием шаблона:
  - пользователям внутри системы;
  - во внешние системы с помощью REST-сервиса.
- Запись всех созданных в системе уведомлений в Журнал сообщений:
  - Выгрузка перечня сообщений в Excel.
  - Сохранение файлов вложений, если сообщение имеет состояние «Ошибка».
  - Удаление файлов вложений, если сообщение имеет состояние «Отправлено».
  - Указание даты актуальности сообщения.
- Повторная отправка сообщений в Журнале сообщений.
- Мониторинг результата отправки сообщений в Журнале сообщений.
- Автоматическая очистка стека сообщений по истечении заданного количества дней с возможностью настройки даты удаления ранее созданных сообщений.
- Возможность отправки сообщений в техподдержку через главное окно системы.
- Возможность отключения отправки сообщений в техподдержку через главное окно системы (при использовании в application.properties настройку ice.supportSender.manualSend.enabled=false).
- Возможность изменения длительности отображения всплывающих уведомлений через группу настроечных параметров «Длительность отображения сообщений».
- Создание и настройка группы уведомлений:
  - Подписка пользователя на группу уведомлений.
  - Включение или отключение подписки на получение уведомлений по выбранной группе или всем группам сразу.
  - Выбор предпочтительных каналов для получения уведомлений из доступного списка в группе.
- Настройка и использование уведомлений о наличии непрочитанных сообщений.

3.6. Программный компонент «Бизнес-процессы (BPM)»

Программный компонент «Бизнес-процессы (BPM-Business Process Management) предназначен для создания, гибкой настройки, инициирования и администрирования бизнес-процессов, смоделированных в нотации BPMN 2.0 с помощью визуального редактора с функциями:

- управление бизнес-процессами;
- исполнение бизнес-процессов;
- администрирование бизнес-процессов;

- применение диаграмм решений приложения для бизнес-процессов.

Программный компонент предназначен для обеспечения программной поддержки концепции процессного управления путем отражения в системе реальных бизнес-процессов и автоматизации их выполнения.

Возможности программного компонента:

- Моделирование бизнес-процессов в формальной нотации BPMN 2.0 с помощью визуального редактора (как встроенного, так и внешнего).
- Графическое представление моделей настраиваемых бизнес-процессов.
- Исполнение событийной цепочки бизнес-процессов в соответствии с настроенными моделями.
- Мониторинг, контроль и анализ исполнения бизнес-процессов.
- Управление ходом и оптимизация бизнес-процессов путем динамического перестроения моделей.
- Хранение истории в отдельной базе данных.
- Взаимодействие через брокер сообщений Kafka.

Составляющие программного компонента:

- Схемы процессов:
  - Настроенные в редакторе BPMN Схемы исполняемых процессов хранятся в модуле в виде xml-файлов (с возможностью загрузки/выгрузки).
  - Схема процесса представляет собой установленную последовательность отдельных Задач.
  - Выполнение в системе загруженной схемы процесса осуществляется после предварительной настройки свойств для нее и ее задач.
- Процессы:
  - Процессы, которым принадлежат задачи пользователя, отображаются на форме просмотра процессов.
  - Возможна настройка сложных процессов, содержащих один или несколько вложенных процессов.
- Задачи:
  - Задачи процесса могут делегироваться между пользователями.
  - Делегирование задач выполняется менеджером процесса.
  - Задачи, назначенные пользователю, отображаются на форме просмотра задач пользователя.

- Возможность оставлять комментарий к назначенной пользователю задаче.
- Возможность подписания файла вложения в объекте приложения через задачу бизнес-процесса.
- Виды задач:
  - Пользовательские – выполняются пользователями.
  - Системные – выполняются системой автоматически.
  - Цикличные – выполняются с заданной периодичностью:
    - Параллельные.
    - Последовательные.
    - Количество этапов цикличной задачи определяется при настройке процесса.

- Возможности по настройке задач и процессов:
  - Логика обработки данных в задачах процессов:
    - Настраивается с помощью скриптового языка (например, Java Script, Groovy).
    - Позволяет настраивать изменения статусов объектов приложений (документов, справочников) в соответствии с их жизненным циклом.
  - Для пользовательских задач возможна настройка визуальных форм ввода/отображения данных, участвующих в обработке.
  - При настройке модели процесса можно указать:
    - Пользователей или группы пользователей, ответственных за каждый этап выполнения процесса (задачу процесса).
    - Максимальную длительность выполнения процесса и каждого из его этапов.
  - При выполнении процесса возможно:
    - Изменить ответственных пользователей/группы пользователей.
    - Изменить ранее установленную длительность процесса и каждого из его этапов.
    - Динамически перестроить выполняемые модели процессов, вне зависимости от наличия незавершенных процессов.
- Средства контроля исполнения процессов:
  - Позволяют собирать и анализировать статистическую информацию:
    - время выполнения процессов;
    - степень завершенности выполнения текущих процессов.

  - Представляют аналитическую информацию по исполнению процессов в графической форме.
  - Для представления информации используется механизм инфопанелей.

- Участники процессов:
  - Менеджер процесса – управляет схемой и выполнением процесса.
  - Создатель процесса – является автором процесса.
  - Инициатор процесса – запускает начало процесса.
  - Исполнитель задачи – выполняет назначенную задачу процесса.
- Функции программного компонента:
  - Управление бизнес-процессами (п. 3.6.1).
  - Конструирование схем и моделей данных бизнес-процессов (п. 3.6.2).
  - Исполнение бизнес-процессов (п. 3.6.3).
  - Администрирование процессов (п. 3.6.4).
  - Конструирование Диаграммы решений приложения для бизнес-процессов (п. 3.3.53.6.5).

3.6.1. Управление бизнес-процессами

Управление бизнес-процессами включает возможности:

- Навигации по дереву процессов. По умолчанию модули в навигаторе отсортированы в алфавитном порядке.
- Фильтрация в дереве процессов.
- Просмотра модулей, групп и процессов в виде дерева.
- Поиска процессов:
  - По наименованию;
  - По значению переменной контекста процесса;
  - По переменным задачи процесса.
- Просмотра избранных процессов.
- Создания и редактирования процесса.
- Копирование процесса.
- Задания основных свойств процесса:
  - Модуль, группа, идентификатор, отображаемое имя.
- Настройки контекста процесса – выбор заранее подготовленного контекстного объекта приложения.

- Настройка объекта процесса – выбор атрибута контекстного объекта приложения с типом Ссылка.
- Открытие контекста в отдельной вкладке браузера, при нажатии на наименование контекста процесса.
- Настройки процесса в визуальном моделере:
  - Возможность применения брокеров для получения и отправки сообщений.
- Настройки процесса скриптом (без моделера).
- Просмотра версионности процесса.
- Сохранения, публикации процесса.
- Копирования процесса вместе с ОП контекста.
- Запуска процесса.
- Экспорта и импорта схем процесса.
- Удаления процесса (со всеми связанными задачами и историей запуска).
- Валидация сохраняемого и публикуемого процесса.
- Запуск симуляции токенов для проверки логики работы бизнес-процесса.
- Запуск перекомпиляции всех скриптов, используемых в бизнес-процессах.

3.6.2. Конструирование схем и моделей данных бизнес-процессов

Конструирование схем и моделей данных бизнес-процессов включает следующие функции:

- Настройка процессов.
- Создание BPMN-схемы процесса во встроенном редакторе:
  - Настройка типов элементов BPMN-схемы процесса.
  - Заполнение свойств элементов процесса:
    - Идентификатор, Наименование, Ответственный, Пользователи-кандидаты, Группы-кандидаты, Срок исполнения, Дата наблюдения, Приоритет, Формат скрипта, Тип скрипта, Скрипт, Тип перехода, Выражение, Идентификатор процесса, Пользовательское наименование процесса,
    - Настройка параметров слушателей (Слушатель выполнения, Тип события, Тип слушателя (Java-класс, выражение, выражение делегата, скрипт), Внедрение полей, Локальное.

- Настройка параметров Ввод/Вывод.
- Автоматическое заполнение полей Модуль и Группа. Происходит, если у пользователя предварительно открыт какой-либо бизнес-процесс. При создании нового бизнес-процесса поля Модуль и Группа заполняются теми же значениями, что у текущего открытого бизнес-процесса.

- Автоматическое заполнение свойств Идентификатор, Наименование пользовательской задачи после выбора формы редактирования.
- Повторный вызов ранее выбранной формы редактирования для пользовательской задачи.

- Переход из моделера процесса в подпроцесс по клику.
- Создание BPMN-схемы процесса в стороннем редакторе.
- Создание BPMN-схемы процесса в визуальном редакторе Camunda Modeller:
  - Задание свойств элементов для BPMN-схемы.
  - Сохранение схемы в файл с расширением *.bpmn.
  - Создание новой записи и сохранение ее в редакторе Процессы приложения.
  - Загрузка в созданную запись BPMN-схемы процесса.
  - Загрузка и выгрузка BPMN-схемы процесса.
  - Загрузка BPMN-схемы процесса выполняется в сохраненную запись описателя процесса.
  - Просмотр и редактирование загруженной BPMN-схемы.
  - Редактирование BPMN-схемы.
  - Загрузка новой версии BPMN-схемы процесса.
  - Выгрузка BPMN-схемы процесса в файл.
  - Настройка форм и переменных процесса.
  - Создание контекстного объекта приложения.
  - Автоматическое формирование формы редактирования задачи процесса.
- Просмотр дополнительной информации на диаграммах через специальные комбинации клавиш (наличие входных-выводных переменных, слушателей и т.д.):
  - Alt + Y – отображение основных элементов.
  - Alt + T – отображение транзакций и приоритетов.
  - Alt + X – отображение идентификаторов элементов и т.д.
- Публикация процесса:
  - Публикация нового процесса.
  - Публикация новой версии процесса.
  - Просмотр перечня опубликованных версий процесса.

- Выполнение проверок при сохранении процесса приложения:
- Проверка наличия в контекстом объекте приложения формы редактирования с идентификатором пользовательской задачи в процессе.
- Проверка заполнения наименования переменной для элемента Событие-условие.
- Проверка применения в выражениях процесса приложения атрибутов, которые есть в контекстном объекте приложения.
- Проверка на наличие выходного потока у элементов процесса приложения, кроме элемента Конечное событие (End Event).
- Проверка на количество входящих и исходящих потоков для шлюза «или/или» (Исключающий шлюз (эксклюзивный, Exclusive Gateway), «или/или» – выбор только одного пути).
- Проверка на включение флага Асинхронное возобновление – после для элементов Расходящаяся развилка и Шлюз «И».
- Проверка на заполнение потока по умолчанию для шлюза «или/или» (Исключающий шлюз (эксклюзивный, Exclusive Gateway), «или/или» – выбор только одного пути).
- Проверка на наличие включенного признака Асинхронно после в пользовательской задаче или событии с получением сообщения.
- Проверка на наличие входного потока у элементов процесса приложения, кроме элемента Стартовое событие (Start Event).
- Проверка на наличие включенного признака Асинхронно после в свойствах пользовательской задачи.
- Проверка на корректность заполнения поля Ответственный в свойствах пользовательской задачи.
- Проверка наличия в процессе нескольких подряд задач, назначенных на одного и того же пользователя.
- Проверка наличия граничных не прерывающих событий-условий.
- Проверка наличия наименования для каждого элемента схемы процесса.
- Проверка определения входящих переменных при вызове внешнего подпроцесса (call activity).
- Проверка на количество входящих потоков для всех задач/шлюзов/событий, кроме шлюза «или/или».
- Проверка на количество исходящих потоков для всех задач/шлюзов/событий, кроме шлюза «или/или».
- Проверка на наличие в процессе цикла, который возвращается в шлюз "И".

- Проверка на наличие внешнего подпроцесса (call activity), для которого не определены исходящие переменные.

- Валидация в пользовательских задачах процесса с помощью сценария.

3.6.3. Исполнение бизнес-процессов

Для исполнения процессов обеспечиваются следующие возможности:

- Запуск процесса (последней или выбранной версии).
- Просмотр перечня запущенных процессов.
- Назначение задач пользователям.
- Исполнение задачи пользователем, включая возможность просмотра исполненных задач.
- Завершение исполнения процесса, с возможностью просмотра перечня завершенных процессов.
- Изменение значения переменных контекста процесса.
- Подписание вложений бизнес-процесса.

3.6.3.1. Назначение задач пользователям

При назначении задач пользователям обеспечиваются следующие возможности:

- Автоматическое назначение задачи пользователю (через настройку BPMN-схемы Assignee – Ответственный) – исполнитель задачи.
- Настройка возможности исполнять и назначать задачу одному и тому же пользователю:
  - Заполнены поля Ответственный, Пользователи-кандидаты, Группы-кандидаты, Ответственный может как исполнять, так и назначать задачу пользователям из кандидатов.
- Настройка возможности назначать задачу для исполнения себе из списка Задачи на исполнении, если ты входишь в список:
  - Пользователи-кандидаты, Группы-кандидаты.
- Просмотр перечня задач, доступных для исполнения пользователю.
- Задачи, доступные по списку пользователей по настройке BPMN-схемы Candidate Users.
- Задачи, доступные по списку пользователей по настройке BPMN-схемы Candidate Groups (в т.ч. для всех пользователей по настройке "*").
- Взятие задачи на исполнение.

- Переназначение задачи на другого исполнителя.
- Выбор нескольких задач и назначение перечню пользователей.
- Просмотр перечня задач, назначенных пользователю.
- Отправление уведомления только инициатору бизнес-процесса, если его роль указана в поле Группы-кандидаты.
- Обработка одновременной работы с одной пользовательской задачей для одного и нескольких пользователей.

3.6.4. Администрирование бизнес-процессов

Возможность администрирования бизнес-процессов включает:

- Управление исполнением процессов.
- Просмотр перечня активных экземпляров процессов.
- Просмотр перечня активных экземпляров задач.
- Просмотр истории запуска процессов.

3.6.4.1. Управление исполнением процессов

В части управления исполнением процессов программный компонент позволяет осуществлять:

- Переход из формы задачи пользователя к окну инстанса бизнес-процесса, который породил эту задачу.
- Для активных экземпляров процессов:
  - поиска инстансов процессов и задач по значению переменной контекста;
  - возможность скрыть поля для быстрого поиска инстансов процессов по значению переменной контекста;
  - приостановки исполнения экземпляра процесса;
  - остановки исполнения экземпляра процесса при наличии у пользователя прав на удаление экземпляров процессов;
  - миграции экземпляра процесса между версиями процесса;
  - миграции с обновлением триггеров;
  - миграции без обновления триггеров;
  - миграция экземпляра процесса с выбранного узла;
  - возможность указывать необходимость обновления контекстных переменных при подготовке процесса к миграции.
- В части истории запуска процессов:

- возобновления исполнения приостановленного экземпляра процесса;
- удаления завершенного экземпляра процесса;
- удаления остановленного экземпляра процесса.

- Для всех активных задач:
  - поиск инстансов процессов и задач по значению переменной контекста;
  - назначение задач пользователям, зарегистрированным в системе.

3.6.4.2. Просмотр перечня активных экземпляров процессов

В части просмотра информации об экземпляре процесса программный компонент позволяет осуществлять:

- Просмотр перечня активных задач.
- Просмотр значения переменных процесса и их изменение.
- Просмотр диаграммы бизнес-процесса.
- Просмотр перечня системных задач.
- Просмотр перечня задач-сервисов.
- Просмотр перечня подпроцессов.
- Просмотр перечня пройденных этапов процесса. Список значений переменных контекста запрашивается только один раз, а также максимальное количество запрашиваемых переменных ограничен 10.000 строк.
- Создание и отправку сообщения в бизнес-процесс.
- Открытие диаграммы запущенной версии бизнес-процесса в отдельной вкладке браузера.
- Просмотр URL с идентификатором экземпляра просматриваемого процесса.
- Выполнять переход в Camunda Cockpit/Excamad с формы просмотра информации об экземпляре процесса.

3.6.4.3. Просмотр истории запуска процессов

Возможность просмотра истории запуска бизнес-процессов:

- Значения переменных процесса.
- Перечень пройденных этапов процесса.
- Возобновление процессов с состоянием «Остановлено внешне» или «Остановлено внутренне».

3.6.4.4. Просмотр перечня активных экземпляров процесса при отсутствии исторических данных Camunda

В части просмотра информации об экземпляре процесса программный компонент позволяет осуществлять просмотр:

- перечня активных задач;
- значения переменных процесса и их изменение;
- диаграммы бизнес-процесса;
- перечня системных задач;
- перечня задач-сервисов;
- перечня подпроцессов;
- перечня пройденных этапов процесса.

3.6.4.5. Просмотр статистики всех опубликованных версий бизнес-процессов.

В части просмотра информации об статистике бизнес-процесса программный компонент позволяет осуществлять просмотр:

- информации по действующим, приостановленным, завершенным экземплярам;
- информацию по инцидентам;
- процентное соотношение эффективности версий процессов.

3.6.4.6. Управление инцидентами.

В части управления инцидентами бизнес-процессов функции позволяют:

- Регистрировать инциденты. При возникновении инцидентов данные о них сохраняются в таблице. Данная таблица позволяет:
  - Мониторить ошибки, возникающие при работе запущенных экземпляров процессов;
  - Просматривать детальную информацию об инциденте;
  - Разрешать инциденты;
  - Удалять ошибочные экземпляры процессов.
- Перезапускать экземпляры процессов.
- Удалять экземпляры процессов.
- Формировать отчет по инцидентам в формате xlsx.
- Выполнять переход в Camunda Cockpit/Excamad с формы просмотра информации об инциденте.

3.6.5. Применение диаграмм решений приложения для бизнес-процессов

В части применения диаграмм решений приложения программный компонент позволяет выполнять:

- Настройку описателей диаграмм.
- Создание диаграммы решений приложения.

3.6.5.1. Настройка описателей диаграмм

- Возможность указывать основные свойства диаграммы:
  - Модуль;
  - Уникальный ключ диаграммы;
  - Отображаемое имя;
  - Группа;
  - Последняя опубликованная версия;
  - Описание;
  - Графический редактор, позволяющий моделировать схемы диаграммы;
  - Перечень опубликованных версий описателя процесса;
  - Поля;
  - Отображаемое имя.
- Возможность просмотра последней опубликованной версии диаграммы:
  - Графическое представление схемы описателя диаграммы выбранной версии;
  - Xml-представление схемы диаграммы;
  - Форма списка диаграмм.
- Возможность выполнять действия с диаграммой:
  - Экспорт схемы;
  - Добавление диаграммы;
  - Удаление диаграммы;
  - Фильтрация диаграмм;
  - Поиск диаграммы;
  - Сохранение диаграммы;
  - Публикация диаграммы;
  - Просмотр версии диаграммы.
- Возможность изменения описателя диаграммы:
  - Изменение схемы диаграммы:
  - В графическом редакторе или загрузка новой версии схемы;
  - Загрузка схемы диаграммы с машины пользователя;
  - Выгрузка схемы диаграммы на машину пользователя;

- Копирование диаграммы.

3.6.5.2. Создание диаграммы решений приложения

- Возможность настройки типов элементов схемы диаграммы:
- Настройка связей между элементами схемы диаграммы.
- Свойства элементов диаграммы:
  - Свойства элемента Решение.
- Идентификатор БД.
- Отображаемое имя.
- Политика обращения.
- Входные данные:
  - Выбор типа входных данных;
  - Добавление новых входных данных;
  - Удаление входных данных.
- Выходные данные:
  - Выбор типа выходных данных;
  - Добавление новых выходных данных;
  - Удаление выходных данных.
- Комментарий.

3.7. Программный компонент «Отчеты Стимулсофт»

Программный компонент обеспечивает возможности создания отчетных форм для приложений, создаваемых на базе «БФТ.Платформы».

Программный компонент предоставляет возможности:

- Получать данные из объектов приложений без использования скриптов.
- За счёт большого функционала программы Stimulsoft:
  - Выводить несколько записей объектов приложения.
  - Фильтровать записи.
  - Сортировать записи по возрастанию/по убыванию.
  - Группировать записи.
- Использовать дополнительные параметры для построения отчета.
- Выполнять предпросмотр отчета во время формирования шаблона отчета.
- Создания переменных в шаблоне отчета.

- Написания SQL-запросов.
- Использования встроенных секций.
- Добавления пользовательских шрифтов.
- Выполнять экспорт отчета в форматах xls и ods – на основе xlsx (опционально), odt на основе docx (опционально).

3.8. Программный компонент «Отчеты BIRT»

В «БФТ.Платформе» шаблонизатор BIRT позволяет создавать шаблоны отчетов с форматом BIRT и является отдельным модулем. Программный компонент позволяет:

- Создавать отчеты различных форматов, включая PDF, HTML, Excel, CSV и другие.
- Создавать отчеты без глубоких технических знаний благодаря интуитивно понятному интерфейсу.
- Легко встраиваться в Java-приложения (интегрирован с Java).
- Поддерживать различные источники данных.
- Создавать сложные отчеты с диаграммами, таблицами и графиками.
- Использовать плагины, API, BIRT для расширения и адаптирования под конкретные задачи.
- Выполнять экспорт отчета в форматах xls и ods – на основе xlsx (опционально), odt на основе docx (опционально).

3.9. Программный компонент «Виджеты»

Программный компонент представления аналитических данных предназначен для моментального перевода больших объемов необработанной информации в осмысленную, удобную для восприятия форму в виде аналитических представлений – рабочей панели виджетов.

Рабочая панель – представляет собой страницу с настроенным по требованиям пользователя набором виджетов, а в режиме администратора на панели доступны все инструменты по их настройке: список всех типов виджетов, источников данных и параметров.

- Отслеживание состояния рабочей панели:
  - Черновик.
  - Опубликована.
  - Редактируется.
  - Архив.
- Работа с источниками данных рабочей панели:

  - SqlSource (SQL-запрос);
  - AppObjSource (Объект приложения);
  - LazyAppObjSource (Объект приложения (без выборки));
  - ScriptSource (Скрипт).
- Использование параметров для фильтрации данных:
  - Булевский.
  - Строковый.
  - Целочисленный.
  - Длинный целочисленный.
  - Десятичный.
  - Дата.
  - Дата со временем.
  - GUID.
  - Ссылка.
  - Ссылка на системный объект.
- Отображение данных источников с помощью виджетов:
  - Гистограмма;
  - Горизонтальная гистограмма;
  - Диаграмма разброса;
  - Древовидная диаграмма;
  - Линейный;
  - Пирог;
  - Полярная диаграмма;
  - Пончик;
  - Пузырьковый;
  - Радар;
  - SpeedometerWidget (Спидометр);
  - GanttWidget (Диаграмма Ганта);
  - BpmWidget (Задачи BPM);
  - BpmButtonWidget (Запуск BPM-процесса);
  - ImageWidget (Изображение);
  - CalendarWidget (Календарь);
  - Календарь событий;
  - MapWidget (Карта со статистикой);
  - ObjCardWidget (Карточка объекта);
  - TabsWidget (Контейнер);

  - ProgressWidget (Полоса прогресса);
  - EmptyWidget (Пустой);
  - DashboardWrapWidget (Рабочая панель);
  - TimelineWidget (Расписание);
  - BlockWidget (Список карточек);
  - Чек-лист;
  - TableWidget (Таблица ОП);
  - TextWidget (Текст);
  - TrendWidget (Тренд);
  - ClockWidget (Часы).
- Создание палитр.
- Управление библиотекой виджетов;
- Настройка размещения;
- Настройка слоев и действий;
- Настройка легенды;
- История изменений с возможностью отката.

