

**Платформа разработки аналитических
приложений
Insight low code™**

INSIGHT
LOW CODE

Функционально-техническое описание

ОГЛАВЛЕНИЕ

ОПРЕДЕЛЕНИЯ.....	4
НАЗНАЧЕНИЕ ДОКУМЕНТА.....	12
ОСНОВНЫЕ СВЕДЕНИЯ О ПЛАТФОРМЕ INSIGHT LOW CODE™	12
ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПЛАТФОРМЫ.....	15
1. ОБЩАЯ АРХИТЕКТУРА РЕШЕНИЯ	15
1.1. КОНЦЕПТУАЛЬНАЯ СХЕМА РАБОТЫ ПЛАТФОРМЫ В СРЕДЕ РАЗРАБОТКИ (DEV)	17
1.2. КОНЦЕПТУАЛЬНАЯ СХЕМА РАБОТЫ ПЛАТФОРМЫ В ТЕСТОВЫХ И ПРОДУКТИВНЫХ СРЕДАХ (TST, PRD).....	19
1.3. ПОДХОД К ТРАНСПОРТИРОВКЕ КОНФИГУРАЦИИ ПЛАТФОРМЫ МЕЖДУ DEV, TST, PRD.....	21
2. ПОДКЛЮЧЕНИЕ ДАННЫХ	22
3. ПОДГОТОВКА ДАННЫХ	25
3.1. ОРКЕСТРАТОРЫ И ETL/ELT ИНСТРУМЕНТЫ	25
3.2. SDK INSIGHT LOW CODE	25
3.3. СОЗДАНИЕ ДАТАСЕТОВ В РЕДАКТОРЕ.....	28
4. ИССЛЕДОВАНИЕ ДАННЫХ.....	31
4.1. СВОДНАЯ АНАЛИТИКА	31
4.2. ВЫГРУЗКА ВИТРИН ДАННЫХ.....	32
5. КОНСТРУКТОР ПРИЛОЖЕНИЙ	33
5.1. ПАРАМЕТРЫ ВИДЖЕТОВ	40
5.2. КОМПОЗИЦИОННЫЕ ВИДЖЕТЫ.....	41
5.3. НАВИГАЦИОННЫЕ ВИДЖЕТЫ	43
5.4. ПРОЧИЕ КОМПОНЕНТЫ.....	45
5.5. CSS-ФРЕЙМВОРК INSIGHT LOW CODE™	47
5.6. ЛОГИРОВАНИЕ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ	50
5.7. ФОРМЫ ВВОДА	50
5.8. СОЗДАНИЕ ВЕРСИЙ ПРОЕКТА	51
5.9. СОВМЕСТНАЯ РАБОТА.....	52

5.10. ПУБЛИКАЦИЯ ПРИЛОЖЕНИЙ	53
5.10.1. МИКРО-ПРИЛОЖЕНИЯ.....	53
5.10.2. ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ	55
5.10.3. ПРОЧИЕ ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ.....	56
6. ВИЗУАЛИЗАЦИЯ ДАННЫХ	57
6.1. АНАЛИТИЧЕСКИЕ ВИДЖЕТЫ.....	57
6.2. ТАБЛИЦЫ	59
6.3. КАРТЫ	61
6.4. ВОЗМОЖНОСТЬ РАЗРАБАТЫВАТЬ СОБСТВЕННЫЕ ВИДЖЕТЫ	62
7. SELF-SERVICE ОТЧЕТНОСТЬ.....	62
7.1. РЕГЛАМЕНТНАЯ ОТЧЕТНОСТЬ.....	62
7.2. ВЫГРУЗКА ВИТРИНЫ ДАННЫХ	63
7.3. ВЫГРУЗКА СТРАНИЦЫ ПРОЕКТА.....	63
7.4. ВЫГРУЗКА ВИДЖЕТА	64
8. УПРАВЛЕНИЕ ДАННЫМИ И БЕЗОПАСНОСТЬ	64
9. ДОПОЛНИТЕЛЬНЫЕ ОПЦИОНАЛЬНЫЕ КОМПОНЕНТЫ	66
9.1. МОДУЛЬ БИЗНЕС-ПРАВИЛ WORKFLOW.....	66
9.2. МОДУЛЬ УВЕДОМЛЕНИЙ.....	66
9.3. МОДУЛЬ ОРГАНИЗАЦИОННОЙ СТРУКТУРЫ.....	67
10. НЕФУНКЦИОНАЛЬНЫЕ ТЕХНИЧЕСКИЕ ВОЗМОЖНОСТИ	68
10.1. ОБЩИЕ