3.10. Программный компонент «Многофункциональный компонент для работы с табличными данными»

По умолчанию, в программном комплексе «Базовый состав платформы» используются компоненты библиотеки Ant.Design, в том числе для отображения таблиц.

Программный компонент «Многофункциональный компонент для работы с табличными данными» обеспечивает возможность настройки формы списка вида AG Grid для объекта приложения и расширяет базовые возможности, добавляя функционал:

- группировки данных;
- применения большого набора встроенных фильтров;
- создания собственных редакторов ячеек;
- сложной сортировки больших баз данных;
- сортировки по нескольким колонкам;
- обновления данных в режиме реального времени;
- скролл с подгрузкой данных вместо пагинации;
- автоподбор ширины колонок;
- объединение ячеек с одинаковыми значениями.

Более детально, программный компонент обеспечивает также:

- Автоподбор ширины колонки.
- Автоподбор ширины всех колонок.
- Автоподбор ширины всех колонок во всех таблицах.
- Закрепление колонок слева/справа.
- Возможность выравнивания на форме списка значений полей записей:
  - По горизонтали.
  - По вертикали.
- Объединение одинаковых значений.
- Показывать итоги по всем колонкам.
- Группировки по колонке.
- Разделение экрана на списочную форму и форму редактирования.
  - Открытие формы редактирования записи двойным кликом.
- Настройка отображения колонок на форме списка через панель инструментов:
  - Управление отображением колонок на форме списка.
  - Настройка видимости и порядка отображения колонок на форме списка.
  - Сброс настроек формы списка к настройкам, применяемым по умолчанию.
- Группировать колонки под одним заголовком и задавать наименование группе колонок на форме списка.
- Расчет нескольких итогов по колонкам одновременно в контекстном меню столбца формы списка.
- Настройка визуального отображения чередующихся строк.
- Сохранение профиля фильтра.
- Включение/отключение видимости всех колонок, кроме колонки по умолчанию.
- Группировка перечня действий в контекстном меню.
- Отображение данных согласно маске ввода.
- Настройка группировки действий объекта приложения в контекстном меню.
- Отображение на форме списка иерархического справочника в поле **Всего** общего количества записей без учета иерархии или общего количества записей, попадающих под условия фильтрации.
- Управление моментом загрузки записей на форму списка.
- Указывать несколько значений для фильтрации по одной колонке.

- Переносить значение в колонках с типом Строковый, Ссылка и Ссылка на системный объект по словам.
- Управление кнопкой «Обновить» для обновления списка записей в компоненте «Список».
- Одновременное перемещение нескольких колонок формы списка для изменения порядка следования без группировки.
- Отображение счетчика добавленных значений в окне фильтрации для составного фильтра.
- Возможность открытия записи в новой вкладке браузера.
- Возможность переопределить действие «Снять выделение со всех записей».
- Возможность выбрать вид группировки для экспорта записей в формате xlsx.
- Настройка оформления строки группы при экспорте записей в формате xlsx (шрифт, заливка ячеек, стиль границы ячеек).

3.11. Программный компонент «Импорт данных из файлов»

Программный компонент «Импорт данных из файлов» обеспечивает импорт данных из файлов форматов JSON, XML, CSV, XLSX со следующими возможностями:

- Добавление, изменение, удаление шаблонов импорта данных.
- Настройка параметров шаблонов импорта данных:
  - Формат файла:
    - XLSX.
    - JSON.
    - CSV.
    - XML.
  - Наименование шаблона импорта данных.
  - Ссылка на ОП, для которого настроен шаблон.
  - Возможность импортировать данные в коллекцию записи:
    - Выбор записи объекта приложения.
    - Выбор поля коллекции.
  - Наименование листа в файле Excel, откуда выгружаются данные.
  - Номер строки заголовков колонок из листа файла Excel.
  - Номер первой строки данных из листа файла Excel.
  - Узел элемента из XML файла.
  - Возможность игнорирования пустых строк в файле Excel при импорте из Excel.

- Возможность прервать импорт из Excel при обнаружении первой пустой строки.
- Возможность обновлять отображаемое имя записи при повторной загрузке данных, чтобы измененные данные корректно отображались. Возможность исчезает при активации признака Импорт данных в коллекцию записи.
- Возможность осуществить следующие действия при нахождении невалидных данных:
  - Обработка записей пачками.
  - Обработка записей построчно.
- Возможность осуществить следующие действия при нахождении некорректных данных в файле Excel:
  - Остановить загрузку файла.
  - Пропускать загрузку поля.
  - Пропускать загрузку записи.
- Определение правил конвертации данных:
  - Колонка/ поле файла.
  - Поле.
  - Отображаемое имя.
  - Ключевое поле.
  - Импортировать.
  - Объект приложения.
  - Условия.
- Обновление списка правил конвертации данных из файла EXCEL.
- Обновление списка правил по данным из ОП.
- Настройка условий для поиска ссылок.
- Настройка приема данных:
  - Синхронизация данных.
  - Перезапись данных.
  - Сохранение результатов загрузки в файл Excel.
- Загрузка файла Excel с данными в шаблон.
- Импорт данных в справочник «БФТ.Платформы».
- Формирование файла-шаблона в Excel на основе правил конвертации существующего в «БФТ.Платформе» шаблона.

- Просмотр результата выполнения импорта данных из файла Excel в объект приложения через Журнал импорта данных.
- Доступ к импорту данных из объекта приложения.

  - На форме редактирования записи справочника.
  - На форме списка справочника.
- Импорт ссылочных полей.

3.12. Программный компонент «Лицензирование»

Программный компонент «Лицензирование» предназначен для создания лицензий с выделенными правами доступа к объектам приложения и пунктам меню рубрикатора. Лицензии создаются с помощью менеджера лицензий. После подключения модуля «Лицензирования» созданные ограничения на объекты приложения и атрибуты не изменяют ролевую модель в базе данных.

Для создания лицензии указываются:

- Параметры:
  - Id лицензии.
  - Клиент.
  - Прикладное приложение, на основе «БФТ.Платформы».
  - Дата активации лицензии.
  - Срок действия лицензии (в днях).
  - Количество допустимых пользователей.
  - Версия «БФТ.Платформы».
  - Набор флагов (список текстовых значений, которые можно использовать в коде прикладного приложения для регулирования доступа к функционалу этого приложению).
  - Дата создания лицензии.
- Конфигурирование объектов предназначено для разграничения доступа к модулям объектов приложения, к объектам приложения и атрибутам объекта приложения.
  - Импорт перечня модулей, объектов приложений и их атрибутов с прикладного стенда в формируемую лицензию.
- Конфигурирование пунктов меню предназначено для разграничения доступа к пунктам меню рубрикатора.
- Лицензии доступны для создания из табличного вида просмотра.
- Доступно создание групп лицензий.
- Проверка валидности загружаемого файла лицензии.

3.13. Программный компонент «Интеграция взаимодействий с другими приложениями»

Интеграция – это инструмент «БФТ.Платформы», который позволяет реализовывать обмен сообщениями в режиме реального времени между приложением, созданным на основе БФТ. Платформы, и удаленным сервисом с помощью брокера сообщений Apache Kafka. Процесс интеграции запускается с использованием сценариев.

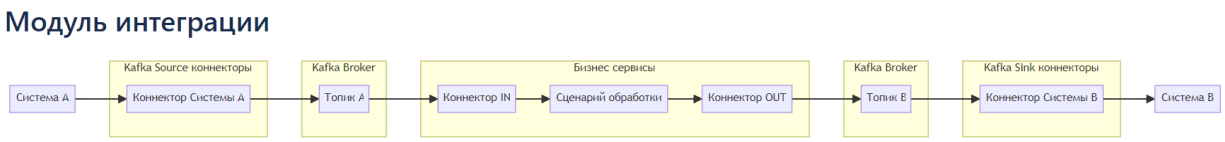


Рисунок 1. Схема работы модуля интеграции

Apache Kafka – это распределенная платформа для обработки потоковых данных в режиме реального времени. Она обеспечивает быстрый, хорошо масштабируемый и надежный обмен сообщениями по модели pub-sub.

Модель публикации-подписки (pub-sub) – это подход в проектировании, при котором отправители сообщений, называемые производителями (Producer), не отправляют сообщения прямо получателям, известным как потребители (Consumer). Вместо этого сообщения публикуются в темах (topics) без учёта того, кто именно их получит. Потребители же подписываются на интересующие их темы и получают оттуда сообщения.

Характеристики Apache Kafka:

- Масштабируемость: Apache Kafka имеет распределенную архитектуру, обрабатывающую входящие сообщения с большим объемом и скоростью. Как результат – Kafka обладает высокой масштабируемостью без каких-либо простоев.
- Высокая пропускная способность: Apache Kafka может обрабатывать миллионы сообщений в секунду. Сообщения, поступающие в большом объеме или с высокой скоростью не влияют на производительность.
- Низкая задержка: Kafka обеспечивает очень низкую задержку передачи, которая составляет около десяти миллисекунд.
- Отказоустойчивость: Используя механизм репликации, Kafka обрабатывает сбои на узлах кластера без потери данных.
- Надежность: Apache Kafka – это распределенная платформа с очень высокой отказоустойчивостью, что делает ее очень надежной в использовании системой.
- Сохранность данных: Реплики данных, хранящихся в кластере Kafka, распределены по разным серверам, при падении или сбое одного сервера копии остаются на других.
- Обработка данных в реальном времени.

3.13.1. Основные понятия интеграции

Программный компонент обеспечивает реализацию следующих функциональных возможностей:

- Создание каналов:
  - Выбор типа каналов:
    - Отправка.
    - Получение.
    - Получение и отправка.
- Создание коннекторов:
  - Выбор типа коннекторов:
    - Kafka.
    - Таблица БД.
    - Мост.
- Запуск работы коннекторов.
- Остановка работы коннекторов.
- Отслеживание состояния коннекторов.
- Установка режима пакетной обработки для ускорения процесса обработки сообщений.
- Загрузка схемы данных для работы коннектора: JSON, Avro, Protobuf, JSON Schema, Avro Schema, Protobuf Schema.
- Журнал сообщений:
  - Просмотр данных отправленных сообщений.
  - Необработанные сообщения:
    - Просмотр информации об ошибке необработанных сообщений.
    - Запуск повторной обработки сообщений.
  - Очередь сообщений:
    - Просмотр данных сообщений коннекторов с типом Таблица БД.
    - Ручная обработка сообщений.
- Диаграмма:
  - Визуализация потоков данных, передаваемых через каналы и их составляющие:
    - Визуализация потоков данных через метки, содержащиеся в каналах.
    - Визуализация потоков данных через выбор конкретного канала.
- Kafka Connect:

  - Просмотр внешних коннекторов.
  - Запуск внешних коннекторов.
  - Остановка внешних коннекторов.
  - Просмотр состояния внешних коннекторов.
  - Скачивание и загрузка конфигурации внешних коннекторов.
  - Создание внешних коннекторов.
- Топики Kafka:
  - Автоматическое создание топика.
  - Ручное создание топика.
  - Просмотр сообщений топика.
- Сообщения Kafka:
  - Просмотр всех сообщений, содержащихся в выбранном топике.
  - Удаление значений сообщений.

3.14. Программный компонент «Управление интеграцией с MDM»

Управление интеграцией с MDM является программным компонентом по настройке и управлению интеграцией между конфигуратором и MDM.

Программный компонент предназначен для обмена структурами и данными справочников между MDM и Конфигуратором, а также для последующей синхронизации данных при изменении записей справочника в одной из систем.

Программный компонент «Управление интеграцией с MDM» обеспечивает следующие возможности по настройке интеграции:

- Интеграция – первоначальная загрузка структуры справочника из MDM в Конфигуратор;
- Публикация – создание таблицы базы данных для справочника в Конфигураторе;
- Загрузка начального решения – первоначальная загрузка данных (набора исходных записей) справочника из MDM в Конфигуратор;
- Синхронизация данных – загрузка данных (набора записей) справочника при изменении данных справочника в одной из систем;
- Перезагрузка данных – обновление данных (набора записей) справочника в одной из систем;
- С использованием брокера сообщений Kafka или ActiveMQ.

3.15. Программный компонент «Полнотекстовый поиск»

Программный компонент «Полнотекстовый поиск» на основе OpenSearch является мощным инструментом для индексации и поиска информации в структурированных и неструктурированных данных, включая текстовые документы и бинарные файлы.

Возможности программного компонента:

- Данные автоматически индексируются в OpenSearch при старте программы.
- Для бинарных файлов используется Ingest Attachment для извлечения текста.
- Результаты сортируются по релевантности и возвращаются частями.
- Фоновая индексация без остановки работы системы.
- Поиск без учета регистра с поддержкой специальных символов.
- Поиск по всем индексированным полям объектов.
- Извлечение и поиск по содержимому PDF, DOC, XLS и других форматов.
- Горизонтальное масштабирование и многопоточность.
- Поддержка нечеткого поиска и специальных символов.

Программный компонент «Полнотекстовый поиск» включает инструмент семантического поиска на основе BERT и OpenSearch, который понимает смысл запросов и находит документы, релевантные по содержанию, а не только по ключевым словам.

Возможности инструмента семантический поиск:

- Поиск информации по смыслу, а не только по точному совпадению ключевых слов.
- Возможность находить релевантный контент даже при использовании различных формулировок.
- Преобразование текстовых данных в числовые векторы с помощью BERT-модели.
- Создание плотных векторных представлений, сохраняющих семантические связи.
- Многоуровневая обработка текста через кодер BERT.
- Сохранение контекстных и смысловых связей в векторном пространстве.
- Автоматический подбор релевантных поисковых запросов.
- Преобразование пользовательского запроса в векторное представление.
- Работа с заранее подготовленными индексами.
- Эффективный поиск по векторным представлениям.
- Использование специального векторного поля для семантического поиска.
- Работа с большими объемами данных через OpenSearch.

- Поддержка распределенной архитектуры поиска.
- Предварительная обработка и индексация контента.
- Создание оптимизированных структур для быстрого семантического поиска.

3.16. Программный компонент «Сервис электронной подписи»

Программный компонент «Сервис электронной подписи» предоставляет криптографические функции для работы с электронными подписями и сертификатами. Программный компонент обеспечивает взаимодействие между клиентским приложением и криптопровайдерами (КриптоПро и VipNet), выполняя сложные криптооперации в защищенном контуре.

Возможности программного компонента:

- Создание квалифицированной электронной подписи с использованием:
  - КриптоПро CSP.
  - VipNet CSP.
- Подпись различных типов данных и документов.
- Удаление электронной подписи.
- Поддержка разных алгоритмов шифрования.
- Проверка корректности и подлинности ЭП и просмотр результата проверки.
- Подтверждение целостности подписанных данных.
- Проверка с использованием:
  - КриптоПро CSP.
  - VipNet CSP.
- Генерация хэш-сумм для данных:
  - С использованием КриптоПро.
  - С использованием VipNet.
- Обеспечение целостности передаваемой информации.
- Управление цепочками сертификатов.
- Проверка валидности сертификатов.
- Получение информации о сертификате:
  - Субъект и издатель.
  - Срок действия.
  - Назначение и ограничения.
- Проверка сертификатов по спискам отзыва.

- Актуализация списков отзыва.
- Верификация неотозванных сертификатов.
- Получение информации о сертификате TSA.
- Проверка временных меток.
- Валидация штампов времени.
- Обработка запросов от клиентских приложений.
- Унифицированный API для различных криптопровайдеров.
- Стандартизированные форматы запросов и ответов.
- Защищенное выполнение криптографических операций.
- Изоляция критических операций на серверной стороне.
- Минимизация рисков на клиентских рабочих местах.

3.17. Программный компонент «Машиночитаемые доверенности»

Программный компонент «Машиночитаемые доверенности» предназначен для создания, обработки и хранения электронных доверенностей, оформленных в структурированном формате.

Программный компонент обеспечивает выполнение следующих функций:

- Загрузка машиночитаемых доверенностей (МЧД) в формате zip и xml;
- Загрузка нескольких файлов МЧД;
- Проверка валидности сертификатов и электронной подписи;
- Валидация файла МЧД;
- Уведомление о некорректном формате при загрузке сертификатов;
- Создание новых МЧД;
- Удаление МЧД, которым не осуществлялось подписание данных;
- Редактирование записи МЧД пользователям с ролью «Супер пользователь» и «Администратора системы» при отсутствии подписания данных выбранной МЧД;
- Сортировка и фильтрация списка записей;
- Экспорт записей справочников в файл формата xlsx;
- Отзыв доверенностей;
- Автоматическое создание записей об участниках;

- Отсутствие возможности ручного добавления участников или редактирования существующих;
- Отображение ссылок на связанные с участниками МЧД выданные доверенности (для доверителей) и полученные полномочия (для представителей);
- Создание и удаление записей о кодах системы;
- Работа с несколькими типами полномочий, ограничениями полномочий, значениями ограничений полномочий (машиночитаемые, человекочитаемые);
- Ручное и автоматическое создание полномочий, ограничений полномочий, значений ограничений полномочий;
- Удаление полномочий, ограничений полномочий, значений ограничений полномочий;
- Просмотр списка регламентов ЭП, использующих полномочия;
- Подписание экземпляров объекта приложения с использованием МЧД;
- Создание регламента ЭП с использованием МЧД;
- Просмотр сведений о машиночитаемой доверенности, используемой для подписания экземпляра объекта приложения, на форме просмотра подписи, открытой из списка ЭП для записи объекта приложения;
- Отключение запроса машиночитаемой доверенности в правилах регламента ЭП;
- Изменение наименований системных статусов;
- Создание статусов вручную;
- Удаление пользовательской записи статуса;
- Создание источников данных;
- Невозможность редактирования системных источников (загруженных из МЧД, вручную и из ЕСНСИ).
- Возможность включать проверку имени файла МЧД на этапе загрузки.
- Ограничение списка доступных расширений файлов машиночитаемой доверенности, загружаемых в раздел «Реестр МЧД».
- Возможность определять, подписывает электронный документ руководитель организации или нет.
- Проверка наличия заполненного элемента «Передов» в файле доверенности.
- Проверка соответствия кода и наименования полномочия в загружаемом файле МЧД с записями в справочнике Полномочия.

3.18. Программный компонент «Компонент электронных таблиц»

Программный компонент «Компонент электронных таблиц» предназначен для расчета показателей (разных значений полей) в объектах приложения (справочниках), а также хранения рассчитанных показателей и их применения в дальнейших расчетах.

Возможности компонента:

- Создание, изменение и удаление многострочных таблиц.
- Добавление, изменение удаление колонок таблиц.
- Сортировка и настройка последовательности расположения колонок, восстановление настроек при открытии расчетной таблицы.
- Автоматическое формирование заголовка расчетной таблицы.
- Добавление справочных колонок, заранее заполненных информацией.
- Применение формул и функций.
- Подбор значений параметров, применяемых для расчета суммы в колонке с последующим распределением полученной суммы между всеми ячейками без остатка.
- Отображение итоговой строки с автоматическим расчетом итога по колонкам разных форматов.
- Закрепление области колонок.
- Автосохранение расчетных таблиц.
- Выделение отдельной ячейки, группы ячеек и отдельных колонок и строк.
- Форматирование ячеек (границы и заливка, свойства ячеек).
- Добавление дополнительных листов для расчета. Дополнительные листы добавляются как в виде связанного листа с первым листом, а также как несвязанные чистые листы. Между листами возможно делать ссылки.
- Копирование значений и формул из Экселя (с сохранением формул между листами).
- Наличие вертикального блока констант для добавления справочной информации.
- Сохранение расчетных таблиц в шаблоны для дальнейшего их выбора и применения в других объектах приложения.
- Пересчет зависимых данных расчетных таблиц из объектов приложений (справочников «БФТ.Платформы»).
- Пересчет данных из расчетных таблиц.
- Пересчет зависимых данных расчетных таблиц из справочников системы.

- Пересчет данных из расчетных таблиц.

3.19. Программный компонент «Конструктор печатных форм»

Программный компонент «Конструктор печатных форм» предназначен для расчета показателей (разных значений полей) в объектах приложения (справочниках) «БФТ.Платформы», а также хранения рассчитанных показателей и их применения в дальнейших расчетах.

Программный компонент обеспечивает следующие возможности:

- Создание, изменение, удаление шаблонов печатных форм.
- Создание копии шаблона печатных форм.
- Добавление в шаблоны печатных форм приложений, сформированных в редакторе Stimulsoft, или в формате PDF.
  - Настройка шаблонов печатных форм при помощи визуального конструктора:
  - Настройка параметров шаблонов печатных форм;
  - Настройка источников данных для шаблона печатных форм;
  - Настройка условий применения шаблонов печатных форм в объектах приложений системы;
  - Настройка редактируемых частей шаблона печатных форм для разных пользователей;
  - Настройка форматирования текста шаблонов печатных форм в визуальном редакторе.
- Предварительный просмотр шаблонов печатных форм.
- Выгрузка шаблонов печатных форм в формате PDF.
- Сохранение печатных форм на АРМ пользователя.

3.20. Программный компонент «Обработка больших данных»

Программный компонент «Обработка больших данных» предназначен для организации сквозной обработки данных: от подключения внешних источников и графической сборки конвейеров загрузки-трансформации-выгрузки до автоматического запуска этих конвейеров в управляемых процессах.

Программный компонент позволяет фиксировать все запуски процессов, ошибки и результаты проверок качества, обеспечивая аудит, воспроизводимость и контроль над каждой стадией потока данных.

ETL (Extract – Transform – Load) – процесс обработки данных, включающий три ключевых этапа:

- Извлечение.
- Преобразование.
- Загрузка.

Основные функции:

- Определение задач ETL – определение формата метаданных, описывающих шаги задач ETL, графический дизайнер для удобного создания задач ETL и сохранения метаданных задачи.
- Оркестровка исполнения задач ETL – запуск задач ETL, передача параметров в задачи, выполнение дополнительных действий, обработка ошибок и повторные попытки выполнения, просмотр журналов.
- Исполнение задач ETL – реальная работа с данными – разбор метаданных задачи ETL и выполнение их на Среде исполнения.
- Определение задач ETL и их оркестровка реализованы в платформе, Среда исполнения – Spark и Hadoop.

3.20.1. Ведение источников данных

Источник данных – это конфигурируемый компонент, предназначенный для задания параметров подключения к внешним источникам, из которых осуществляется загрузка данных в рамках ETL-процессов. Источник данных описывает:

- Откуда система получает данные (Kafka, файлы, база данных, таблица каталога среды исполнения).
- В каком формате представлены данные. Поддерживаемые форматы файлов: JSON, CSV, Excel файл, Parquet, Avro, текстовый, бинарный.
- По каким правилам происходит чтение данных:
  - Партиционирование – разбиение данных на части для параллельной загрузки и обработки,
  - Декодирование – преобразование данных из исходного формата в структурированный вид,
  - Фильтрация – отсеивание ненужных данных на этапе чтения.

Раздел «Источники данных» модуля «ETL» обеспечивает следующие функциональные возможности:

- Поддержка различных форматов источника данных на чтение/запись: Файлы (CSV, XLSX/XLS, XML, JSON, Parquet, Avro, текстовый, бинарный), работа с реляционной базой данных (JDBC), интеграция Kafka), таблица каталога среды исполнения (Spark).
- Генерация объекта схемы (объекта приложения) источника данных.
- Предварительный просмотр данных.
- Копирование источника данных.
- Просмотр истории изменений источника данных.
- Настройка правила чтения данных (партиционирование, фильтрация, декодирование).
- Игнорировать отсутствующие или поврежденные файлы.
- Выбор режима доступа к источнику данных.
- Удаление набора данных.
- Шифрование данных (at rest, in transit) – настройка с привлечением разработчиков.

3.20.2. Управление конвейерами данных

Конвейер данных – это автоматизированный процесс, который выполняет сквозную обработку информации: от приема сырых данных до передачи готового результата в объекты приложения. Представляется как направленный граф шагов задачи ETL :

- Extract – извлечение данных из источников в среду исполнения.
- Transform – преобразование данных.
- Load – загружает данные в целевую систему (внешние источники или целевое хранилище).

Выполнение конвейера осуществляется в транзакциях, что обеспечивает целостность и согласованность обрабатываемой информации.

- Просмотр истории изменений конвейера данных.
- Отладка конвейера данных.
- Перечень операций в конвейере данных:
  - Источник – извлечение данных из внешних систем (базы, файлы) без их преобразования.
  - Выгрузка – извлечение данных с возможностью первичной фильтрации.
  - Загрузка – сохранение обработанных данных в целевом хранилище (объект приложения).

- Загрузка МО – сохранение обработанных данных в целевом хранилище (объект приложения с включенным признаком Мастер-объект). Создание заявки после сохранения результата в объект приложения
- Приемник – сохранение потока обработанных данных в целевое хранилище внешнего источника данных.
- Трансформация – изменяет структуру или формат данных, маппинг полей, нормализация/денормализация данных.
- Фильтрация – отбирает данные по заданным условиям фильтра, возможность указывать максимальное количество записей.
- Агрегация – объединяет данные для вычисления метрик: сумма, среднее, количество, минимум, максимум, уникальное количество. Возможность группировки (GROUP BY), оконные функции (OVER).
- Обогащение – дополнение данных новой информацией, построение ссылок между объектами, добавление вычисляемых полей, соединение с внешними данными (JOIN).
- Валидация – проверка данных на соответствие установленным правилам валидации, создание записи в журнале Проверки качества данных при несоответствии установленным правилам валидации, валидация форматов (даты, числа, строки).
- Распределение – разделение данных на части для параллельной обработки.
- Соединение – объединение данных из разных источников по ключевым полям
- Консолидация – объединение разрозненных данных в единую согласованную структуру.
- Конвейер данных – выбор конвейера данных.
- Скрипт.
- SQL.

- Подключение к разнородным источникам (файлы, базы данных, API, потоковые платформы, облачные хранилища).
- Поддержка инкрементальной загрузки (только новых/измененных данных).
- Поддержка полной/инкрементальной перезаписи.
- Xml-представление конвейера данных.
- Потоковая обработка данных (обработка событий по мере поступления, микропакетная обработка данных (Spark Streaming, Flink), управление задержками в потоках (watermarks)).
- Распределенное исполнение операций конвейера (Spark, Dask, Flink).
- Кэширование промежуточных результатов в среде исполнения.

- Поддержка машинного обучения (настройка с привлечением разработчиков):
- Повторяемость результатов при перезапуске.

3.20.3. Управление процессами

Процессы – это раздел, отвечающий за автоматизацию, оркестрацию и мониторинг выполнения задач обработки данных. Процесс объединяет все этапы работы с данными: от загрузки до доставки результата.

Раздел обеспечивает построение логики обработки в виде бизнес-процесса, который может включать условия, ветвления, работу с пользовательскими задачами и обработку ошибок. Каждый этап процесса фиксируется, контролируется его статус и результаты выполнения. Использование процессов позволяет централизованно управлять задачами обработки данных, формировать сложные последовательности шагов и обеспечивать согласованность их выполнения.

Раздел «Процессы» обеспечивает следующие возможности:

- Автоматизация запуска процесса (по расписанию или событию).
- Xml-представление процесса.
- Отладка процесса.
- Перечень элементов процесса:
  - Старт.
  - Выход.
  - Ошибка.
  - Условие.
  - Задача пользователя.
  - Установка контекста.
  - Процесс.
  - Конвейер данных.
  - Сценарий.
  - BPM процесс.
  - SQL.
  - Командная строка.
  - Операция с файлами (копировать, переместить, удалить, архивировать, разархивировать).
- Повторные попытки запуска задачи в случае ошибки или сбоя и указание интервала между повторными запусками задачи.

- Просмотр опубликованных версий процесса.
- Маскирование/анонимизация конфиденциальных полей.

3.20.4. Управление исполнением процессов

В части исполнения ETL процессов программный компонент позволяет:

- Просмотр перечня экземпляров процессов.
- Просмотр информации об экземпляре процесса:
  - Перечень выполненных шагов экземпляров процесса.
  - Состояние экземпляра процесса: Запущен, Приостановлен, Завершен, Остановлено внешне.
  - Продолжительности выполнения экземпляра процесса.
  - Перечень значений переменных процесса.
  - Изменение значений переменных процесса для запущенных экземпляров процесса.
  - Перечень ошибок и предупреждений, обнаруженных в записях объекта приложения при выполнении операции Валидация в конвейере данных.
  - Последовательность и результаты выполнения задач процесса.
- Приостановка и остановка исполнения запущенного экземпляра процесса.
- Поиск экземпляра процесса по значению полей Идентификатора процесса и Ид экземпляра процесса.

3.20.5. Просмотр перечня результатов валидации данных источников данных

Раздел «Проверки качества данных» предназначен для анализа и отслеживания результатов валидации данных после выполнения операции «Валидация» в конвейере данных.

Раздел «Проверки качества данных» модуля «ETL» предоставляет следующие возможности:

- Просмотр перечня результатов проверки данных после выполнения операции Валидация в конвейере данных.
- Просмотр информации отдельной проверки качества данных:
  - Дата и время выполнения проверки.
  - Отображаемое имя объекта приложения.
  - Количество проверенных записей.
  - Количество записей с обнаруженными ошибками и предупреждениями.

- Список записей с обнаруженными ошибками и предупреждениями.
- Сообщение ошибки.
- Код записи с ошибкой/предупреждением.
- Отображаемое имя с ошибкой/предупреждением.

3.21. Программный компонент «Мастер-данные»

Программный компонент «Мастер-данные» содержит инструменты, позволяющие управлять изменениями и версиями записей объектов приложения, обеспечивать историчность данных и единственность актуальной (эталонной) версии.

Инструменты программного компонента «Мастер-данные» предоставляют возможности:

- Вносить изменения через регламентированные заявки с прохождением по статусной модели.
- Обеспечивать историзацию состояний записей и хранение актуальной версии как эталон.
- Вести цепочку версий (мажорную и минорную) и просматривать конфигурацию объекта приложения, а также его записей на выбранной версии.
- Управлять составом изменений: добавление, изменение, удаление записей, исключение записей из заявки, перенос записей в отдельную заявку.
- Поддерживать заявки на один или несколько объектов приложений.

3.22. Программный компонент «Дедупликация»

Программный компонент «Дедупликация» содержит инструменты, позволяющие выполнять поиск записей-дублей в одном или двух справочниках, их анализ и обработку дублирующихся данных в записях по заранее заданным правилам и параметрам. В том числе:

- Выбор методов сравнения записей.
- Указание сравниваемых полей записей.
- Установку параметров механизма дедупликации, например, пороги малой и высокой вероятности того, что записи являются дубликатами, и другие параметры.
- Получение результатов дедупликации с возможностью просмотра, удаления дублирующихся записей.

3.23. Программный компонент «Качество данных»

Программный компонент «Качество данных» содержит инструменты для выполнения проверок качества данных в записях объектов приложений. Проверки могут включать правила валидации, условия, форматы ввода значений, скрипты и наличие ссылочной целостности, указанные в объекте приложения.

Программный компонент обеспечивает анализ результатов выполнения проверок и обработку инцидентов для записей, не соответствующих установленным правилам в проверках.

Инструменты программного компонента «Качество данных» предоставляют возможности:

- Создание правил проверки записей объекта приложения.
- Запуск проверок с использованием ранее созданных правил.
- Исправление значений полей записи по настроенным правилам проверки.

3.24. Программный компонент «Редактор документов»

Программный компонент «Редактор документов» позволяет пользователям платформы напрямую работать с файлами-вложениями интерфейса платформы, без необходимости скачивания, работы с файлами-вложениями в отдельных внешних приложениях и повторной загрузки их в платформу.

Редактор предоставляет полноценную среду для редактирования и совместной работы над документами в рамках единого интерфейса платформы и обеспечивает:

- Открытие файлов вложений в отдельной вкладке браузера с редактором.
- Внесение изменений и последующее сохранение.
- Совместное редактирование вложения несколькими пользователями.
- Выбор способа редактирования вложения:
  - Редактирование – позволяет вносить любые изменения.
  - Рецензирование – позволяет предлагать внесенные изменения, чтобы второй пользователь их подтвердил или отклонил.
  - Просмотр – позволяет только просматривать файл и не вносить изменений.
- Замена вложения на новую версию только после сохранения изменений и выхода из редактора всех участников редактирования.
- Восстановление вложения из хранилища редактора в течение 24 часов.
- Отображение ошибки о устарелой версии и необходимости перезагрузке при входе стороннего пользователя в редактор в процессе сохранения изменений.
- Работу с форматами: doc, docx, xls, xlsx, ppt, pptx, pdf (просмотр и аннотирование).

3.25. Программный компонент «Хранилище вложений»

Программный компонент «Хранилище вложений» обеспечивает создание и настройку хранилища файлов вложений сущностей разных типов. Возможности программного компонента:

- Создание и редактирование хранилища вложений с типом:
  - Архив документов – для хранения вложений сущности во внешнем хранилище приложения «БФТ.ХЭД».
  - БД приложения – системное хранилище, не доступное пользователю для корректировки его свойств, для сохранения, удаления, изменения сущностей, хранящихся в данном хранилище вложений;
  - База данных (Large Object) – для хранения объектов больших размеров из таблицы базы данных, созданной пользователем.
  - База данных (bytea) – для хранения небольших объектов (до 8 КБ) в конкретном столбце таблицы базы данных, созданной пользователем.
  - Облачное хранилище S3 – для хранения вложений на удаленном облачном сервере.
- Использование хранилища вложений по умолчанию.
- Удаление вложений из архива документов.
- Включение и отключение хранилища вложений.
- Создание исключений для хранения вложений.
- Способы хранения вложений:
  - Хранение всех сущностей в одном хранилище.
  - Хранение вложений сущностей в одном хранилище с указанием исключений для полей, вложения которых должны храниться в другом хранилище.
- Перенос вложений из одного хранилища в другое.

3.26. Программный компонент «Модуль интеграции взаимодействий с другими приложениями (Модуль интеграции «БФТ.ПИВ»)»

Программный компонент интеграции взаимодействий с другими приложениями («Модуль интеграции БФТ.ПИВ») предназначен для автоматизации процессов формирования, маршрутизации и гарантированной доставки сообщений в адрес систем-получателей.

Программный компонент интеграции взаимодействий с другими приложениями («Модуль интеграции БФТ.ПИВ») обеспечивает выполнение следующих функций:

- Управления перечнем подключенных информационных систем;

- Выполнения настройки маршрутов взаимодействия с информационными системами, управления маршрутами и мониторинг сообщений обмена;
- Просмотр запросов, направляемых Rest-приложениями;
- Мониторинг сообщений очередей подключенных брокеров;
- Сохранение сообщений и запросов в базу данных;
- Сохранение информации в Журнале информационного обмена.

3.27. Программный компонент «Исполнение длительных задач»

Программный компонент «Исполнение длительных задач» предназначен для управления исполнением такими задачами, как:

- создание, запуск, формирование статуса, исполнение задач;
- управление задачами из внешних приложений;
- управление задачами через REST;
- передачи сообщений в Kafka с целью получения заданий для исполнения рабочими узлами из очереди;
- передачи сообщений в AMQ с целью получения заданий для исполнения рабочими узлами из очереди;
- для отображения интерфейсных форм просмотра, редактирования и управления задачами.

В программном компоненте «Исполнение длительных задач» для реализации основных функций используется Spring Batch – платформа, предназначенная для пакетной обработки больших объёмов данных и надежного выполнения заданий. Она позволяет реализовать функции запуска распределённых задач, в том числе удаленно. Пакетный процесс инкапсулируется заданием, которое может состоять из нескольких этапов – шагов, проходящих последовательность чтения, обработки и записи данных.

В блоке управления происходит деление задач на шаги и запуск процесса их исполнения путем раздачи заданий из блока управления в брокер сообщений. Брокеры могут быть различные, программа поддерживает взаимодействие с Kafka и ActiveMQ. Блок исполнения получая задание из брокера приступает к его выполнению. При этом Spring Batch позволяет подключать функции по запуску и исполнению задач как раздельно, так и совместно.

- Управление задачами:
  - Создание.
  - Запуск.

  - Получение статусов задач.
- Исполнение задач.
- Управление задачами из внешних приложений, используя клиент на языке Kotlin.
- Управление задачами из внешних приложений, используя клиент на основе Feign.
- Выполнение задач на выделенном микросервисе (микросервисах).
- Управление задачами через REST.
- Передача сообщений в Kafka с целью получения заданий для исполнения рабочими узлами из очереди.
- Передача сообщений в AMQ с целью получения заданий для исполнения рабочими узлами из очереди.
- Отображение интерфейсных форм просмотра, редактирования и управления задачами.

3.28. Программный компонент «Реестр приложений»

Программный компонент «Реестр приложений» обеспечивает возможность создавать приложения, которые используются при необходимости перекрёстной работы между данными, хранящимися в разных источниках данных, и отображать связи между приложениями, модулями и источниками данных. Возможности программного компонента:

- Создание и подключение приложения.
- Работа с данными из другой базы данных.
- Работа с диаграммой приложений.
- Мониторинг доступности включенных приложений.
- Автоматическая или ручная проверка соединения с приложением при сохранении внесенных изменений в настройки приложения.

3.29. Программный компонент «Создание и настройка ИИ-агентов»

Программный компонент «Создание и настройка ИИ-агентов» предназначен для создания, настройки и управления универсальными интеллектуальными агентами, реализующими обработку данных и выполнение прикладных задач на основе технологий обработки естественного языка реализует универсальную платформу построения агентов, где:

- агент является базовой единицей обработки;
- ансамбль агентов – способом композиции сложной логики;
- конфигурация определяет поведение без изменения кода.

Таким образом, любые прикладные сценарии (ИИ-ассистент, анализ документов и др.) реализуются как частные случаи настройки агентов и ансамблей.

3.29.1. Особенности реализации

Программный компонент не содержит встроенных больших языковых моделей, а предоставляет механизм подключения внешних моделей (включая open-source решения) через систему конфигурационных профилей по REST API, совместимому с OpenAI.

Рекомендованная большая языковая модель для функционала парсинга и анализа файлов – gpt-oss-120b.

Рекомендованная большая языковая модель для функционала ИИ-ассистента:

- Qwen3.6-35B-A3B
- Qwen3-Next-80B-A3B

3.29.2. Область применения

Программный компонент может применяться в следующих областях:

- построение интеллектуальных сервисов;
- автоматизация обработки текстовой информации;
- анализ и извлечение данных из документов;
- реализация пользовательских сценариев взаимодействия.

3.29.3. Архитектура

Основные компоненты архитектуры:

- Агент – реализует конкретную функцию обработки данных и представляет собой конфигурируемую структуру, включающую:
  - большую языковую модель (подключаемую внешне);
  - системный промпт (инструкцию выполнения);
  - набор зависимостей (модели, хранилища, инструменты).
- Ансамбль агентов – набор агентов, объединённых для выполнения сложного сценария обработки. Ансамбль является механизмом реализации прикладной логики и имеет функции:
  - распределение ролей между агентами;
  - последовательное или параллельное выполнение;
  - агрегирование результатов.

- Профили – обеспечивают независимость системы от конкретных технологий и обеспечивают подключение внешних компонентов:
  - профили больших языковых моделей;
  - профили моделей эмбеддингов;
  - профили моделей ранжирования;
  - профили хранилищ (векторных, графовых, файловых).

3.29.4. Технические особенности реализации

Технические особенности

- модульная архитектура;
- конфигурационный подход;
- масштабируемость через ансамбли;
- независимость от поставщиков моделей;
- поддержка open-source больших языковых моделей;
- интеграция с различными типами хранилищ.

3.29.5. Функциональные возможности

Перечень функциональных возможностей программного компонента:

- Управление агентами, включает:
  - создание агента;
  - редактирование конфигурации;
  - удаление агента;
  - просмотр и поиск;
  - ведение истории изменений;
  - экспорт конфигураций.
- Настройка поведения агента, включающая:
  - Общие параметры, такие как:
    - имя конфигурации;
    - контекст применения.
  - Шаблоны агентов. Шаблон определяет тип поведения агента. Поддерживаются шаблоны:
    - извлечение данных;
    - сопоставление значений;

- кластеризация;
- выбор ключевых параметров;
- диалоговое взаимодействие.

- Интеграция внешних компонентов:
  - подключение большой языковой модели;
  - подключение моделей эмбеддингов;
  - подключение моделей ранжирования;
  - настройка хранилищ данных.
- Параметры выполнения:
  - управление параллелизмом;
  - настройка глубины поиска;
  - управление качеством результатов;
  - повторные попытки обработки.
- Управление промптами:
  - просмотр системного промпта;
  - редактирование;
  - сохранение изменений.

- Управление ансамблями:
  - создание ансамбля;
  - назначение ролей агентам;
  - конфигурацию взаимодействия.

3.29.6. Конфигурации, реализованные на базе компонента

Программный компонент позволяет реализовывать различные конфигурации. Примеры реализации:

- Агент для консультации по базе знаний продукта, далее ИИ-ассистент.
- Агент для реализации парсинга и анализа файлов, далее ПИАФ.

3.29.6.1. Конфигурация ИИ-ассистент

ИИ-ассистент предназначен для организации диалогового взаимодействия с пользователем и выполняет функции:

- обработка пользовательских запросов;
- генерация ответов с использованием большой языковой модели;

- глубокий поиск информации в базе знаний с поддержкой семантического индексирования и извлечения фрагментов из больших документов;
- выдача ответов с указанием цитаты или ссылки на документ;
- поддержка диалоговых сессий.

ИИ-ассистент реализуется как:

- один ансамбль, включающий один или несколько агентов, что позволяет гибко маршрутизировать запросы к нужному обработчику документов;
- с шаблоном диалогового взаимодействия (возможность настроить глубину и стиль ответов под специфику документов);
- с подключением базы знаний (возможность загрузки сложных больших документов без потери структуры).

3.29.6.2. Конфигурация ПИАФ

Агент для реализации парсинга и анализа файлов (ПИАФ) предназначен извлечения и анализа структурированных данных из документов и выполняет функции:

- Извлечение данных:
  - анализ текстов документов, в том числе больших неструктурированных документов;
  - выделение структурированных значений.
- Семантический поиск:
  - использование векторных представлений;
  - поиск релевантных фрагментов.
- Сопоставление и нормализация:
  - сопоставление со справочниками;
  - унификация данных.
- Анализ:
  - сравнение данных между документами;
  - выявление расхождений.
- Шаблонный парсинг документов:
  - Дополнительно поддерживается механизм извлечения данных на основе заранее заданных шаблонов документов.

ПИАФ реализован как ансамбль агентов, каждый из которых выполняет специализированную функцию.

Типовые роли агентов ПИАФ:

- извлечение данных;
- сопоставление значений;
- кластеризация;
- извлечение ключей;
- выбор ключевых параметров.

Функциональные возможности ПИАФ:

- использование предопределённых структур документов;
- извлечение данных по фиксированным позициям, меткам и правилам;
- применение специализированных сценариев обработки для типовых документов.

Поддерживаемые типы документов для шаблонного распознавания:

- платежное поручение;
- счет на оплату;
- счет-фактура;
- накладная.

Особенности подхода:

- обеспечивает повышенную точность извлечения для стандартизированных форм;
- снижает зависимость от интерпретации большой языковой модели.

Процесс выполнения состоит из этапов:

- загрузка документов;
- выбор конфигурации ансамбля;
- настройка параметров;
- запуск обработки;
- получение результатов.

Результаты выполнения:

- структурированные данные;
- данные для заполнения объектов системы;
- аналитическая информация;
- сравнение данных, полученных из документов.

3.30. Программный компонент «Конфигуратор API»

Программный компонент «Конфигуратор API» позволяет массово выгружать записи из объектов приложений (пользовательских или системных). Программный компонент оптимизирован для работы с большими массивами данных (до нескольких миллионов строк) за счет использования потокового режима чтения данных. Возможности программного компонента:

- Получение JSON схемы объекта приложения.
- Получение записей выбранного объекта приложения:
  - Фильтрация записей объекта приложения.
  - Сортировка записей объекта приложения.
  - Настройка возвращаемых полей в ответе.
  - Получение записей вложенного списка с включенным признаком Основной.

3.31. Программный компонент «Сервер аутентификации по OAuth 2.0»

Программный компонент «Сервер аутентификации по OAuth 2.0» позволяет приложению «БФТ.Платформы» выступать в качестве сервера авторизации по протоколу OAuth 2.0. При подключении программного компонента приложение предоставляет сторонним сервисам интерфейс для авторизации по данному протоколу.

Программный компонент предназначен для использования в интеграционных сценариях, в которых сторонний сервис не имеет собственной аутентификации и использует «БФТ.Платформу» в качестве сервера авторизации. Пользователь проходит аутентификацию на стороне «БФТ.Платформы», после чего сторонний сервис получает токен доступа для работы с защищёнными ресурсами.

Возможности программного компонента:

- Предоставление интерфейса для входа внешним системам.
- Использование «БФТ.Платформы» в качестве сервера авторизации по OAuth 2.0.
- Выдача токенов доступа.
- Поддержка KeyStore для подписи токенов.
- Регистрация клиентов по OAuth 2.0.
- Автоматическая очистка устаревших токенов.

3.32. Возможности интерфейса пользователя

Программный комплекс «Базовый состав платформы» обеспечивает следующие возможности интерфейса пользователя:

- Доступ к функциям приложений с помощью web-клиента.
- Настройка для пользователя перечня доступных объектов и функций:
  - Отображение в меню только перечня доступных пользователю объектов и функций, конфигурирование структуры меню;
  - Индивидуальная настройка для пользователя формы отображения перечней записей в справочниках и списках системы: настройка перечня отображаемых колонок, сортировка значений в колонках;
  - Индивидуальная выборка для перечней записей в справочниках и списках системы: фильтрация записей списков;
  - Возможность сохранения и повторного применения к спискам индивидуальных пользовательских настроек отображения колонок списка и параметров фильтрации по полям объекта приложения или колонкам формы списка: профили фильтров;
  - Возможность автоматического сохранения фильтров, созданных на форме списка пользовательских справочников;
  - Возможность скрывать/отображать сведения «О системе» для пользователей без роли администратора;
  - Возможность изменять наименование организации, скрывать сведения о базе данных для всех пользователей, не обладающих правами администратора, в разделе «О системе»;
- Возможность настраивать дополнительные ключевые данные и отображать их в разделе «О системе».

4. Описание функциональности программного комплекса «Платформа высоконагруженных микросервисных приложений»

Программный комплекс «Платформа высоконагруженных микросервисных приложений» включает следующие программные компоненты:

- Базовые библиотеки прикладной разработки.
- Безопасность прикладного программного обеспечения.
- Криптографические операции.
- Потоковая обработка данных.
- Система управления распределенными базами данных.
- Шлюз для фронтэнда.
- Сбор информации о сервисах.
- Печатные формы.
- Аудит доступа к данным и истории изменений.
- Обеспечение миграции данных.
- Фронтенд для веб-клиента.

4.1. Программный компонент «Базовые библиотеки прикладной разработки»

Программный компонент «Базовые библиотеки прикладной разработки» представляет собой набор библиотечных модулей, применение которых обеспечивает ускорение разработки микросервисов в части минимизации трудозатрат на написание кода программного обеспечения и быстрого выпуска релизов продукта. Библиотечные модули обеспечивают работу базовых механизмов, которые используются в сервисах платформы, включая (но не ограничиваясь):

- авторизацию (управление ролями и правами доступа);
- работу с агрегатами (сущностями) платформы и их согласованное хранение в Postgres и Phoenix HBase;
- работу с такими хранилищами, как Picodata SQL и Picodata Radix;
- работу с сообщениями и топиками Kafka;
- модули для работы с задачами бизнес-процессов: чтение статуса, обновление данных, выполнение команд и т.д. (Camunda);

- обработку событий различного рода в платформе в унифицированном виде;
- журналирование системных событий;
- мониторинг поведения частей платформы;
- механизмы поддержки изменения параметров конфигурации системы без перезапуска ее модулей;
- унифицированного клиента для работы с продуктом «БФТ.ЕНСИ»;
- технологические модули и классы для типового построения сервисов программного комплекса «Платформа высоконагруженных микросервисных приложений».

Библиотека позволяет абстрагировать процесс разработки прикладного программного обеспечения от сложностей низкоуровневой настройки gRPC, управления транзакциями, балансировки нагрузки, метаданных сервисов и интеграции с внешними сервисами (реестры, кэши, очереди). Обеспечивает стандартизацию, безопасность и надежность коммуникации между микросервисами в рамках системы в целом.

Основные функциональные блоки:

- Блок поддержки работы с базами данных PostgreSQL и Phoenix HBase (высокоуровневая ORM) обеспечивает прозрачную работу в следующих режимах:
  - в режиме только PostgreSQL;
  - в режиме только Phoenix HBase;
  - в гибридном режиме PostgreSQL + Phoenix HBase (в том случае часть запросов выполняется в выделенной БД PostgreSQL, а часть в общем кластере Phoenix HBase).
- Блок поддержки построения gRPC-сервисов (серверная и клиентская часть), обеспечивает:
  - генерацию proto-файлов для описания сервиса;
  - java-реализацию клиента;
  - java-реализацию сервера;
  - конвертацию DTO (Java) <-> Proto-сообщений;
  - автоматическое создание пула каналов gRPC с настраиваемым размером;
  - автоматическую передачу метаданных в заголовках (контексты транзакций, трассировки и т.д.);
  - настройку лимитов сообщений, дедлайнов (deadline);
  - встроенный механизм ограничения скорости запросов (Rate Limiting) на уровне сервера.
- Интеграция с сервисом noble-registry-service для динамического поиска адресов сервисов.

- Интеграция с системой MDM (БФТ.ЕНСИ) через gRPC. Блок предназначен для работы со справочниками (классификаторами) и поддерживает:
  - общие аннотации для разметки DTO (указание имени справочника и имен полей в MDM);
  - настраиваемое кеширование;
  - автоматическое преобразование типов между MDM и gRPC вариантами;
  - историчность данных (запросы на определенную дату);
  - возможность выполнения запросов к нескольким разным справочникам за один раз.
- Блок балансировки и DNS обеспечивает:
  - кастомный NameResolver для кэширования и обновления DNS записей;
  - поддержку политик балансировки (round_robin, pick_first).
- Блок общего клиента для работы с транзакциями обеспечивает:
  - открытие/закрытие транзакций (RC/RR уровни изоляции);
  - блокировки строк и пакетные операции;
  - ожидание завершения транзакций;
  - получение снимков (snapshots);
  - кэширование статусов транзакций для снижения нагрузки на сервис transact-mananger.
- Механизмы динамического обновления конфигурации без перезапуска сервисов.
- Клиент для поддержки работы с Picodata Radix с поддержкой работы кластерного режима и шардированных конфигураций.
- Блок автоматической генерации кода для валидации DTO на основе аннотаций.
- Блок общего клиента для работы с запросами авторизации (работа с authz-server).
- Блок интеграции Camunda BPM с другими сервисами, системами очередей (Kafka) и бизнес-логикой с целью создания единого и легко поддерживаемый набора инструментов для разработки сервисов на базе Camunda, который:
  - обеспечивает упрощенное взаимодействие с Camunda REST API;
  - обеспечивает взаимодействие с External Tasks: управление пулом потоков, поддержка динамического регулирования количества задач в очереди на основе времени их обработки, управление стратегиями группировки задач, наличие защиты от повторного выполнения задач в случае сбоев, возможность выноса хранения статуса во внешний сервис;
  - обеспечивает асинхронное взаимодействие (Command/Event);

  - предоставляет кастомные REST API для управления процессами и задачами вне стандартного API Camunda;
  - интегрирует сквозную трассировку в контекст процессов Camunda.
- Механизм единого централизованного управления и передачи важной информации (контекста) между микросервисами и внутри них через gRPC вызовы, а также управление транзакциями:
  - контекст запроса/пользователя;
  - контекст задачи пользователя;
  - транзакционный контекст;
  - бизнес-контекст;
  - трассировка.
- Блок валидаций и управления доступом на основе окружения (среды разработки и тестирования, а также продуктивные среды) для защиты от ошибок настройки тестирования и отладки.
- Блок автоматизированной обработки Dead Letter Queue (DLQ) в Kafka-архитектуре – перехватчик сообщений, с возможностью сохранения сообщений в базу данных для последующего анализа и возможностью их ручной обработки (исправления и повторной отправки):
  - приложение слушает указанные DLQ-топики в Kafka;
  - каждое сообщение, попавшее в DLQ, преобразуется в объект DeadLetter и сохраняется в базу данных PostgreSQL;
  - сохраняется не только тело сообщения, но и метаданные (заголовки, ключ, офсеты, ошибки, JSON-представление), что позволяет точно воспроизвести ситуацию;
  - предоставлен REST API для просмотра и управления статусом записей.

4.2. Программный компонент «Безопасность прикладного программного обеспечения»

Программный компонент «Безопасность прикладного программного обеспечения» представляет собой набор инструментов, обеспечивающих возможность аутентификации и авторизации в системе с предоставлением унифицированного gRPC-API другим модулям программного комплекса «Платформа высоконагруженных микросервисных приложений».

Программный компонент обеспечивает безопасный вход и контроль доступа без дублирования логики в каждом сервисе, обеспечивая централизованное управление пользователями и правами их доступа совместно с продуктами на базе протокола SCIM (получение

информации о пользователях) и JWT-Token (аутентификация), предоставляя унифицированное gRPC-API для остальных сервисов разрабатываемого приложения.

Для своей работы требует работы продуктов с поддержкой SCIM-протокола (например, Avanpost IDM).

Состав модуля:

- сервис authz-server;
- сервис auth-cache-manager;
- сервис idm-rest-service;
- go-сервис oauth2proxy;
- база данных Picodata SQL.

Программный компонент обеспечивает реализацию следующих ключевых возможностей:

- выполнение авторизации ABAC/RBAC для выполнения бизнес-функций других модулей платформы;
- выполнение аутентификации совместно как для внутренних пользователей системы, так и для внешних (ЕСИА);
- получение информации об имеющихся пользователях системы от ЕСИА (внешние пользователи) и SCIM-систем (в случае внутренних пользователей платформы). В БД хранятся получаемые от вышеупомянутых систем сведения о пользователях и их группах.

4.2.1. Сервис authz-server

Сервис authz-server представляет собой централизованный gRPC-сервис, обеспечивающий унифицированное управление пользователями, ролями, группами и проверку прав доступа к ресурсам на основе моделей RBAC, ABAC и их гибридных комбинаций. Сервис интегрируется с другими компонентами системы, которым требуется проведение авторизации выполняемых операций, и обеспечивает высокую производительность за счёт кэширования, потоковой обработки и асинхронной синхронизации данных.

Возможности сервиса:

- Управление информацией о пользователях:
  - получение полной информации о пользователе по идентификатору пользователя;
  - получение информации о пользователе с перечнем присвоенных ролей;
  - кеширование запросов к информации о пользователях (без запроса к БД);

  - поддержка пакетной выгрузки для формирования витрин данных;
  - асинхронное обновление данных пользователя в кэше (синхронизация с источником);
  - выборка пользователей по имени группы (роли);
  - выборка пользователей по территориальному подразделению;
  - выборка пользователей по организационному подразделению;
  - поддержка составных фильтров по ФИО, email, телефону, ИНН, СНИЛС, ОКПО, номеру сотрудника, статусу, типу участника и другим атрибутам.
- Управление ролями и группами:
  - получение списка ролей, назначенных пользователю;
  - получение роли по ID или имени;
  - проверка существования группы (роли) по имени.
- Проверка прав доступа RBAC (используется Casbin), Role-Based Access Control – проверка доступа к ресурсу по следующим параметрам:
  - идентификатор пользователя;
  - запрашиваемый объект;
  - область;
  - действие;
  - дополнительные произвольные политики.
- Проверка прав доступа ABAC (используется Casbin), Attribute-Based Access Control:
  - проверка доступа к произвольному объекту на основе его атрибутов и атрибутов пользователя;
  - массовая проверка доступа к множеству объектов в одном вызове;
  - операции массовой проверки:
    - проверка множества RBAC-запросов в одном gRPC-вызове (оптимизация для высоконагруженных сценариев);
    - массовая ABAC-проверка для сотен объектов;
    - проверка доступа к множеству ресурсов – указанному множеству объектов.
- Хранение данных в Picodata SQL.
- Предоставление метрик Prometheus (работа с БД, работа с API).
- Аудит доступа (формирование сообщений в указанный топик Kafka с отказами в доступе).

4.2.2. Сервис auth-cache-manager

Сервис auth-cache-manager представляет собой сервис, предназначенный для валидации и обработки JWT-токенов от системы ЕСИА, преобразования их в структурированные данные и сохранения в базу данных (Picodata SQL) для последующего использования в системе. Также сервис отвечает за редиректы пользователей после успешной аутентификации, валидируя целевой URL перед перенаправлением.

Сервис является интеграционным звеном между внешней системой аутентификации (ЕСИА) и сервисом авторизации authz-server, обеспечивая следующие возможности:

- Проведение декодирования JWT-токен от внешней системы (например, ЕСИА) для извлечения и формирования следующих структурированных данных:
  - пользователь (физическое лицо);
  - организация пользователя;
  - группы безопасности пользователя.
- Сохранение полученных из JWT-токена сведений в базу данных Picodata SQL (для последующего использования сервисом авторизации authz-server).
- Валидация URL, на который пользователь должен быть перенаправлен после аутентификации, с целью проверки легитимности выполняемого перенаправления.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.2.3. Сервис idm-rest-service

Сервис idm-rest-service представляет собой RESTful API-сервис – интеграционный адаптер для управления идентификацией и доступом (IAM) в соответствии со стандартом SCIM, позволяющий получать сведения о пользователях организации и их ролях, преобразуя их в структурированные данные для сохранения в базу данных (Picodata SQL) для последующего использования в системе.

Сервис является интеграционным звеном между внутренней системой аутентификации (по протоколу SCIM, который используется в рамках интеграции с продуктами Avanpost, в частности с Avanpost IDM) и сервисом авторизации authz-server, обеспечивая следующие возможности:

- Обновление сведений о пользователях с использованием SCIM API:
  - пользователи: создание, чтение, обновление, удаление пользователей (CRUD);
  - группы: управление группами безопасности (ролями);
  - патч-операции: частичное обновление пользовательских данных;
  - фильтрация: поддержка SCIM-фильтров (например, userName eq «username»).
- Интеграция и возможности расширения:

  - поддержка дополнительных атрибутов: организация, подразделение, должность, коды;
  - динамическая генерация схемы для использования на стороне источника через протокол SCIM.
- Поддержка интеграции с MDM-системой (БФТ ЕНСИ) для работы со справочниками:
  - получение данных об организациях, территориальных подразделениях, должностях;
  - кеширование справочников для уменьшения нагрузки на MDM-систему.
- Сохранение полученных из SCIM-протокола сведений в базу данных Picodata SQL (для последующего использования сервисом авторизации authz-server).
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.2.4. Сервис oauth2-proxy

Сервис oauth2-proxy представляет собой готовый сервис аутентификации и авторизации пользователей через OAuth2/OIDC для обеспечения процедуры аутентификации конечных пользователей.

Сервис является аутентификационным звеном между используемым провайдером аутентификации и остальными модулями системы, позволяя пользователю выполнить процедуру входа один раз, после которой он получает ко всем защищённым сервисам (поддержки SSO).

Обеспечивает следующие основные функциональные возможности:

- Проверка аутентификационной cookie.
- Перенаправление пользователя на провайдера аутентификации OAuth2/OIDC для прохождения процедуры аутентификации.
- Формирование и шифрование аутентификационной cookie.
- Добавление заголовков с информацией об аутентифицированном пользователе при передаче запроса внутрь системы.
- Интеграция как sidecar или Ingress-контроллер (например, с Nginx Ingress) с Kubernetes.
- Предоставление метрик Prometheus (работа с API).

4.3. Программный компонент «Криптографические операции»

Программный компонент «Криптографические операции» обеспечивает ГОСТ-совместимую защиту данных и документооборота, отказоустойчивость за счёт балансировки работы криптографических средств, управляя подключаемыми средствами ViPNet PKI Service (с возможностью быстрой интеграции альтернативных поставщиков криптографических решений). Это позволяет внедрить подписание/проверку ЭЦП (CMS/WSS/XML) и возможность работы с шифрованием документов.

Программный компонент реализует возможность использования подключаемых сертифицированных криптографических средств для выполнения различных операций (на текущий момент поддерживается ViPNet PKI Service от компании Инфотекс) с предоставлением унифицированного gRPC-API другим компонентам программного комплекса «Платформа высоконагруженных микросервисных приложений».

В состав программного компонента модуля входит сервис cert-manager.

Программный компонент обеспечивает реализацию следующих ключевых возможностей:

- Подписание и проверка ЭЦП документов по следующим технологиям:
  - CMS;
  - WSS;
  - XML.
- Шифрование и расшифровывание документов, получаемых в системе или передаваемых из системы в другие системы.
- Обеспечение балансировки запросов, подключаемых к cert-manager, у необходимого количества экземпляров ViPNet PKI Service. В БД хранятся настройки параметров работы с кластером ViPNet PKI Service.

4.3.1. Сервис cert-manager

Cert-manager представляет собой сервис, предназначенный для предоставления gRPC-API, обеспечивающий возможность использования функций по работе с ЭЦП и шифрованием с помощью ViPNet PKI Service, которое используется остальными компонентами программного комплекса «Платформа высоконагруженных микросервисных приложений».

Сервис cert-manager обеспечивает следующие функциональные возможности:

- Предоставление gRPC-API для выполнения криптографических операций.
- Подпись (Signing): XML, бинарных файлов, WSS (Web Services Security).

- Верификация (Verification): проверка XML, CMS и WSS подписей.
- Шифрование/Дешифрование (Encryption/Decryption):
  - готовых конечных файлов,
  - произвольных двоичных данных, передаваемых от других модулей системы.
- Интеграция с БФТ.ХЭД (электронный архив документов), а именно загрузка и выгрузка файлов для:
  - подписания электронной подписью для последующей отдачи смежным системам;
  - проверки электронной подписи файлов, получаемых извне системы;
  - расшифровывания файлов, получаемых извне системы;
  - зашифровывания для последующей отдачи смежным системам.
- Реализация балансировки нагрузки между несколькими серверами ViPNet PKI Service с поддержкой повторных попыток выполнения операция при сбоях на разных экземплярах ViPNet PKI Service.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.4. Программный компонент «Потоковая обработка данных»

Программный компонент «Потоковая обработка данных» позволяет быстро строить ReadView/витрины данных и обрабатывать потоковые события с минимальным написанием собственного кода, обеспечивая управление задачами Apache Flink, интеграцию с Postgres, Phoenix HBase, OpenSearch, Kafka, gRPC-API.

Программный компонент обеспечивает возможность запуска и управления жизненным циклом отдельных задач преобразования данных для Apache Flink для построения ReadView/витрин данных.

Основная среда выполнения модуля потоковой обработки данных – Apache Flink на кластере Hadoop, который управляется YARN.

Состав программного компонента:

- flink-connector-hbase;
- flink-connector-jdbc;
- flink-connector-opensearch;
- flink-admin-service;
- flink-event-processor.

Программный компонент обеспечивает реализацию следующих ключевых функциональных возможностей:

- Предоставляет базовые механизмы построения Flink Job для последующей интеграции в экосистему Apache Flink с учетом основных архитектурных решений системы в целом.
- Обеспечивает загрузку, запуск и отслеживание событий жизненного цикла каждой задачи Apache Flink.
- Предоставляет унифицированные способы работы со следующими источниками/целями данных:
  - Postgres;
  - Phoenix HBase;
  - OpenSearch;
  - gRPC-сервисы системы (для унифицированного API работы с данными, разработанного в Платформе).
- Поддерживает механизмы обновления/актуализации данных на основе поступающих событий топиков Окна Целостности платформы.

4.4.1. Коннектор flink-connector-hbase

Коннектор flink-connector-hbase представляет собой функциональный блок для Apache Flink, который обеспечивает интеграцию между потоковой обработкой данных в Flink и базой данных Apache HBase. Проект позволяет использовать HBase как источник данных (source) и как приемник данных (sink) в конвейерах обработки данных Flink.

Основные возможности:

- Запись данных (Sink): позволяет записывать данные из Flink-потоков в HBase. Поддерживает буферизацию (batching) для повышения производительности, используя BufferedMutator от HBase.
- Чтение данных (Source): позволяет читать данные из HBase как для пакетных, так и для потоковых заданий.
- Lookup (Поиск): реализует асинхронный или синхронный поиск записей по ключу (row key) для операций join (например, temporal table join).
- Поддержка типов данных: автоматическая сериализация/десериализация стандартных типов Flink (String, Integer, Long, Boolean, Decimal, Timestamp, Binary и др.) в формат HBase (byte arrays).
- Гибкая конфигурация: поддержка пользовательских метаданных (время жизни TTL, временные метки) и кэширования результатов поиска.

4.4.2. Коннектор flink-connector-jdbc

Коннектор flink-connector-hbase представляет собой функциональный блок для Apache Flink, который обеспечивает интеграцию между потоковой обработкой данных в Flink и базой данных PostgreSQL.

Основные возможности:

- Запись данных (Sink): запись данных в БД с поддержкой буферизации, группировки по пакетам (batching), атомарных транзакций (XA) для обеспечения семантики Exactly-Once.
- Чтение данных (Source): чтение данных из БД с поддержкой параллельного чтения (splitting по диапазонам, временным окнам), различных режимов доставки (At-Least-Once, Exactly-Once).
- Поддержка фильтра pushdown (условия WHERE выносятся в запрос к БД).
- Поддержка Lookups (поиск по ключу).
- Поддержка кэширования результатов запросов.

4.4.3. Коннектор flink-connector-opensearch

Коннектор flink-connector-opensearch представляет собой Flink-коннектор для OpenSearch (Elasticsearch), предназначенный для интеграции приложений Apache Flink с кластерами OpenSearch, позволяя потоково записывать данные в индексы OpenSearch с поддержкой различных гарантий доставки.

Основные возможности и функции:

- Динамическая генерация имен индексов:
  - поддержка статических имен индексов (например, my_index);
  - поддержка динамических имен на основе полей данных (например, users_{user_id});
  - поддержка динамических имен на основе временных меток с форматированием (например, data_{log_ts|yyyy-MM-dd}), что позволяет автоматически создавать индексы по дате.
- Гарантии доставки (Delivery Guarantees):
  - NONE: нет гарантий, данные могут быть потеряны при сбоях;
  - AT_LEAST_ONCE: данные гарантированно доставляются хотя бы один раз (ожидают подтверждения на чекпоинте);
  - поддержка конфигурации sink.delivery-guarantee.
- Оптимизация записи (Bulk Processing):

- буферизация операций перед отправкой в кластер.
- конфигурация размера буфера: sink.bulk-flush.max-actions (макс. количество операций в буфере), sink.bulk-flush.max-size (макс. размер буфера в МБ), sink.bulk-flush.interval (интервал автоматического сброса буфера).
- Backoff Policy (политика повторных попыток):
  - поддержка экспоненциальной (EXPONENTIAL) и постоянной (CONSTANT) задержки при повторных попытках;
  - настройка количества повторных попыток (sink.bulk-flush.backoff.max-retries) и начальной задержки (sink.bulk-flush.backoff.delay).

4.4.4. Сервис flink-admin-service

Сервис flink-admin-service представляет собой сервис, предназначенный для администрирования и управления Flink-заданиями, схемами данных (Read-Views и проекциями) и кластерами Flink. Сервис предоставляет инструменты для автоматизации жизненного цикла потоковой обработки данных, валидации схем, управления версиями и мониторинга состояний.

Основные возможности и функции:

- Регистрация и управление схемами.
- Жизненный цикл Job'ов: запуск, остановка, пауза, перезапуск и удаление Flink-заданий.
- Управление версиями: поддержка семантического версионирования, автоматическая остановка предыдущих версий при запуске новой.
- Использование кластеров Flink: динамическое управление кластерами (добавление/удаление), поддержка режимов YARN и Standalone, автоматическое определение актуального кластера для модулей.
- Сохранение состояния (Savepoints): автоматическое создание и управление savepoint'ами для возможности восстановления заданий.
- Использование ElasticSearch: автоматическое создание индексов, настройка алиасов для «zero-downtime» развертывания новых версий данных.

4.4.5. Сервис flink-event-processor

Сервис flink-event-processor – это Flink-джоб (приложение на Apache Flink), предназначенный для обеспечения согласованности и целостности данных в распределенной системе, работающей с транзакциями.

Основная задача джоба – читать поток сырых событий (raw events) из Kafka, отслеживать жизненный цикл транзакций (начало, изменения, коммит или откат) и на выходе выдавать гарантированно упорядоченные события о сущностях и событиях только после того, как транзакция будет окончательно подтверждена (committed).

Ключевые возможности и концепции:

- Отслеживание транзакций (Transactional Awareness):
  - слушает топик iw-raw, куда поступают события: TX_STARTED, ENTITY (изменения сущностей), EVENT (события), TX_COMMITED, TX_ABORTED;
  - группирует изменения по txId (идентификатору транзакции);
  - ожидает окончательного сигнала (TxCommit или TxAbort) перед тем, как пропустить изменения дальше по конвейеру.
- Глобальная сортировка и Watermarks:
  - использует механизм Sorter и RealTimeAdjuster для обеспечения глобального порядка событий (это критично для систем, где порядок обработки событий влияет на состояние, например, обработка событий пользователя или изменение сущностей;
  - использует «логическое время» (sequence ID транзакции) и корректирует «реальное время» (timestamp) для предотвращения дедлоков и обеспечения монотонности.
- Поддержка сущностей и Broadcast:
  - динамически получает список известных сущностей (Pojo) из Noble Registry Service (через gRPC);
  - использует CommitBroadcaster для рассылки последних известных состояний всем сущностям, чтобы обеспечить корректную обработку «опоздавших» событий или событий для сущностей, которые еще не были обработаны (например, при ребалансе партиций).
- Разделение потоков данных (Outputs): пишет результаты в несколько топиков Kafka в зависимости от типа события и настроек:
  - Entities (iw-entity-*): изменение конкретных сущностей (EntityChange);
  - Events (iw-events-*): интеграционные события (Event);
  - IW Committed (iw-commited): запись всех изменений транзакции после коммита.
  - IW Aborted (iw-aborted): события, связанные с откатом транзакций (TxAbort);
  - IW Error (iw-error): опоздавшие сообщения (Late messages), пришедшие после завершения транзакции (ошибки логики).
- State Backend и Checkpointing:

  - использует RocksDB для хранения состояния (маппинг транзакций, счетчики, последние коммиты);
  - чекпоинты делаются раз в минуту, инкрементальные, с сохранением при отмене.

- Метрики и Мониторинг:
  - собирает метрики через Dropwizard metrics: количество поздних записей, размер транзакций, время исправления времени (watermark lag), количество ошибок и откатов;
  - метрики агрегируются по типу Pojo, сервису и модулю (благодаря подключению к noble-registry-service.

4.5. Программный компонент «Система управления распределенными базами данных (СУРБД)»

Программный компонент «Система управления распределенными базами данных» (СУРБД) гарантирует целостность данных в масштабируемой среде, упрощает разработку бизнес-логики без ручного управления транзакциями в распределенной микросервисной архитектуре, обеспечивая инициирование, commit/rollback транзакций, блокировки строк, обнаружение взаимоблокировок и длительное хранение истории транзакций в архиве.

Программный компонент СУРБД представляет собой полнофункциональный менеджер распределённых транзакций с поддержкой блокировок, идемпотентности, обнаружения взаимоблокировок, построенный на логике конечного автомата управления состояниями транзакций и широкими возможностями мониторинга и конфигурации с предоставлением унифицированного gRPC-API другим функциональным блокам и компонентами.

Состав программного компонента:

- сервис transact-manager;
- сервис transact-scheduler;
- сервис lock-scheduler;
- сервис history-manager;
- база данных Picodata Trantool ps-tm;
- база данных Picodata Trantool ps-lm;
- база данных Postgres ps-tm-archive;
- база данных HBase ps-tm-archive.

Программный компонент обеспечивает реализацию следующих ключевых функциональных возможностей:

- Управление транзакциями:
  - инициация новой транзакции (начало, генерация идентификатора и последовательности);
  - подтверждение транзакции (commit);
  - откат транзакции (rollback).
- Блокировка строк:
  - выдача блокировок на строки таблиц;
  - проверка и освобождение блокировок.
- Обнаружение взаимоблокировок (deadlockов), основанное на:
  - построении графа ожиданий и алгоритме поиска циклов блокировок;
  - предоставлении информации о блокирующих транзакциях.
- Конечный автомат для жизненного цикла транзакций:
  - создание/обновление версии строк;
  - подтверждение и откат;
  - время ожидания завершения транзакции (таймаут) для устранения проблем с избыточно длительными транзакциями.
- Поддержка хранилищ данных о транзакциях и блокировках:
  - оперативные (текущие данные):
  - Picodata Tarantool для всех активных и недавно закрытых транзакций;
  - Picodata Tarantool для всех открытых блокировок;
  - архивные (исторические данные):
    - Postgres для всех закрытых транзакций, созданных на протяжении работы системы на глубину 6-10 месяцев;
    - HBase для всех закрытых транзакций, созданных когда-либо на протяжении всей работы системы.

4.5.1. Сервис transact-manager

Transact-manager представляет собой сервис, предназначенный для предоставления gRPC-API, обеспечивающий возможность управления транзакциями, которые используются остальными модулями системы. Реализует распределенную модель управления транзакциями, обеспечивая изоляцию данных, обнаружение и разрешение взаимоблокировок, а также работу с чтением архива завершенных транзакций.

Сервис transact-manager обеспечивает следующие функциональные возможности в части работы с транзакциями:

- beginTx: инициация транзакции с указанием:
  - уровня изоляции транзакции;
  - таймаута операции;
- commitTx: успешное завершение транзакций с указанием завершаемой транзакции;
- abortTx: откат указанной транзакции с поддержкой указания причины отмены;
- acquireRowVersionLock: запрос на получение блокировок указанных версий строк (ключевой механизм MVCC) с указанием:
  - типа блокируемой сущности;
  - ссылки на версию блокируемой сущности;
  - типа блокировки;
- getTxStatus: проверка статуса транзакции (с поддержкой работы с завершенными транзакциями из архивной БД);
- waitForTx: ожидание завершения другой указанной транзакции.

Для обеспечения хранения имеющихся транзакций transact-manager использует распределенный шардированный кластер Picodata Tarantool, хранящий сведения о транзакциях.

Для обеспечения высокой производительности внутри transact-manager использует локальный кэш статусов транзакций для снижения нагрузки на кластер Tarantool:

- Сервис transact-manager обеспечивает следующие детальные функциональные возможности в части работы с блокировками:
  - createRowVersion: регистрация новой версии строки.
  - createRowVersionLock: создание блокировки на версию строки.
  - waitForRowVersionLock: ожидание освобождения блокировки.
  - finishTx: удаление блокировок после завершения транзакции.
  - checkForDeadlock: анализ графа ожидания для поиска взаимоблокировок.

Transact-manager поддерживает механизм обнаружения и разрешения взаимоблокировок, а также предоставление метрик Prometheus (работа с БД, работа с API).

4.5.2. Сервис transact-scheduler

Transact-scheduler представляет собой сервис, предназначенный для системы распределенной транзакционности, обеспечивающий обнаружение и разрешение конфликтов (таймауты, взаимоблокировки) и надежную доставку событий о завершении транзакций в условиях распределенной среды системы.

Сервис transact-manager обеспечивает следующие функциональные возможности в части работы с блокировками и транзакциями:

- Поиск и отмена транзакций, по которым истекло время существования. При обнаружении таких транзакций сервис инициирует их отмену посредством обращения к сервису transact-manager.
- Поиск и снятие блокировок, по которым обнаружены взаимоблокировки, также обращаясь для снятия блокировки к transact-manager.
- Отправка в Kafka уведомлений о завершении транзакций для модуля Окна целостности.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.5.3. Сервис history-manager

History-manager представляет собой сервис, предназначенный для управления историей транзакций в системе. Основная задача сервиса – архивация данных из оперативного хранилища Picodata в «горячий» архив в виде базы данных PostgreSQL с последующим переносом в «холодное» архивное хранилище HBase, а также верификация корректности этих данных.

Сервис history-manager обеспечивает следующие функциональные возможности в части работы с архивацией закрытых транзакций:

- Архивация транзакций из оперативного хранилища: сервис получает список завершенных транзакций из оперативного хранилища (Picodata Tarantool) и сохраняет их в «горячий» архив закрытых транзакций (PostgreSQL) для обеспечения целостности и быстрого доступа к метаданным даже для закрытых транзакций.
- Архивация транзакций из «горячего» хранилища: сервис получает список завершенных транзакций из горячего хранилища (PostgreSQL) и сохраняет их в «холодный» архив закрытых транзакций (HBase) для возможности долгосрочного хранения всех транзакций в системе.
- Верификация – регулярная проверка целостности данных: сервис сравнивает записи в PostgreSQL с записями в HBase, чтобы убедиться, что архивация прошла успешно и данные не повреждены.
- Управление партициями: автоматическое создание новых партиций в PostgreSQL для таблиц хранения сведений о транзакциях для обеспечения оптимальной производительности PostgreSQL и упрощения операция обслуживания базы данных.

Особенности реализации:

- Автоматически обеспечивает формирование ключа хранения таким образом, чтобы обеспечить равномерную загрузку всех серверов кластера HBase, что позволяет обеспечить равномерное распределение записей по регионам HBase.

- Позволяет во время выполнения корректировать выполняемые работы по архивации, а также их расписание.

4.5.4. Сервис lock-scheduler

Lock-scheduler представляет собой сервис, предназначенный для системы распределенной транзакционности, обеспечивающий обнаружение и корректное освобождение брошенных блокировки.

Сервис lock-scheduler обеспечивает следующие функциональные возможности в части работы с блокировками и транзакциями:

- Поиск и снятие блокировок, которые оказались брошенными. Освобождение блокировки в таком случае происходит после истечения установленного времени.
- Набор метрик Prometheus (работа с БД, работа с API).

4.6. Программный компонент «Шлюз для фронтенда»

Программный компонент «Шлюз для фронтенда» предоставляет единый API для клиентских приложений, что упрощает интеграцию и ускоряет разработку UI/UX, скрывая сложность устройства большого количества групп различных сервисов, объединяя все под общим механизмом запросов на языке GraphQL и позволяя работать с данными из gRPC-API, Apache Flink и OpenSearch, представляя собой связующий слой между клиентскими приложениями и внутренними функциональными блоками.

Шлюз для фронтенда реализуется в виде следующих составляющих:

- сервис graphql-mesh, который представляет собой полнофункциональный GraphQL-Mesh фреймворк, способный объединять данные из различных источников (gRPC, Apache Flink, Opensearch) в единый GraphQL-интерфейс с возможностью динамической генерации схемы.

Ключевые особенности:

- Модульность – каждый адаптер источников данных реализован как отдельный функциональный блок с собственными конфигурациями.
- Асинхронность – все fetcheры работают в асинхронном режиме, что позволяет эффективно обрабатывать множество запросов одновременно.
- Расширяемость схемы – через extenders можно добавить произвольные поля/операции без изменения исходных protobuf-описаний.
- Интеграция с Apache Flink и Opensearch – Read-View адаптер автоматически генерирует GraphQLзапросы к Flink-базам данных, включая фильтрацию и сравнение полей.

- Кеширование и ретраи – конфигурации позволяют гибко настроить поведение при ошибках и повторных попытках.
- Инструмент ui-task-manager представляет собой реализацию менеджера задач конечного пользователя, который позволяет выполнить:
  - Регистрацию новых задач.
  - Получение списка активных задач.
  - Обновление состояния существующих задач.
  - Публикацию результатов выполнения.
  - Получение результата для задач (готово/в работе).

  Также предоставляет набор метрик Prometheus (работа с БД, работа с API).

4.6.1. Сервис graphql-mesh

Graphql-mesh представляет собой комплексный инфраструктурный сервис, который превращает набор микросервисов, общающихся через gRPC, в удобное GraphQL-представление с поддержкой асинхронных операций, транзакций и динамического расширения схемы. Сервия берет на себя сложную работу по маппингу типов, управлению контекстом транзакций и интеграции с внешними системами (noble-registry-service, витрины данных, НСИ и т.д.).

Такая общая точка позволяет для структурных частей фронтенд части скрыть сложное внутреннее устройство нижележащих частей за простым и понятным способом коммуникации, учитывающим весь объем объектной модели и доступных операций в системе путем агрегации данных из различных источников (преимущественно gRPC-микросервисов), и предоставляет их в виде GraphQL API.

Ключевые возможности и архитектура:

- Генерация GraphQL-схемы на базе структур данных, описанных в прикладных адаптерах доступа к конкретным gRPC-сервисам:
  - сканирует пакеты на наличие классов, отмеченных специальной аннотацией;
  - извлекает метаданные о типах запросов и ответов;
  - строит карту связанных типов, определяя вложенные сообщения Protobuf;
  - обрабатывает обертки Protobuf, превращая их в соответствующие GraphQL-типы.
- Генерация GraphQL-схемы на базе отдельных JSON-файлов, формируемых под отдельные задачи.
- Поддержка инициации создания и сопровождения распределенных транзакций, в т.ч. при работе с группами нижестоящих сервисов для обеспечения согласованности данных (транзакции, блокировка строк, уровни изоляции):

- передает и формирует транслируемые контексты системы.
- получает из сервиса noble-registry-service сведения о сущностях системы.
- поддерживает запросы на асинхронное выполнение сложной операции с возвратом идентификатором операции.
- обеспечивает повторы при возникновении ошибок получения ответов от отдельных сервисов.
- проверяет права доступа на уровне полей GraphQL.
- реализует интеграцию с MDM-системой (БФТ ЕНСИ):
  - обеспечивая расширения типов GraphQL сведениями из НСИ;
  - поддерживая автоматическое преобразование структур данных НСИ в структуры GraphQL.
- реализует интеграцию с подсистемой витрин данных (на базе Opensearch), обеспечивая (как и в случае MDM-системой):
  - расширение типов GraphQL сведениями из витрин данных;
  - поддержку автоматического преобразования структур данных витрин данных в структуры GraphQL;
  - поддержку указания фильтров для обеспечения фильтрации сведений из витрин данных;
  - добавление новых схем витрин данных без ручного изменения кода – подстраивается автоматически к происходящим изменениям.
- реализует полностью асинхронную обработку всех запросов для минимизации потребляемых ресурсов.
- предоставляет набор метрик Prometheus (работа с БД, работа с API).

4.6.2. Сервис ui-task-manager

Ui-task-manager представляет собой сервис, предназначенный для работы с задачами конечного пользователя, а именно для регистрации, отслеживания и управления результатами выполнения асинхронных задач, обеспечивая для клиентов следующее:

- Регистрацию новые задач.
- Опрашивание статуса задачи с возможностью ожидания (poll).
- Обновление прогресса выполнения задачи.
- Сохранение результата выполнения задачи.
- Получение списка задач по различным критериям.

Сервис ui-task-manager работает с базой данных Picodata SQL для сохранения сведений об активных и завершенных задачах пользователя.

Задача пользователя позволяет хранить следующие сведения:

- Уникальный идентификатор задачи внутри системы.
- Тип внешней задачи или системы, для которой выполняется работа.
- Идентификатор задачи во внешней системе.
- Идентификатор пользователя, инициировавшего задачу.
- Текущий статус задачи.
- Время создания задачи.
- Время последнего обновления статуса или прогресса.
- Текущий прогресс выполнения задачи (строка, описывающая процент или этап).
- Результат выполнения задачи (если задача завершена).

Также предоставляет набор метрик Prometheus (работа с БД, работа с API).

4.7. Программный компонент «Сбор информации о сервисах»

Программный компонент «Сбор информации о сервисах» позволяет:

- Динамически обнаруживать доступные сервисы без статической конфигурации.
- Повысить гибкость и масштабируемость, предоставляя готовый реестр сервисов, агрегатов данных, их метаданных и зависимостей между ними.

Программный компонент «Сбор информации о сервисах» представляет собой средство обеспечения взаимосвязи различных модулей платформы, работающих в рамках микросервисной инфраструктуры, которое реализует реестр для сервисов и агрегатных сущностей, а также метаданные о них.

Программный компонент предоставляет метаданные о сервисах (конечные точки API, зависимости, типы хранения), структуру агрегатов, а также инструменты для получения информации о gRPC-сервисах. Это позволяет другим программным компонентам динамически узнавать доступные сервисы, их связи и зависимости без статической конфигурации.

Программный компонент обеспечивает реализацию следующих ключевых возможностей:

- Метаданные сервисов:
  - Общая информация.
  - Внутренний/внешний gRPC-эндпоинт.
  - Информация о мониторинговом эндпоинте.
  - Структура агрегатов.
- Зависимости:

- Другие сервисы (доступные по унифицированному API).
- Хранилища данных.
- Сторонние сервисы (с произвольным, частным API).

В состав программного компонента входит сервис noble-registry-service.

4.7.1. Сервис noble-registry-service

Noble-registry-service представляет собой сервис, являющийся централизованным хранилищем и предназначенный для регистрации различных программных компонентов (сервисов, баз данных и т.д.) с возможностью предоставления по запросу сведений о конфигурации системы.

Сервис noble-registry-service предоставляет следующие возможности:

- Регистрировать сервисы и их метаданные (адреса, версии, зависимости).
- Отслеживать состояние доступности сервисов (мониторинг).
- Хранить маппинги между различными слоями абстракции (DTO, POJO, Proto, GraphQL).
- Предоставлять API (через gRPC) для поиска сервисов, зависимостей и сущностей.

На основе базовой функциональности предоставляет в систему мониторинга онлайн-граф связей частей системы.

Для мониторинга собственного состояния предоставляет набор метрик Prometheus (работа с БД, работа с API).

Данные об имеющихся зависимостях и взаимосвязях хранятся в базе данных Postgres:

- Общая информация о сервисах:
  - базовая информация о сервисе (имя, версия, имя приложения, имя функционального модуля, транслитерация модуля);
  - корневая таблица метаданных сервиса, связывающая основную информацию с точками доступа и зависимостями.
- Точки доступа (Endpoints):
  - информация о gRPC endpoint (внутренний и внешний адреса, список gRPC-сервисов, реализуемых сервером);
  - информация о HTTP endpoint для мониторинга (целевой адрес, статус доступности).
- Сущности данных (агрегаты и структуры):
  - список сущностей-агрегатов (POJO), зарегистрированных в сервисе;

  - скалярные поля сущности (имя, тип, имя поля в POJO и БД);
  - межсервисные внешние ключи (FK), определяющие связи между сущностями;
  - контексты сохранения сущности (URL БД, схема, таблица);
  - структуры агрегатов (определяют жизненный цикл, корневую сущность, сущность истории снепшотов, индексацию);
  - сущности внутри структуры (корневая, снепшот);
  - индексы для вложенных сущностей;
  - внешние ключи внутри структуры;
  - метаданные связей между сущностями (входящие и исходящие ссылки, типы ссылок, ссылки на таблицы связей);
  - информация о таблицах связей (много-ко-многим).
- Маппинги и сопоставления типов: связей между типами данных разных слоев (DTO, POJO, Proto, GraphQL):
  - сопоставление DTO-классов и POJO-классов;
  - сопоставление Proto-классов и DTO-классов;
  - сопоставление GraphQL-типов и Proto-типов (с указанием источника запроса request_source).
- Зависимости сервисов (информация о том, от чего зависит сервис):
  - зависимости от других прикладных сервисов;
  - зависимости от баз данных/хранилищ данных;
  - тип хранилища (Kafka, PostgreSQL, Phoenix и т.д.), адрес;
  - доп. данные (например, список топиков для Kafka);
- Информация о сторонних сервисах (имя, адрес);
- Зависимости от сторонних сервисов.

Также предоставляет набор метрик Prometheus (работа с БД, работа с API).

4.8. Программный компонент «Печатные формы»

Программный компонент «Печатные формы» позволяет быстро создавать печатные документы разных типов, гарантировать их юридическую целостность и обеспечивать долговременное хранение в БФТ.ХЭД (электронный архив документов) без дополнительных вложений в доработку, т.к. требуется выделять время только на проектирование шаблонов форм.

Программный компонент реализует возможность использования подключаемых шаблонов печатных форм для нескольких встроенных механизмов формирования, что позволяет выбрать конкретный инструмент для наиболее корректного формирования печатной формы

как с точки зрения итогового вида, так и с точки зрения вопросов производительности для больших форм.

Поддерживаемые механизмы формирования печатных форм для конечного пользователя:

- Jasper Reports.
- Apache FOP.
- WeasyPrint.
- Velocity Report.
- Формирование XML по XSLT.

Состав программного компонента:

- fo-json-document;
- fo-printing-apache-fop;
- fo-printing-form;
- fo-printing-formatalbum;
- fo-printing-grpc-client;
- fo-printing-jasper;
- fo-printing-weasyprint;
- fo-service-controlle.

4.8.1. Сервис fo-json-document

Fo-json-document представляет собой сервис, предназначенный для генерации JSON-документов (печатных форм) на основе данных, полученных GraphQL-Mesh, и шаблонов Velocity.

Основные функциональные задачи сервиса:

- При поступлении нового входящего запроса с помощью Groovy-скрипта и шаблона Velocity формирует итоговый json-файл, выполняя запросы в GraphQL-Mesh и возвращая по готовности ответ.
- Постоянно отслеживает изменения в БД PostgreSQL на предмет наличия новых Groovy-скриптов и Velocity-шаблонов для поддержки онлайн подключения новых возможностей.
- Предоставляет метрики Prometheus (работа с БД, работа с API).

4.8.2. Сервис fo-printing-apache-fop

Fo-printing-apache-fop представляет собой сервис, предназначенный для генерации печатных форм (Printed Forms) в формате PDF с использованием библиотеки Apache FOP (Formatting Objects Processor).

Основные функциональные задачи сервиса:

- При поступлении нового входящего запросов формате JSON или XML, сервис загружает соответствующие схемы (XSD), шаблоны (XSLT) и зависимости из базы данных, выполняет валидацию и трансформацию данных в итоговый PDF-документ.
- Постоянно отслеживает изменения в БД PostgreSQL на предмет наличия новых схем (XSD) и шаблонов (XSLT) для поддержки онлайн подключения новых возможностей.
- Предоставляет метрики Prometheus (работа с БД, работа с API).

4.8.3. Сервис fo-printing-form

Fo-printing-form представляет собой сервис, предназначенный для управления непосредственно созданием файла печатной формы из исходных данных.

Основные задачи сервиса:

- Управление генерацией печатных форм: создание документов различных форматов (PDF, XLSX, XML, HTML) на основе шаблонов (Jasper, WeasyPrint, Apache FOP).
- Работа с БФТ.ХЭД (электронный архив документов) для загрузки/выгрузки файлов в/из архива.

Ключевые возможности и архитектура:

- Основные функции:
  - генерация документов: создание печатных форм из данных;
  - использование множественных форматов вывода: поддержка PDF, DOCX, XLSX, XML;
  - поддержка различных источников данных: работа с данными в форматах JSON и XML, включая трансформацию JSON в XML;
  - работа с БФТ.ХЭД (электронный архив документов): загрузка данных из архива (через DaClient/S3), сохранение результатов, работа с метаданными;
  - обработка электронных подписей: парсинг и валидация XML-подписей, извлечение информации из сертификатов;
  - валидация данных: проверка входных данных по XSD-схемам.
- Движки обработки: проект поддерживает несколько стратегий генерации, выбираемых динамически:

  - Apache Velocity: для генерации форматов DOCX, XLSX – шаблон распаковывается, данные подставляются через Velocity, затем архивируется обратно;
  - JasperReports: для PDF – запрос отправляется в отдельный сервис fo-printing-jasper;
  - Format Album (XSLT + WeasyPrint): для PDF – использует XSLT-преобразования (поддержка стратегий Xalan и Saxon-HE), валидацию XML по XSD, затем конвертацию HTML в PDF через сервис WeasyPrint;
  - Apache FOP (через внешний сервис fo-printing-apache-fop): альтернативный движок для генерации PDF;
  - HTML Handler: прямая конвертация HTML в PDF через WeasyPrint.
- Управление шаблонами:
  - шаблоны хранятся в базе данных (PostgreSQL);
  - поддержка выбора шаблона по версии (ByVersion), по дате актуальности (ByDate) или получение последней (Latest);
  - автоматическое определение типа обработчика на основе NS файла.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.8.4. Сервис fo-printing-formatalbum

Fo-printing-formatalbum представляет собой сервис, предназначенный для генерации PDF-документов (форматов) на основе XML/JSON данных и XSLT/XSD шаблонов, использующий внешний сервис WeasyPrint для рендеринга.

Основные функциональные задачи сервиса:

- Формирование форм по входящему запросу:
  - прием запросов с данными (в формате JSON или XML);
  - выбор соответствующих шаблонов (XSLT для трансформации и XSD для валидации);
  - трансформация данных в HTML;
  - запрос на генерацию PDF-файл к сервису fo-printing-weasyprint с возвратом полученного файла, как результата(поддерживается обработка сбоев в weasyprint с конфигурируемой стратегией повторных попыток).
- Постоянное отслеживание изменений в БД PostgreSQL на предмет наличия новых схем (XSD) и шаблонов (XSLT) для поддержки онлайн подключения новых возможностей.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.8.5. Сервис fo-printing-grpc-client

Fo-printing-grpc-client представляет собой сервис, предназначенный для реализации интеграции программного компонента «Печатные формы» с другими программными компонентами с предоставлением gRPC-API, содержащий возможность генерации печатных форм.

Основные функциональные возможности сервиса fo-printing-grpc-client:

- Прием запросов на генерацию печатных форм (ПФ) из внешних систем.
- Отправка этих запросов в Kafka для асинхронной обработки.
- Предоставление возможности отслеживания статуса задач генерации.
- Поддержка уведомлений о статусе работ (для отправки сообщений пользователям о готовности или ошибке соответствующими модулями фронт-части).
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.8.6. Сервис fo-printing-jasper

Fo-printing-jasper представляет собой сервис, предназначенный для генерации печатных форм (PDF, HTML, Excel, Word и др.) на основе шаблонов JasperReports.

Основные функциональные возможности сервиса:

- Генерация отчетов по шаблонам JasperReports (поддерживает форматы: PDF, HTML, XLSX, XML, DOCX).
- Компиляция и заполнение шаблонов данными из JSON или XML.
- Работа с шаблонами:
  - загрузка шаблонов из базы данных PostgreSQL;
  - поддержка версионирования шаблонов;
  - выбор шаблона по версии, дате или получение последней версии.
- Работа с электронной подписью (ЭЦП):
  - автоматическое добавление информации об электронной подписи на отчеты;
  - обработка сертификатов, владельцев, сроков действия;
  - вставка детализированной информации об ЭП в итоговый документ.
- Работа с PostgreSQL для хранения шаблонов.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.8.7. Сервис fo-printing-weasyprint

Fo-printing-weasyprint представляет собой сервис, предназначенный для генерации PDF-файлов на основе HTML и CSS с использованием библиотеки WeasyPrint.

Основные функциональные возможности сервиса:

- Предоставление REST API, позволяющее принимать запросов на генерацию печатных форм PDF и отдавать полученный ответ.
- Учет стилей, указанных в принимаемых CSS-файлах, прикладываемых к запросу.
- Предоставление метрик Prometheus (работа с БД, работа с API).

4.8.8. Сервис fo-service-controller

Fo-service-controller представляет собой сервис, являющийся центральным управляющим узлом, который принимает запросы на генерацию документов из различных источников (Kafka, gRPC), управляет жизненным циклом задач формирования печатных форм, реализует систему приоритезации и разбиения по независимым потокам формирования и взаимодействует с внешними системами (архивы, базы данных, кэш).

Основные задачи сервиса:

- Управление процессом генерация печатных форм: создание документов различных форматов (PDF, XLSX, XML, HTML) на основе шаблонов (Jasper, Velocity, WeasyPrint, Format Album).
- Управление задачами формирования ПФ: создание, отслеживание состояния и обработка ошибок для каждой задачи генерации.
- Приоритизация: динамическое управление очередью задач на основе стратегий (FIFO, Strict Priority, Weighted) и внешних факторов (различные стратегии выбора приоритета).
- Интеграция с БФТ.ХЭД (электронный архив документов): сохранение сгенерированных документов и загрузка исходных данных (XML).

Сервис поддерживает формирование независимых конвейеров формирования печатных форм для обеспечения горизонтального масштабирования и отказоустойчивости при сбое движков генерации печатных форм.

Текущие задачи работы конвейера печатных форм сервис fo-service-controller хранит в базе данных PostgreSQL, что обеспечивает должную отказоустойчивость в хранении очереди выполняющихся задач на формирование печатных форм.

Также предоставляется набор метрик Prometheus (работа с БД, работа с API).

4.9. Программный компонент «Аудит доступа к данным и история изменений»

Программный компонент «Аудит доступа к данным и история изменений» представляет собой набор средств обеспечения фиксирования истории изменений объектов системы, для которых требуется вести журнал вносимых конечными пользователями правок.

Программный компонент «Аудит доступа к данным и история изменений» состоит из следующих сервисов:

- entity-audit-service.

Программный компонент обеспечивает реализацию следующих ключевых функциональных возможностей:

- Фиксирование изменений в значениях атрибутов необходимых сущностей.
- Предоставление gRPC-API для отображения сведений о внесенных в объекты системы изменений для использования в других модулях системы.
- Обработка накопленных данных по истории изменений объектов системы и действий пользователей.

4.9.1. Сервис entity-audit-service

Entity-audit-service представляет собой сервис, предназначенный для приема потока изменений в значениях атрибутов различных объекто, для которых подключена функциональность отправки событий об изменениях. Он действует как централизованная аудит-база данных.

Сервис entity-audit-service обеспечивает следующие функциональные возможности в части работы с приемом и хранением вносимых в объекты системы изменений (создание, обновление, удаление):

- сбор сведений о вносимых изменениях;
- хранение собранных изменений;
- предоставление истории изменений сущностей для других модулей системы.

Ключевые возможности и архитектура:

- Сервис потребляет сообщения из топика Kafka, в который другие модули системы публикуют сведения об изменениях своих объектов.
- Учитывает транзакционные сессии (проводит фиксацию изменений только на момент успешного завершения транзакции).

- Позволяет выполнять операции записи в БД группами для настройки производительности потока записи.
- Отправляет метрики Prometheus о скорости обработки, размерах батчей и длительности операций.
- Предоставляет gRPC-API для следующих основных сценариев:
  - поиск изменений сущностей по фильтрам (имя типа сущности, идентификатор объекта, временной диапазон);
  - получение информации о транзакциях по списку ID, включая связанные с ними сущности.
- Хранит данные в Phoenix HBase.

4.10. Программный компонент «Компонент обеспечения миграции данных»

Программный компонент «Компонент обеспечения миграции данных» представляет собой набор сервисов и библиотек, позволяющих в унифицированном виде из витрин нормализованных данных исторических систем настроить и осуществить перенос данных в целевые БД новой системы.

Состав программного компонента:

- ps-migration-base;
- ps-migration-data-router;
- ps-migration-export-metadata.

4.10.1. Библиотека ps-migration-base

Библиотека ps-migration-base представляет собой базовую библиотеку, предназначенную для реализации шлюза (роутера) интеграции между системой нормализованных витрин данных исторических систем и целевыми базами данных новой системы.

Основная задача библиотеки – предоставлять базовые механизмы для получения данных о миграции из Kafka, обрабатывать их в соответствии с бизнес-логикой конкретных типов данных и отправлять результаты обработки обратно в Kafka.

Также библиотека представляет собой базовую библиотеку, содержащую базовые механизмы работы сервисов ps-migration-data-router и ps-migration-export-metadata.

Основная логика работы:

- Входящий поток: чтение сообщений из указываемого Kafka топика для входящих сообщений.

- Обработка: десериализация Protobuf-сообщений, вызов специфичных для типа данных обработчиков (разрабатывается для конкретных потоков данных независимо).
- Исходящий поток: отправка результатов в Kafka топик для результатов для конкретного потока данных.
- Для обмена используется protobuf.

4.10.2. Сервис ps-migration-data-router

Сервис ps-migration-data-router представляет собой сервис маршрутизации данных для системы миграции данных из нормализованных витрин данных исторических систем в целевые базы данных новой системы.

Сервис обеспечивает автоматизацию и оркестрацию процесса миграции данных между системами, обеспечивая надежность, приоритизацию и трассировку операций.

Ключевые возможности и концепции:

- Обработка сообщений:
  - читает сообщения из входящих топиков Kafka;
  - управляет приоритетами обработки (если в приоритетном топике появляются сообщения, он ставит на паузу регулярного потребителя и начинает обработку приоритетных, после чего, как только приоритетный топик опустеет, обработка регулярных сообщений возобновляется).
- Логика обработки миграции:
  - проверка дубликатов: перед обработкой проверяется база метаданных для пропуска уже мигрированных данных или данных с известным результатом;
  - поиск сервиса: через noble-registry-service определяются целевые сервисы/базы данных;
  - обеспечение целостности: обеспечивается транзакционная целостность изменений.
- Отправка результатов: после обработки сообщения результат (успех, ошибка, пропуск) отправляется в исходящий топик.

4.10.3. Сервис ps-migration-export-metadata

Сервис ps-migration-export-metadata представляет собой сервис для управления метаданными экспорта для системы миграции данных.

Это сервис-реестр для отслеживания метаданных экспорта. Он позволяет хранить и обновлять информацию о том, какие данные (protoType, kndId) были экспортированы или связаны с конкретным процессом с сохранением истории изменений (временные метки). Является вспомогательным сервисом для основного сервиса миграции ps-migration-data-router.

4.11. Программный компонент «Фронтенд для веб-клиента»

Программный компонент «Фронтенд для веб-клиента» представляет собой фреймворк для создания веб-приложений, позволяющий вести унифицированную разработку форм и клиентской логики для веб-интерфейса конечного пользователя в совокупности с серверной частью фронтенда в единой концепции, опирающейся на xml для разметки форм и GraphQL как единый способ работы с данными платформы.

Основные принципы построения решений на базе фреймфорка:

- Java отвечает за логику на стороне сервера.
- React предоставляет набор UI-компонентов (панели, прогресс-бары, таблицы и т.д.) с использованием SCSS/Bootstrap на стороне клиента (приложение в браузере).
- XML используется как декларативный язык для описания структуры страниц и прав доступа, позволяющий разрабатывать сложные UI без глубокого погружения в стандартные веб-технологии и ручного написания фронтенд-логики.

Состав фреймворка:

- Серверная часть содержит следующие основные блоки:
  - REST API для взаимодействия с клиентом;
  - Конфигурация и парсинг XML;
  - MVC паттерн для роутинга и обработки запросов;
  - Модуль безопасности и авторизации (опирающийся на API authz-server).
  - Ядро серверной части фреймворка: обеспечивает интерпретацию модели данных, выполняет логику приложений, управляет жизненным циклом UI-компонентов и взаимодействует с данными. Является основной системы.
- Клиентская часть содержит следующие основные блоки:
  - Ядро клиентской части (React-компоненты).
  - Блок динамической регистрация компонентов, который позволяет использовать и рендерить UI-компоненты по имени из конфигурации (XML бэкенде);
  - Типовые компоненты пользовательского интерфейса (Input, Select, Checkbox, Slider, DatePicker и т.д.), виджеты (AdvancedTable, Calendar, Chart, List, Card и т.д.), элементы формирования структуры UI (Page, Sidebar, Header, Breadcrumb, Footer, PanelRegion и т.д.);

- Типовой набор работы с состоянием: данные моделей (формы, таблицы), виджетов (пагинация, фильтры), модальных окон, попапов и т.п.
- Блок работы с API, обеспечивающий поддержку асинхронной работы с серверной стороной фронтенда системы;
- Блок работы с действиями (actions);
- Различные вспомогательные функции (парсинг форматов данных, кэширование, обработка ошибок, горячие клавиши и т.д.

Программный компонент предназначен для быстрой разработки корпоративных веб-приложений, где важна сложная логика взаимодействия с данными и валидация, но нет возможности писать много «шаблонного» фронтенд-кода. Фреймворк берет на себя генерацию UI на основе Java-моделей и XML-конфигураций.

5. Описание функциональности программного комплекса «Сбор, хранение данных и построение отчетности»

Программный комплекс «Сбор, хранение данных и построение отчетности» позволяет создать корпоративное хранилище данных, с автоматическим получением и обновлением информации, очисткой и объединением данных из различных систем – источников информации, построить витрины данных, сформировать слой для бизнес-пользователей по ключевым метрикам путем формирования оперативных, аналитических, регламентированных и графических отчетов с возможностью их рассылки и публикации во внешних ресурсах.

Программный комплекс «Сбор, хранение данных и построение отчетности» обеспечивает следующие возможности:

1) Интеграция и импорт: получение данных из систем-источников, баз данных (SQL/NoSQL) и внешних источников через API, ETL-инструменты.
2) Преобразование и хранение данных: формирование витрин данных на основе преобразования многолетних данных, полученных из систем-источников. Хранение данных в организованном массиве данных, в едином аппаратно-программном комплексе.
3) Формирование регламентированных форм отчетов и экспорт их в различные форматы (xlsx, pdf, docx и др.). Рассылка регламентированных отчетов по расписанию пользователям системы.
4) Формы ввода информации для организации сбора произвольной информации с неограниченного круга участников.
5) Доступ и управление: многоуровневая система прав доступа.
6) В части обработки данных регионального сегмента единого регистра населения (далее – региональный сегмент ЕРН):
    - возможность хранения и регулярного обновления сведений о населении региона, получаемых из Федеральной государственной информационной системы ведения Единого федерального информационного регистра (ФГИС ЕРН), а также истории их изменения;
    - возможность предоставления сведений региональным потребителям к данным о населении региона по подписке и запросам;
    - возможность формирования произвольных запросов к сведениям о населении региона уполномоченным сотрудником.

Программный комплекс «Сбор, хранение данных и построение отчетности» также включает в себя программный комплекс «Решение бизнес-анализа Полиматика», который обеспечивает следующие возможности:

1) Конструктор информационных панелей (Polymatica Dashboards) – self-service инструмент для настройки и визуализации информации по ключевым бизнес-метрикам организации в красивом и понятном для восприятия пользователя виде.
2) In-Memory аналитика: высокая скорость работы на больших объемах данных в реальном времени.
3) Работа с многомерными массивами данных, поддержка Big Data.
4) Отчеты по запросу (Ad hoc) в режиме self-service: работа с данными без SQL-запросов.

Программный комплекс «Сбор, хранение данных и построение отчетности» состоит из следующих программных компонентов:

1) Программный компонент «Импорт и преобразование данных».
2) Программный компонент «БФТ.Хранилище».
3) Программный компонент «Администрирование».
4) Программный компонент «Конфигуратор объектов приложений».
5) Программный компонент «Сбор и консолидация произвольной информации».
6) Программный компонент «Формирование регламентированной отчетности (СтимулСофт)».
7) Программный компонент «Формирование регламентированной отчетности (JXLS)».
8) Программный компонент «Массовая рассылка отчетов».
9) Программный компонент «Распределенное выполнение регламентированных отчетов».
10) Программный компонент «Формирование универсального отчета».
11) Программный компонент «Предоставление данных внешним системам».
12) Программный компонент «Базовый набор аналитических отчетов о финансовом состоянии ре-гиона (муниципального образования) на основе данных продуктов семейства АЦК с применением Polymatica Analytics».
13) Программный компонент «Базовый набор аналитических отчетов о контрактной деятельности региона на основе данных продуктов семейства АЦК с применением Polymatica Analytics».
14) Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта Российской Федерации с применением Polymatica Analytics».
15) Программный компонент «Хранение данных регионального сегмента ЕРН».
16) Программный компонент «Предоставление данных регионального сегмента ЕРН по подписке и запросам».
17) Программный компонент «Формирование произвольных запросов к данным регионального сегмента ЕРН».

Программный комплекс «Решение бизнес-анализа Полиматика» состоит из программных компонентов:

1. Программный компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)».

2. Программный компонент «Конструктор информационных панелей (Polymatica Dashboards)».

5.1. Программный компонент «Импорт и преобразование данных»

Программный компонент «Импорт и преобразование данных» обеспечивает следующую функциональность:

- Выполнение настроек для импорта (извлечения) данных из источников.
- Выполнение извлечения данных в соответствии с настройками (в том числе автоматически в соответствии с заданным расписанием).
- Использование графического интерфейса для разработки и настройки процедур импорта и преобразования данных.
- Просмотр результатов импорта и преобразования данных.
- Получение данных посредством:
  - обмена файлами;
  - веб-сервисов;
  - использования прямых запросов к базам данных.
- Импорт данных из структурированных файлов перечисленных форматов: xls/xlsx, xml, csv, txt, json.
- Выполнение операций по преобразованию исходных данных и формированию витрин данных в аналитическом хранилище:
  - извлечение данных из источника;
  - отбор нужных столбцов из имеющегося множества;
  - настройка порядка вывода столбцов;
  - переименование столбцов;
  - создание новых вычисляемых столбцов;
  - объединение столбцов;
  - объединение различных источников данных (конструкции join, union);
  - фильтрация данных;
  - сортировка данных;
  - группировка данных;

- обработка данных таблицы с помощью процедур на любом поддерживаемом средой исполнения трансформации языке (java, javascript, sql и пр.).

- Протоколирование выполняемых операций импорта и загрузки данных с фиксацией данных:
  - название операции поступления данных, в том числе источник данных;
  - результат выполнения (в процессе исполнения, завершено успешно, завершено с ошибкой);
  - информация об ошибке, в случае завершения с ошибкой;
  - дата и время начала и окончания.

5.2. Программный компонент БФТ.Хранилище

Программный компонент «БФТ.Хранилище» обеспечивает следующие функциональные возможности:

- Обеспечение автоматизированного поступления данных из различных источников информации, включая продукты линейки АЦК.
- Организация хранения данных в аналитическом хранилище, включая многолетние данные.
- Формирование отчетов посредством инструмента формирования регламентированной отчетности, в том числе:
  - настройка шаблонов отчетов;
  - использование данных хранилища данных Системы в качестве источника данных для формирования отчетов;
  - создание, редактирование и удаление шаблонов отчетов и форм представления отчетов;
  - конструирование форм представления отчетов с предоставлением следующих возможностей по визуализации:

– выбор полей для отображения, в том числе полей списка;

– выбор формата отображения данных (текстовый / числовой / денежный / процентный / дата / время);

– представление данных в табличном виде;

– применение фильтрации, группировки, сортировки и подведения промежуточных итогов;

– выбор формата выгрузки сформированных форм отчетов в один из форматов: XLSX, DOCX, PDF, ODT, ODS;

  - настройка места вызова отчетной формы:

– из рубрикатора;

– из формы списка объекта данных;

– из формы редактирования объекта данных;

- настройка диалогового окна параметров отчета в разрезе источников данных, включающая:

– выбор источника данных;

– обеспечение установления связи параметров источников данных и диалогового окна параметров отчета;

– определение в диалоговом окне параметров вывода отчета;

– сохранение часто используемых пользователем наборов параметров отчета;

- настройка расписания автоматического формирования отчета:

– определение параметров выполнения отчета;

– определение параметров места сохранения выполненного отчета;

– настройка возможности автоматической рассылки сформированного отчета пользователям Системы по электронной почте;

- обеспечение формирования отчета по запросу пользователя:

– вызов отчетной формы из расположения, указанного в настройках места вызова;

– заполнение параметров выполнения отчета;

– указание возможности получения сформированного отчета на адрес электронной почты пользователя;

– отправка запроса на формирование отчета на исполнение.

- администрирование системы, в том числе:
  - управление учетными записями пользователей – создание, изменение, удаление, блокировка;
  - настройка ролевого доступа – назначение пользователям:

– разрешений на выполнение функций;

– ограничений доступа к данным;

  - аутентификация и авторизация пользователей по логину и паролю;
  - поддержка условий безопасности паролей – хранение паролей в зашифрованном виде, настройка правил проверки, установка сроков действия, проверка количества неудачных попыток ввода;
  - аудит – автоматическая фиксация действий, выполняемых пользователями в системе;

- настройка структур рубрикатора для пользователей;
- автоматическое выполнение серверных заданий (системных задач) по заданному графику, хранение информации о правилах их выполнения;
- сервис нотификаций – формирование и отправка информационных сообщений пользователям, их хранение и отображение.

5.3. Программный компонент «Администрирование»

Программный компонент «Администрирование» обеспечивает следующие возможности:

- Управление учетными записями пользователей – создание, изменение, удаление, блокировка.
- Настройка ролевого доступа – назначение пользователям:
  - разрешений на выполнение функций;
  - ограничений доступа к данным.
- Аутентификация и авторизация пользователей по логину и паролю.
- Поддержка условий безопасности паролей:
  - хранение паролей в зашифрованном виде;
  - настройка правил проверки;
  - установка сроков действия;
  - проверка количества неудачных попыток ввода.
- Аудит – автоматическая фиксация действий, выполняемых пользователями в системе.
- Настройка структур рубрикатора для пользователей.
- Автоматическое выполнение серверных заданий (системных задач) по заданному графику. Хранение информации о правилах их выполнения.
- Сервис нотификаций – формирование и отправка информационных сообщений пользователям, их хранение и отображение.

5.4. Программный компонент «Конфигуратор объектов приложений»

Функциональные возможности программного компонента «Конфигуратор объектов приложений» включают в себя:

- Ведение объектных моделей приложений и данных обеспечивает добавление, изменение, удаление и настройки структур объектов приложения (конфигураций), таких как, например, справочники, документы, реестры. При этом управление объектами данных осуществляется через пользовательский интерфейс.

- Конфигуратор является инструментом для проектирования систем с объектами приложений в реальном времени, без привлечения разработчиков, и предназначен как для пользователей, так и для администраторов (аналитиков). Администраторы системы (аналитики) проектируют и настраивают объекты приложения системы, а пользователи работают с данными настроенных объектов приложения системы. Разработчики могут быть привлечены для решения редких и частных задач, которые невозможно выполнить с помощью методов и средств функционального блока Конфигуратора.
- Изменения, внесенные в объектную модель приложения через функциональный блок Конфигуратор, отражаются в системе без перезапуска сервера приложения.
- Управлению подлежат объекты, для которых не предусмотрен жизненный цикл (например, справочники) и объекты, для которых предусмотрен жизненный цикл (например, документы со статусной моделью).
- Ведение объектных моделей приложений и данных включает следующие возможности:
  - Создание объекта приложения, включая создание описания формы редактирования и формы списка создаваемого объекта приложения.
  - Редактирование созданного объекта приложения.
  - Удаление созданного объекта приложения.
  - Загрузка модели данных объекта приложения из файла.
  - Сохранение модели данных в файл.
  - Добавление в Избранное моделей данных объектов приложения.
  - Создание копии модели данных объекта приложения.
  - Построение ER-диаграммы модели данных объекта приложения.
  - Просмотр таблицы базы данных модели данных объекта приложения (свойств таблицы, колонок таблицы, индексов колонок таблицы), ведение политик RLS.
  - Создание статических представлений модели данных объекта приложения.
  - Просмотр истории изменений модели данных объектной модели приложения.
  - Автоматическая генерация документации – описаний логической модели, статусной модели, форм редактирования, правил валидации.
  - Создание и настройка статусной модели для объектной модели приложения.
  - Управление данными объекта приложения осуществляется через пользовательский интерфейс.
  - Ведение записей с данными включает следующие возможности:
    - добавление, редактирование, копирование, удаление записей;
    - просмотр истории изменения записи;

  - ведение шаблона для записи;
  - наложение электронной подписи для одной или нескольких записей;
  - изменение статуса записи;
  - изменение нескольких выбранных записей;
  - загрузка данных из файлов формата XLS.

- Использование конструктора REST API – быстрая разработка API на основе HTTP методов без привлечения разработчиков.
- Ведение библиотеки скриптов – создание, хранение и обновление библиотеки скриптов для применения в объектных моделях данных.
- Управление сценариями – автоматизация действий системы с помощью применения функциональных блоков, выполняющих определенные действия.
- Управление импортом и экспортом моделей данных – выгрузка моделей в файлы zip-архива и загрузка из этих файлов.
- Управление репозиториями – ведение каталога репозиториев (обновлений объектных моделей данных) в виде файловой системы.
- Самодиагностика – хранение и запуск проверок объектных моделей приложений.
- Ведение производственного календаря – настройка информации о количестве рабочих, выходных и праздничных дней, а также норме рабочего времени за указанный период.
- Управление подписываемыми данными – ведение и настройка правил подписания пользователями экземпляров объекта приложения и их вложений, а также правил автоматической проверки наличия и валидности ЭП пользователей под экземплярами объекта приложения и их вложениями со стороны системы.
- Управление импортом данных из файлов формата XLS – настройка правил и загрузка данных в объектные модели приложений.

5.5. Программный компонент «Сбор и консолидация произвольной информации»

Программный компонент позволяет организовать сбор информации и формировать сводные отчеты, а также:

- Решать широкий спектр задач по сбору произвольной информации без привлечения разработчиков.

- Формировать сводные консолидированные отчеты на основе полученной информации с неограниченного круга участников процесса (например, ГРБС, муниципальные образования, учреждения).
- Сокращать издержки и время на сбор и консолидацию отчетности.

Программный компонент обеспечивает следующую функциональность:

- Настройка экранных форм для сбора информации под решаемую задачу без участия разработчика, включая:
  - настройку полей ввода различных типов – расчетных, числовых, логических, текстовых, с выбором из справочника, в виде прикрепленных файлов, в виде таблиц, дат, обязательных для заполнения полей;
  - настройку формул для расчетных полей.
- Ввод информации в настроенную экранную форму такими способами:
  - ручной ввод;
  - частичное автоматическое заполнение данными из аналитического хранилища (делается по заказу под конкретные требования Заказчика).
- Формирование отчетов (с возможностью печати и выгрузки в файлы различных форматов):
  - по отдельной форме, заполненной на дату отдельным пользователем.
  - сводного консолидированного отчета, который содержит сводную информацию на основе данных, заполненных всеми пользователями.
- Разграничение прав на выполнение функций для разных групп пользователей.

5.6. Программный компонент «Формирование регламентированной отчетности (Стимулсофт)»

Программный компонент обеспечивает следующие возможности:

- Использование аналитического хранилища данных в качестве источника данных для формирования отчетов.
- Создание, редактирование и удаление шаблонов отчетов и форм представления отчетов.
- Конструирование форм представления отчетов с предоставлением следующих возможностей по визуализации:
  - выбор полей для отображения, в том числе полей списка;
  - выбор формата отображения данных (текстовый / числовой / денежный / процентный / дата / время);
  - представление данных в табличном виде;

  - применение фильтрации, группировки, сортировки и подведения промежуточных итогов;
  - выбор формата выгрузки сформированных форм отчетов в один из форматов: XLSX, DOCX, PDF, ODT, ODS.

- Настройка места вызова отчетной формы:
  - из рубрикатора;
  - из формы списка объекта данных;
  - из формы редактирования объекта данных.
- Настройка диалогового окна параметров отчета в разрезе источников данных, включающая:
  - выбор источника данных;
  - обеспечение установления связи параметров источников данных и диалогового окна параметров отчета;
  - определение в диалоговом окне параметров вывода отчета;
  - сохранение часто используемых пользователем наборов параметров отчета.
- Обеспечение формирования отчета по запросу пользователя:
  - вызов отчетной формы из расположения, указанного в настройках места вызова;
  - заполнение параметров выполнения отчета;
  - указание возможности получения сформированного отчета на адрес электронной почты пользователя;
  - отправка запроса на формирование отчета на исполнение.

5.7. Программный компонент «Формирование регламентированной отчетности (JXLS)»

Программный компонент обеспечивает следующие возможности:

- Использование аналитического хранилища данных в качестве источника данных для формирования отчетов.
- Создание, редактирование и удаление шаблонов отчетов.
- Использование в качестве шаблона отчета файла, настроенного в офисном ПО, работающим с форматом XLSX – с поддержкой всех возможностей оформления, которые предоставляет данный формат.
- Обработка разметки ячеек отчета, указывающей значения для подстановки. Для определения значений ячеек могут использоваться:

- обращения к запросам к витрине данных аналитического хранилища с указанием выбираемых кодов для используемых размерностей;
- формулы, аргументами которых выступают запросы к витрине данных;
- формулы, поддерживаемые в ПО, работающим с форматом XLSX.

- Использование скрипта предварительной подготовки данных, который обеспечивает возможность гибкой обработки параметров, отбора и нужного структурирования данных отчета.
- Настройка места вызова отчетной формы:
  - из рубрикатора;
  - из формы списка объекта данных;
  - из формы редактирования объекта данных.
- Настройка диалогового окна параметров отчета в разрезе источников данных, включающая:
  - выбор источника данных;
  - обеспечение установления связи параметров источников данных и диалогового окна параметров отчета;
  - определение в диалоговом окне параметров вывода отчета;
  - сохранение часто используемых пользователем наборов параметров отчета.
- Обеспечение формирования отчета по запросу пользователя:
  - вызов отчетной формы из расположения, указанного в настройках места вызова;
  - заполнение параметров выполнения отчета;
  - указание возможности получения сформированного отчета на адрес электронной почты пользователя;

  отправка запроса на формирование отчета на исполнение

5.8. Программный компонент «Массовая рассылка отчетов»

Программный компонент «Массовая рассылка отчетов» реализует следующие возможности:

- Настройка заданий на автоматическое формирование отчетов. В настройке указываются:
  - отчет, который требуется формировать;
  - формат формируемого файла отчета;
  - значения параметров отчета;

  - способ дистрибуции отчета (можно выбрать несколько одновременно):
  - сохранить в хранилище Системы;
  - выгрузить в заданную папку;
  - отправить получателям по электронной почте;
  - список получателей отчета по электронной почте через их выбор:
    - из отдельного справочника получателей;
    - из списка пользователей Системы;
    - в виде списка электронных адресов;
  - формат электронного письма с указанием отправителя, заголовка и текста;
  - условия запуска отчета, определяющие периодичности запуска отчета, и пользователя Системы, от имени которого выполняется задание.
- Выполнение заданий в соответствии с заданными настройками.
- Фиксация результатов формирования отчетов в виде журнала, в котором отображаются характеристики процесса формирования и отправки отчета:
  - имя отчета;
  - идентификатор отчета;
  - наименование задания формирования отчета;
  - дата и время создания отчета;
  - дата и время завершения формирования;
  - состояние формирования;
  - состояние отправки;
  - сообщение об ошибке.

5.9. Программный компонент «Распределенное выполнение регламентированных отчетов»

В части формирования отчетов программный компонент обладает возможностями аппаратного и программного масштабирования по мере возрастания нагрузки, связанной как с ростом числа пользователей и количества одновременно выполняемых процессов, так и с увеличением объема обрабатываемой информации.

Масштабирование заключается в подключении новых серверов приложений. Сервер отчетов предназначен для автоматического и, при необходимости, ручного распределения нагрузки между серверами приложений. Он позволяет осуществлять мониторинг и управление запуском отчетов на исполнение.

Основные функции работы сервера отчетов состоят в следующем:

- Настраиваются список и параметры серверов приложений, которые участвуют в работе сервера отчетов.

- Активность серверов приложений отражается в экранной форме, через которую можно:
  - наблюдать состояние серверов;
  - управлять серверами (останавливать или возобновлять их работу).
- Обрабатывается единая очередь заданий, отчеты в которую поступают при штатном запуске отчетов пользователями.
- Задания на обработку отчетов выдаются на наименее загруженный сервер. При этом:
  - загрузка определяется по количеству отчетов, выполняемых на сервере в текущий момент времени;
  - количество одновременно обрабатываемых одним сервером отчетов определяется настройкой (по умолчанию – количеством ЦПУ, выделенных JVM (Java virtual machine)).
- Задания выдаются с учетом приоритета – в первую очередь с более высоким приоритетом.
- Если сервер остановлен пользователем, его задания передаются другим действующим серверам.
- Очередь заданий сервера отчетов отражается в экранной форме. Через эту форму можно:
  - наблюдать характеристики и состояние отчетов в очереди;
  - управлять отчетами – приостанавливать выполнение, менять приоритет, менять сервер приложений, возобновлять выполнение, отменять выполнение.
- Предусмотрена возможность мониторинга показателей работы сервера отчетов (определенных характеристик за единицу времени). Типовые показатели:
  - размер результирующего файла;
  - отчетов выполнено;
  - отчетов с ошибками;
  - отчетов в очереди.
- Для взаимодействия внешних систем с сервером отчетов имеется API.

5.10. Программный компонент «Формирование универсального отчета»

Программный компонент «Формирование универсального отчета» позволяет выгружать логически связанные данные (например, реквизиты электронных документов), полученные из транзакционных систем, в таблицу формата XLSX с предоставлением пользователю

возможности самостоятельно выбирать состав колонок и фильтров отчета из преднастроенных. При этом в отчет выгружаются данные без дополнительной обработки и расчетов.

Программный компонент обеспечивает следующую функциональность:

- Настройка отчета:
  - настройка метаданных отчета, определяющих состав доступных колонок и фильтров, и их соотношение с атрибутами таблиц базы данных;
  - определение в метаданных:
    - иерархического списка разделов/подразделов/колонок отчета с указанием порядка их следования;
    - атрибутов каждой колонки отчета:
      - таблицы базы данных, из которой берутся данные;
      - поля таблицы базы данных, из которой берутся данные;
      - характеристик справочника, с которым связано поле;
      - наличия или отсутствия фильтрации по полю (в том числе возможен вариант, когда поле не выводится в отчет, но по нему выполняется фильтрация);
  - настройка метаданных как через экранную форму, так и через импорт из файла формата XLSX определенной структуры;
  - настройка экранных форм справочников для задания фильтров.
- Формирование отчета с возможностью:
  - выбрать поля для фильтрации;
  - указать в выбранных полях значения фильтров;
  - выбрать колонки для отображения;
  - сохранить указанные выше настройки в виде профилей для последующего использования;
  - сформировать отчет в XLSX формате с учётом выбранных колонок и заданных фильтров.

5.11. Программный компонент «Предоставление данных внешним системам»

Программный компонент обеспечивает следующую функциональность:

- Передача данных во внешние системы путем обращения внешней системы напрямую к базе данных аналитического хранилища.
- Передача данных во внешние системы в формате JSON путем обращения внешней системы к аналитическому хранилищу через веб-сервис.

5.12. Программный компонент «Базовый набор аналитических отчетов о финансовом состоянии региона (муниципального образования) на основе данных продуктов семейства АЦК с применением Polymatica Analytics»

Программный компонент «Базовый набор аналитических отчетов о финансовом состоянии региона (муниципального образования) на основе данных продуктов семейства АЦК с применением Polymatica Analytics» позволяет решать комплекс задач по анализу ключевых направлений финансовой деятельности субъекта Российской Федерации (муниципального образования) с применением Polymatica Analytics и компонента построения регламентной отчетности и включает в себя:

- Пять OLAP-кубов по направлениям:
  - доходная часть бюджета;
  - расходная часть бюджета;
  - источники финансирования дефицита;
  - государственные программы;
  - межбюджетные трансферты;
- Отчетность по многолетним данным муниципальных образований.

Программный компонент реализует следующие возможности:

- Формирование отчетности по многолетним данным местных бюджетов посредством конструктора отчетов, в том числе по тематикам:
  1. По основным показателям доходной части местных бюджетов, включая:
     - группировку показателей в разрезе муниципальных образований;
     - детализация информации по видам доходов;
     - возможность фильтрации и выбора даты отображения информации;
     - сопоставление данных по плановым и фактическим показателям;
     - сопоставление данных за разные периоды;
     - изменения к предыдущему периоду.
  2. По основным показателям расходной части местных бюджетов, включая:
     - группировку показателей в разрезе муниципальных образований;
     - группировку по видам расходов;
     - группировку по отраслям;
     - возможность фильтрации и выбора даты отображения информации;
     - сопоставление данных по плановым и фактическим показателям;

- сопоставление данных за разные периоды;
- изменения к предыдущему периоду.

- Формирование OLAP-кубов посредством Polymatica Analytics с табличным и графическим отображением по следующим направлениям:

1. Доходная часть бюджета:

1.1. Формирование сводной и детальной информации, о структуре и динамике исполнения доходов за последние несколько лет, включая данные:

- о группах и видах доходов бюджета;
- о распределении доходов бюджета по ГАДБ;
- об исполнении по доходам в течение года;
- о структуре доходов в разрезе групп/видов/бюджетов/периодов.

1.2. Предоставление расчетных показателей структуры и динамики исполнения доходов, включая:

- процент исполнения;
- неисполненный остаток;
- информацию об изменении к предыдущему периоду в абсолютном выражении по плановому и фактическому значению;
- темп роста;
- темп прироста;
- долю в общем объеме доходов;
- долю собственных доходов бюджета.

2. Расходная часть бюджета:

2.1. Формирование сводной и детальной информации о структуре и динамике исполнения расходов за последние несколько лет, включая данные:

- о направлениях, отраслях (разделах/подразделах) и видах расходов бюджета;
- о распределении расходов бюджета в разрезе ГРБС;
- об исполнении по расходам в течение года;
- о структуре расходов в разрезе разделов/видов/направлений/бюджетов/периодов.

2.2. Предоставление расчетных показателей структуры и динамики исполнения расходов, включая:

- процент исполнения;
- неисполненный остаток;
- информацию об изменении к предыдущему периоду в абсолютном выражении по плановому и фактическому значению;
- темп роста;

- темп прироста;
- долю в общем объеме расходов;
- долю расходов на социальную сферу в общем объеме расходов.

3. Межбюджетные трансферты:

3.1. Формирование сводной и детальной информации об исполнении полученных и переданных межбюджетных трансфертов по различным уровням бюджета, включая данные:

- о формах (дотации, субсидии, субвенции, иные МБТ) и видах трансфертов, полученных соответствующим бюджетом, в том числе с учетом источника получения МБТ;
- о формах и видах трансфертов, переданных из соответствующего бюджета, в том числе в разрезе получателей МБТ;
- о конкретной форме/виде трансферта в разрезе получателей/отправителей;
- об исполнении межбюджетных трансфертов в течение года;
- о структуре полученных/переданных межбюджетных трансфертов в разрезе форм/видов/бюджетов/периодов.

3.2. Предоставление расчетных показателей об исполнении полученных и переданных межбюджетных трансфертов, включая:

- процент исполнения;
- остаток к перечислению;
- информацию об изменении к предыдущему периоду в абсолютном выражении по плановому и фактическому значению;
- темп роста;
- темп прироста;
- долю в общем объеме трансфертов;
- нераспределенный остаток.

4. Государственные программы:

4.1. Формирование сводной и детальной информации о финансировании государственных программ, включая данные:

- о перечне и структуре государственных программ региона;
- о распределении программных расходов бюджета по отраслям и ГРБС;
- об исполнении по расходам на реализацию государственных программ/подпрограмм/основных мероприятий в течение года;
- о структуре программных расходов в разрезе программ/отраслей/ГРБС.

4.2. Предоставление расчетных показателей о финансировании государственных программ, включая:

- процент исполнения;

- неисполненный остаток;
- информацию об изменении к предыдущему периоду в абсолютном выражении по плановому и фактическому исполнению проектов;
- темп роста;
- темп прироста;
- долю в общем объеме расходов на государственные программы;
- долю программных расходов бюджета.

5. Источники финансирования дефицита бюджета:

5.1. Формирование сводной и детальной информации об исполнении бюджета в части источников финансирования дефицита бюджета (их привлечения и погашения), включая данные:

- о группах (внутренние и внешние) и видах источников финансирования дефицита бюджета;
- о распределении источников финансирования дефицита бюджета по ГАИФДБ;
- об исполнении по источникам финансирования дефицита бюджета в течение года;

5.2. Предоставление расчетных показателей об исполнении бюджета в части источников финансирования дефицита бюджета, включая:

- процент исполнения;
- неисполненный остаток по источникам финансирования;
- информацию об изменении к предыдущему периоду в абсолютном выражении по плановому и фактическому исполнению проектов;
- темп роста;
- темп прироста.

- Самостоятельное формирование отчета на основе данных преднастроенных OLAP-кубов посредством инструмента многомерного анализа:

  Инструмент позволяет:

  - настраивать самостоятельно внешний вид отображения данных с помощью интерактивных таблиц, диаграмм, графиков;
  - настраивать самостоятельно дополнительные расчетные показатели;
  - задавать условия фильтрации данных;
  - выгружать отчеты в формат XLS;
  - выгружать интерактивные диаграммы, графики в PDF формат.

5.13. Программный компонент «Базовый набор аналитических отчетов о контрактной деятельности региона на основе данных продуктов семейства АЦК с применением Polymatica Analytics»

Программный компонент «Базовый набор аналитических отчетов о контрактной деятельности региона на основе данных продуктов семейства АЦК с применением Polymatica Analytics» позволяет решать комплекс задач по анализу размещения государственного заказа субъекта Российской Федерации (муниципального образования) с применением Polymatica Analytics и включает в себя:

- Пять OLAP-кубов:
  - сводные данные электронных документов;
  - данные планов-графиков и закупок, в них включенных;
  - данные контрактов и договоров;
  - данные извещений;
  - сводные данные для мониторинга хода исполнения многолетних контрактов.
- Преднастроенные примеры аналитических представлений на данных кубов по направлениям:
  - плановые и фактические объемы закупок;
  - проведение закупок с детализацией по способам определения поставщика или исполнителя и стадиям реализации;
  - структура закупок в разрезе условий размещения заказа (аванс, обеспечительные меры, установленные требования, преимущества);
  - сведения о реализации закупочных процессов в отношении социально значимых объектов (например, в рамках государственной программы, национальных проектов);
  - статистика и динамика заключения и исполнения контрактов;
  - экономия бюджетных средств;
  - контракты с региональными поставщиками;
  - контракты с привлечением СМП/СОНКО;
  - мониторинг многолетних контрактов.

5.14. Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта

Российской Федерации с применением Polymatica Analytics»

Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта Российской Федерации с применением Polymatica Analytics» предназначен для формирования отчетов по национальным проектам. Для применения модуля необходимы данные, содержащиеся в подсистеме «Ведения и исполнения региональных проектов» системы АЦК-Планирование (Автоматизация планирования бюджета региона (муниципального образования) – АЦК-Планирование).

Программный компонент «Базовый набор аналитических отчетов для мониторинга хода реализации региональных проектов субъекта Российской Федерации с применением Polymatica Analytics» включает в себя 3 OLAP-куба по направлениям:

- план мероприятий и его исполнение;
- целевые показатели;
- сводные данные по проектам.

По данным OLAP-кубов посредством Polymatica Analytics с табличным и графическим отображением можно сформировать следующие представления:

1. План мероприятий и его исполнение:

1.1. Формирование сводной и детальной информации об исполнении региональных проектов по событиям проекта в течение текущего года, включая данные:

- о плане мероприятий по реализации регионального проекта;
- об исполнении плана мероприятий по реализации регионального проекта;
- об исполнителях контрольной точки;
- информации о событии и статусе исполнения контрольной точки;
- об уровне контроля и риске исполнения контрольной точки.

1.2. Предоставление расчетных показателей об исполнении плана мероприятий региональных проектов, включая:

- количество дней просрочки;
- количество дней до планового срока.

1.3. Табличное и графическое представление сведений о просроченных контрольных точках по исполнителям.

2. Целевые показатели:

2.1. Формирование сводной и детальной информации об исполнении региональных проектов по целевым показателям в текущем году, включая данные:

- цели;
- показатели цели и характеристики их исполнения, в т.ч.:

- базовое значение;
- целевое значение;
- плановое значение;
- фактическое значение.

2.2. Предоставление расчетных показателей об исполнении планов по целевым показателям региональных проектов, включая:

- отклонения от целевого значения (абсолютное, относительное);
- отклонения от базового значения (абсолютное, относительное);
- процент достижения.

3. Сводные данные по проектам:

3.1. Формирование сводной и детальной информации о финансировании региональных проектов в течение текущего года, включая данные:

- о структуре региональных проектов (национальный/федеральный/региональный);
- об исполнении по расходам на реализацию региональных проектов;
- о результате проекта;

3.2. Предоставление расчетных показателей о финансировании региональных проектов, включая:

- процент исполнения;
- неисполненный остаток;
- долю в общем объеме расходов на региональные проекты;

3.3. Формирование сводной и детальной информации об источниках финансирования региональных проектов в текущем году и следующих трех лет, включая данные:

- об источнике финансирования регионального проекта;
- о результате проекта;
- план на 2 года;
- план на 3 года;

3.4. Предоставление расчетных показателей о финансировании региональных проектов, включая:

- процент исполнения;
- неисполненный остаток;
- долю в общем объеме расходов на региональные проекты;

3.5. Формирование сводной и детальной информации об участии ведомств в реализации региональных проектов в текущем году и следующих трех лет, включая данные:

- об источнике финансирования регионального проекта;

3.6. Предоставление расчетных показателей о финансировании региональных проектов, включая:

- процент исполнения;
- неисполненный остаток;
- долю в общем объеме расходов на региональные проекты;

3.7. Формирование сводной информации, об исполнении региональных проектов в текущем году, включая данные:

- о финансировании регионального проекта;
- о количестве контрольных точек (плановом и фактическом);
- о количестве целевых показателей (плановом и фактическом);

3.8. Предоставление расчетных показателей о финансировании региональных проектов, включая:

- по финансированию:
  - процент исполнения;
  - неисполненный остаток;
- по контрольным точкам:
  - процент исполнения;
  - неисполненный остаток;
- по целевым показателям:
  - процент исполнения;
  - неисполненный остаток.

5.15. Программный компонент «Хранение данных регионального сегмента ЕРН»

Программный компонент «Хранение данных регионального сегмента ЕРН» обеспечивает возможность хранения и регулярного обновления сведений о населении региона, получаемых из Федеральной государственной информационной системы ведения Единого федерального информационного регистра (ФГИС ЕРН), а также истории их изменения.

Функции компонента состоят в следующим:

- Хранение сведений о населении региона:
  - настроенная модель хранения данных соответствует структуре вида сведений в СМЭВ «Рассылка сведений о физических лицах по территориальному признаку из ЕРН (директивная рассылка)»;
  - модель обеспечивает сохранение истории изменения записи сведений о физическом лице.

- Загрузка данных:
  - загрузка XML файлов директивной рассылки, получаемых от ФГИС ЕРН;
  - протоколирование выполняемых операций загрузки сведений ЕРН с фиксацией в журнале;
  - отправка администратору уведомлений по электронной почте с информацией о результате загрузки данных.

Примечание: Транспортная часть взаимодействия через СМЭВ с ФГИС ЕРН не входит в функционал программного компонента и обеспечивается средствами региональной системы РСМЭВ.

- Просмотр и поиск данных уполномоченным пользователем:
  - просмотр данных в виде списка сведений о физических лицах региона из ЕРН
    - с возможностью отбора по условиям, фильтрации, сортировки записей;
    - с возможностью просмотра детализации информации по конкретной записи, выбранной в списке;
  - поиск сведений о физическом лице по его реквизитам
    - фамилия (обязательно);
    - имя (обязательно);
    - отчество (при наличии);
    - обязательно один из идентификаторов сведений о физического лица:
      - номер записи федерального регистра сведений о населении – ИдЕРН;
      - документ, удостоверяющий личность (серия и номер, дата выдачи);
      - идентификационный номер налогоплательщика;
      - страховой номер индивидуального лицевого счета зарегистрированного лица в системе индивидуального(персонифицированного) учета

    с последующим просмотром детализации информации по найденной записи;
  - мониторинг и логирование выполнения операций просмотра детальной информации о физическом лице.

5.16. Программный компонент «Предоставление данных регионального сегмента ЕРН по подписке и запросам»

Программный компонент «Предоставление данных регионального сегмента ЕРН по подписке и запросам» реализует следующие возможности:

- Ведение перечня поддерживаемых наборов данных, которые предназначены для отправки потребителям. Запись о наборе данных содержит:
  - код (уникальный идентификатор);
  - наименование;
  - описание (включая описание списка предоставляемых сведений о физическом лице и условия отбора физических лиц);
  - список параметров, которые участвуют в условиях отбора.
- Ведение реестра потребителей для управление доступом к данным для информационных систем-потребителей данных. Для каждого потребителя в реестре хранятся следующие сведения:
  - идентификатор потребителя (мнемоника информационной системы потребителя в СМЭВ);
  - наименование потребителя;
  - перечень наборов данных, доступных потребителю;
  - для каждого набора:
    - доступный режим получения данных («запрос-ответ» или «подписка»);
    - для режима «Подписка» – режим выгрузки и расписание;
    - для режима «Подписка» значения параметров.
- Предоставление сведений потребителям:
  - прием и обработка запросов на получение наборов данных, включая
    - проверку прав доступа потребителя к набору данных;
    - проверку корректности и соответствия параметров в запросе тем, которые разрешены в рамках согласованного набора;
    - формирование ответов в виде XML файлов;
  - выдача данных в соответствии с подписками в виде XML файлов;
  - мониторинг и логирование выполнения операций.

Примечание: Транспортная часть взаимодействия через СМЭВ с системами-потребителями данных не входит в функционал программного компонента и обеспечивается средствами региональной системы РСМЭВ.

5.17. Программный компонент «Формирование произвольных запросов к данным регионального сегмента ЕРН»

Программный компонент «Формирование произвольных запросов к данным регионального сегмента ЕРН» реализует возможность получения выборок данных в ответ на запрос уполномоченного пользователя, включая:

- ввод параметров запроса с учетом устанавливаемых ограничений;
- отправку запроса в компонент «Хранение данных регионального сегмента ЕРН»;
- получение ответа в UI-интерфейс (список физических лиц с атрибутами персональных данных);
- выгрузку в виде файла в формате xlsx;
- мониторинг и логирование выполнения операций, в том числе фиксацию обоснования выполнения уполномоченным пользователем данного запроса к персональным данным.

5.18. Программный комплекс «Решение бизнес-анализа Полиматика»

Программный комплекс «Решение бизнес-анализа Полиматика» состоит из программных компонентов:

1. Компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)».
2. Компонент «Конструктор информационных панелей (Polymatica Dashboards)».

5.18.1. Программный компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)»

Программный компонент «Инструмент многомерного OLAP-анализа (Polymatica Analytics)» реализует следующие возможности:

- Использование данных аналитического хранилища.
- Выбор размерностей и фактов для отображения:
  - предоставление перечня размерностей и фактов OLAP-куба;
  - добавление или исключение размерностей и фактов;
  - возможность создания дополнительных пользовательских размерностей (с целью классификации или группировки объектов);
  - естановление и изменение порядка расположения выбранных размерностей и фактов в колонках или строках для отображения.
- Создание вычислимых фактов.
- Просмотр данных со следующими возможностями:
  - просмотр данных в виде таблицы;

  - обеспечение просмотра данных в виде следующих диаграмм, размещаемых на аналитических панелях:
  - гистограммы;
  - диаграммы для визуализации иерархий;
  - графики;
  - диаграммы с областями;
  - точечные диаграммы;
  - круговые диаграммы;
  - смена направления расчета факта в сводной таблице (по горизонтали, по вертикали);
  - связь табличных форм с диаграммами;
  - поддержка Drill-Down (возможность углубления из агрегированных значений в более детальные уровни данных куба);
  - применение фильтрации, группировки, сортировки и скрытия данных;
- Сортировка фактов, включающая:
  - сортировку данных в табличном представлении в возрастающем или убывающем порядке;
  - изменение направления сортировки;
  - возможность отмены сортировки.
- Фильтрация данных, обеспечивающая:
  - фильтрацию в таблицах и диаграммах;
  - возможность отмены фильтрации;
  - применения фильтров по фактам в виде формулы.
- Корректировка данных (изменение значений размерностей при нахождении в данных ошибок, без прекращения аналитической работы).
- Выгрузка представлений данных OLAP-куба:
  - выгрузка и сохранение табличных данных форматах: XLS, CSV;
  - выгрузка и сохранение графических данных в виде рисунка в формате PNG.
- Настройка доступа пользователей к многомерным данным в соответствии с их полномочиями, включая:
  - настройку доступа к многомерным данным пользователю, группе пользователей;
  - настройку прав доступа к OLAP-кубу в части введения полного доступа или ограничений к следующим данным:
    - OLAP-кубу;

  - размерностям;
  - фактам.

- Настройка следующих ролей пользователя:
  - Администратор – роль пользователей, осуществляющих настройку и администрирование аналитических панелей, администрирование ролей пользователей;
  - Аналитик – роль пользователей, осуществляющих настройку и просмотр аналитических панелей.

5.18.2. Программный компонент «Конструктор информационных панелей (Polymatica Dashboards)»

Программный компонент «Конструктор информационных панелей (Polymatica Dashboards)» реализует следующие возможности:

- Использование данных аналитического хранилища.
- Создание и настройка наборов данных.
- Создание и настройка визуализации данных.
- Визуализация данных в виде:
  - столбчатой диаграммы;
  - линейной диаграммы;
  - линейчатой диаграммы;
  - круговой диаграммы;
  - паутины;
  - розы ветров;
  - древовидной диаграммы;
  - числового индикатора;
  - таблицы;
  - сводной таблицы;
  - SVG-карты;
  - картинки/текста/ссылки.
- Изменение цветовой гаммы дашборда.
- Использование фильтров для дашбордов или виджетов.
- Настройка переходов между дашбордами.
- Настройка вкладки дашбордов.

- Копипаст ссылкой на дашборд.
- Сохранение виджетов в форматах: XLSX, PNG, JPG и другие.
- Логирование действий пользователя.
- Настройка доступа пользователей или групп пользователей к виджетам, дашбордам, отчетам.

6. Описание функциональности программного комплекса «Геоинформационные сервисы»

Программный комплекс «Геоинформационные сервисы» (сокращенно «БФТ.Карта») позволяет выполнять задачи, связанные с обработкой и отображением пространственных данных, а также сопутствующей информации.

Программный комплекс «Геоинформационные сервисы» обеспечивает следующие возможности:

- навигация и отображение пространственных данных;
- управление пространственными данными и сопутствующей семантической информацией;
- поиск пространственных данных;
- выполнение картографических операций и пространственного анализа;
- формирование (печать) фрагмента карты;
- взаимодействие с НСПД и другими внешними сервисами.

Программный комплекс «Геоинформационные сервисы» состоит из программных компонентов:

- Картографическое ядро.
- Работа с векторными данными.
- Работа с растровыми данными.
- Сервис геоаналитики для формирования тепловых, кластерных и гексагональных слоев.
- Редактор объектов карты.
- Импорт и экспорт векторных данных.
- Подключение внешних тайловых сервисов.
- Кэширование тайловых слоев.
- Интерфейс администратора.
- Интерфейс пользователя.
- Поиск данных.
- Генератор электронных документов.
- Работа с карткомпонентом НСПД.
- Передача векторных данных в НСПД.

- Разбор данных из XML выписок Росреестра.

6.1. Программный компонент «Картографическое ядро»

Программный компонент обеспечивает реализацию следующих картографических функций по работе с картой и пространственными объектами, а именно:

- Навигацию по карте, включая:
  - изменение масштаба отображения карты;
  - изменение области отображения карты;
  - инструмент «Обзорная мини карта»;
  - позиционирование на объекты слоя;
- Копирование ссылки на текущее состояние карты или на объект;
- Измерения на карте, включая:
  - получение координат выбранной точки на карте;
  - измерение длин прямых и ломаных линий;
  - измерение многоугольником (площадь и периметр);
  - измерение кругом (площадь, длина окружности и радиус);
- Работа с отображением слоев, включая:
  - включение и отключение отображения слоя/всех слоев на карте;
  - тематические наборы;
  - поиск слоя в дереве по имени;
  - настройка отображения слоя - выбор стиля слоя;
  - настройка порядка и прозрачности отображения объектов слоя;
  - фильтрация отображаемых объектов слоя;
- Работа с отображением картооснов, включая:
  - отображение перечня доступных картооснов;
  - включение и отключение отображения картоосновы;
  - подключение/отключение картооснов к карте;
- Получение краткой информации о объектах активных векторных слоев, включая:
  - в выбранной точке карты («прокол» карты или в точке по координатам);
  - в границах нарисованной области (круг, многоугольник) в т.ч. с буфером;
  - пересекаемых выбранным (активным) объектом, в т.ч. с буфером;
- Отображение семантической информации по пространственному объекту в виде карточки при выборе из списка объектов (в т.ч. по результатам поиска)
- Формирование легенды по активным слоям карты;

- Отображение текущих координат курсора мыши;
- Отображение текущего масштаба карты;
- Добавление и хранение систем координат;

Добавление и хранение вспомогательных графических элементов.

6.2. Программный компонент «Работа с векторными данными»

Программный компонент обеспечивает реализацию следующих функциональных возможностей по работе с векторными данными:

- Создание, изменение и отображение на карте слоев векторных данных по протоколу OGC API – Features, Tiles.

6.3. Программный компонент «Работа с растровыми данными»

Программный компонент обеспечивает реализацию следующих функциональных возможностей по работе с растровыми данными:

- Загрузка и оптимизация растровых данных.
- Отображение на карте растровых данных в виде слоя карты.
- Отображение на карте растровых данных в виде картооснов.

6.4. Программный компонент «Сервис геоаналитики для формирования тепловых, кластерных и гексагональных слоев»

Программный компонент обеспечивает реализацию следующих функциональных возможностей по работе с векторными данными:

- Визуализация на карте векторных данных в виде MVT-слоя градиентной тепловой карты.
- Визуализация на карте векторных данных в виде MVT-слоя гексагональной тепловой карты.
- Визуализация на карте векторных данных в виде кластерного MVT-слоя.

6.5. Программный компонент «Редактор объектов карты»

Программный компонент обеспечивает реализацию функций, связанных с созданием и редактированием векторных пространственных данных в интерфейсе карты:

- Создание и редактирование объекта (точечного, линейного или полигонального типа) указанием его точек на карте или через таблицу координат.

- Создание прямоугольного объекта на карте указанием его длины и ширины.
- Создание и редактирование объекта методом загрузки объекта в слой из файла внешней системы (поддерживаются форматы geojson, shp, gpkg, csv).
- Создание объекта методом копирования геометрии другого объекта.
- Объединение и разделение выбранного объекта.
- Вырез пересечений с выбранным объектов или объектами выбранного слоя.
- Перемещение, поворот и масштабирование выбранного объекта.
- Удаление из слоя части объекта или объекта целиком.
- Редактирование атрибутивной информации.
- «Магнитная привязка» к объектам выбранных слоёв.
- Функция отменить/вернуть последние действия пользователя.

6.6. Программный компонент «Импорт и экспорт векторных данных»

Программный компонент обеспечивает реализацию следующих функциональных возможностей по работе с векторными данными:

- Загрузка векторных данных в слой из файла форматов GEOJSON, SHP, GPKG, CSV с возможностью выбора системы координат (из определенных в Системе) и сохранения семантической информации;
- Экспорт векторных данных по выбранному объекту, слою или списку объектов, сформированного в результате поиска, по в форматы GEOJSON, SHP, GPKG, CSV с возможностью выбора системы координат (из определенных в Системе) и сохранения семантической информации.

6.7. Программный компонент «Подключение внешних тайловых сервисов»

Программный компонент обеспечивает реализацию подключения и отображения на карте слоев пространственных данных из внешних тайловых сервисов по протоколам WMS и WMTS.

6.8. Программный компонент «Кэширование тайловых слоев»

Программный компонент обеспечивает реализацию кэширования Системой слоев, отображаемых по стандарту OGC API Tiles.

6.9. Программный компонент «Интерфейс администратора»

Программный компонент обеспечивает реализацию интерфейса администратора по настройке программного комплекса в части:

- Управления наборами векторных данных, включая загрузку данных и настройку полей семантической информации.
- Управления наборами растровых данных, включая загрузку растров.
- Управления слоями, включая создание слоев на основе векторных или растровых данных, а также внешних тайловых сервисов или данных НСПД.
- Управления отображением слоями, в т.ч. настройка стилей и области отображения.
- Управления списком картооснов и деревом слоев, в т.ч. тематическими наборами.
- Управления процессами кэширования слоев.
- Управления шаблонами для генератора электронных документов.
- Ведение реестра системам координат.
- Управления графическими элементами (шрифты, иконки, заливки, миниатюры).
- Настройка подключения и взаимодействия с НСПД.
- Настройкой поиска.

6.10. Программный компонент «Интерфейс пользователя»

Программный компонент обеспечивает реализацию интерфейса пользователя в части отображения пространственных данных в виде карты, управления отображением и взаимодействия с отображаемыми данными.

6.11. Программный компонент «Поиск данных»

Программный компонент обеспечивает реализацию функций, связанных с поиском пространственных объектов по значениям атрибутов:

- Полнотекстовый поиск по группе слоёв.
- Полнотекстовый поиск по выбранному слою.
- Поиск по атрибуту или группе атрибутов.
- Поиск по предустановленной маске (кадастрового номера).

6.12. Программный компонент «Генератор электронных документов»

Программный компонент обеспечивает реализацию функций, связанных с печатью и выгрузкой данных по подготовленным шаблонам, а именно:

- Печать фрагмента карты в формате PDF/PNG.
- Экспорт результатов поиска в виде списка в формате CSV по всем слоям или по отдельно выбранному слою.

6.13. Программный компонент «Работа с карткомпонентом НСПД»

Программный компонент обеспечивает реализацию взаимодействия с НСПД в части осуществления запросов и получения информации, а именно:

- Управление отображением слоев НСПД.
- Получение информации по пространственным данным карты.
- Поиск объектов, в т.ч. по атрибуту или группе атрибутов, по координатам.
- Полнотекстовый поиск сведений, документов, материалов по слоям НСПД.
- Получение слоев, тематических наборов пространственных данных.
- Фильтрация объектов слоя по значениям атрибутов.
- Печать выбранного фрагмента карты.
- Редактирование объектов слоя векторных данных, включая добавление, редактирование и удаление объектов.

6.14. Программный компонент «Передача векторных данных в НСПД»

Программный компонент обеспечивает реализацию передачи (загрузки) хранимых в базе данных Системы векторных данных в базу данных НСП.

6.15. Программный компонент «Разбор данных из XML-выписок Росреестра»

Программный компонент обеспечивает реализацию функции загрузки объектов геометрий из XML-выписок Росреестра следующих видов:

- Кадастровый план территории.
- Выписка из ЕГРН об объекте недвижимости.
- Выписка об основных характеристиках и зарегистрированных правах на объект недвижимости.

- Выписка о границе между субъектами Российской Федерации, границе муниципального образования, границе населённого пункта.
- Выписка о зонах с особыми условиями использования территорий и иных территориях.

7. Техническое описание платформы

7.1. Общая информация

БФТ.Платформа – это Российский продукт:

- Имеет 3-звенную архитектуру (web-клиент, сервер, база данных).
- СУБД:
  - PostgreSQL.
  - GreenPlum и другие.
- Брокеры сообщений ActiveMQ, Kafka.
- Программно-поисковые системы ElasticSearch и OpenSearch.
- Имеет высокую совместимость:
  - сервер приложений может быть развернут на Linux или Windows;
  - WEB-интерфейс работает на всех популярных браузерах и ОС;
  - универсальный REST API для доступа к данным.
- Позволяет создавать модели объектов данных, печатных форм, визуальных форм, главного меню приложения, настроек и пр.

7.2. Архитектура платформы

На рисунке ниже представлен состав архитектурных компонентов платформы.

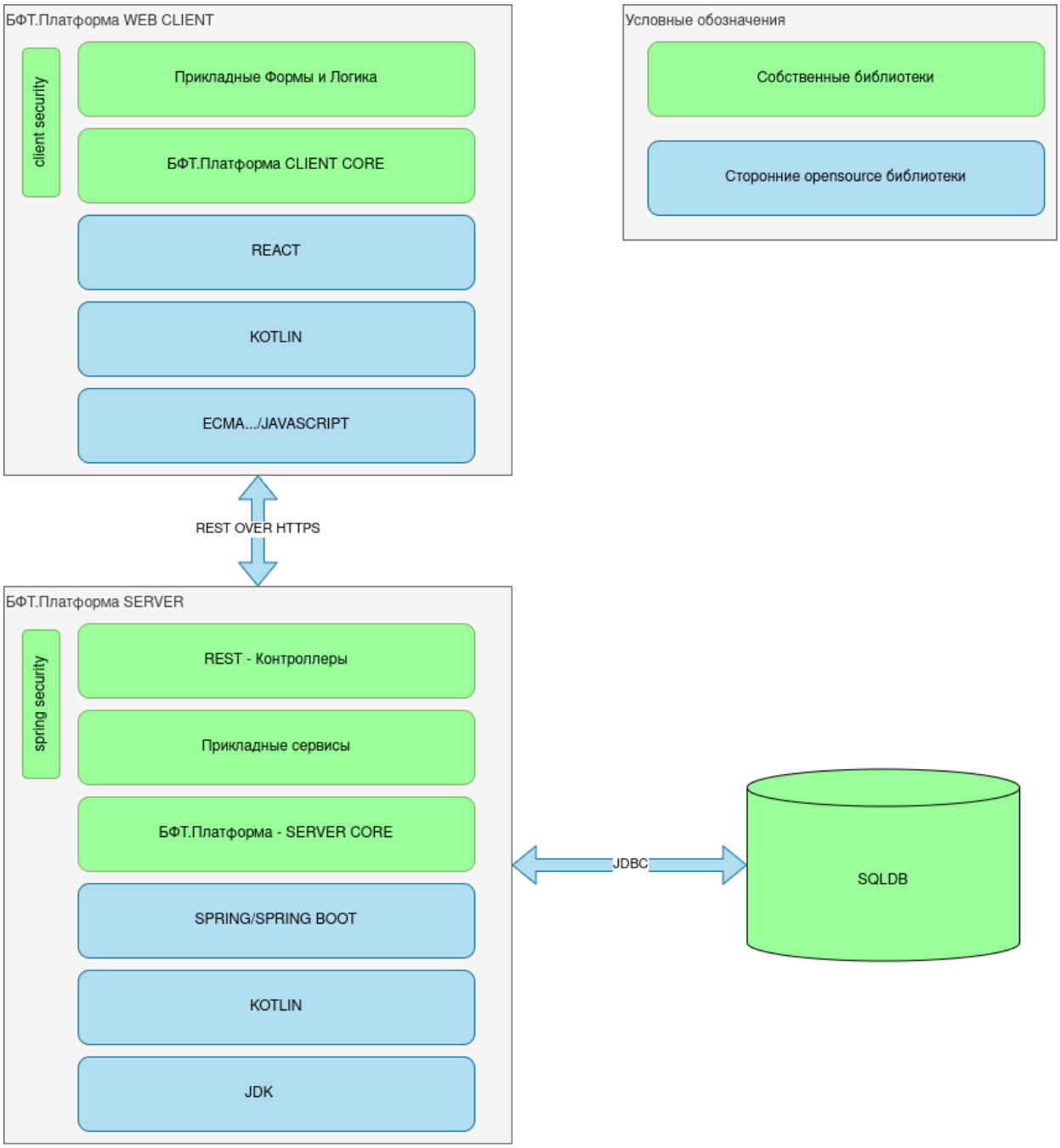


Рисунок 2 – Схема состава и связей архитектурных компонентов «БФТ.Платформы»

В таблице ниже представлено описание и назначение архитектурных компонентов «БФТ.Платформы».

Таблица 2 – Описание и назначение архитектурных компонентов «БФТ.Платформы»

	Наименование компонента	Расшифровка	Назначение/Описание
1	БФТ.Платформа WEB CLIENT	Уровень веб-клиента	Обеспечение взаимодействия с: • интернет-браузером; • серверным уровнем платформы.
	Библиотеки разработок «БФТ.Платформа» для веб-клиента		
1.1	CLIENT SECURITY	Фильтр безопасности веб-клиента	Обеспечение аутентификации и авторизации. Устанавливает на HTTP-запросы контекст безопасности в соответствии с настройками ролевого доступа для передачи на серверный уровень.
1.2	Прикладные формы и логика	Модель интерфейса пользователя	Описание визуальных форм, с которыми взаимодействует пользователь: • Форма редактирования. • Форма списка. • Панель фильтрации.
1.3	БФТ.Платформа CLIENT CORE	Общая (ядровая) для всех прикладных модулей функциональность на стороне клиента	Логика отображения визуальных компонентов. Логика связи объектов данных приложения и визуальных компонентов, позволяющая: отображать получаемые с сервера данные на визуальных формах; передавать вводимые пользователем данные с клиентского уровня на серверный.
	Языки программирования		

	Наименование компонента	Расшифровка	Назначение/Описание
1.4	KOTLIN	Статически типизированный, объектно-ориентированный язык программирования	Разработка как ядровых, так и прикладных модулей осуществляется на Kotlin, который транслируется в JavaScript для исполнения в среде браузера. Версия 1.8.22.
2	БФТ.Платформа SERVER	Серверный уровень	Обеспечение взаимодействия с: • базой данных; • клиентским уровнем платформы.
	Библиотеки разработок «БФТ.Платформа» для сервера приложений		
2.1	REST-Контроллеры	Веб-сервисы, отвечающие за обмен данными по протоколу HTTP/HTTPS	Предоставление доступа к данным внешним системам.
2.2	Прикладные сервисы	Сервисы, реализующие прикладную логику приложения	
2.3	БФТ.Платформа SERVER CORE		Общая (ядровая) для всех прикладных модулей функциональность на стороне сервера.
	Языки программирования библиотек для сервера приложений		
2.4	JAVA	Статически типизированный, объектно-ориентированный язык программирования	Разработка как ядровых, так и прикладных модулей осуществляется на Java. Версия 17.
2.5	KOTLIN	Статически типизированный, объектно-ориентированный язык программирования	Разработка как ядровых, так и прикладных модулей осуществляется на Kotlin. Версия 1.8.22.
2.6	PYTHON	Динамически типизируемый, объектно-ориентированный язык программирования	Разработка компонентов искусственного интеллекта осуществляется на Python. Версия 3.11.

	Наименование компонента	Расшифровка	Назначение/Описание
2.7	GOLANG	Статически типизированный язык программирования	Разработка как ядровых, так и прикладных модулей геоинформационного комплекса осуществляется на Golang. Версия от 1.26 и выше.
3	SQLDB	Database/База данных	Хранение данных.
4	JDBC	Протокол	Протокол обмена данными между БД и БФТ.Платформа SERVER.
5	REST OVER HTTPS	Протокол	Протокол обмена данными между БФТ.Платформа SERVER и БФТ.Платформа WEB CLIENT.

8. Лист регистрации изменений

Дата	Раздел	Описание изменений
	6. Описание функциональности программного комплекса «Геоинформационные сервисы»	Добавлено описание нового программного комплекса «Геоинформационные сервисы».
	1.2.1. Программный комплекс «Базовый состав платформы»	Из программного комплекса «Базовый состав платформы» исключен программный компонент «Геоинформационный модуль».
	3.29. Программный компонент «Создание и настройка ИИ-агентов»	Добавлено описание программного компонента «Создание и настройка ИИ-агентов».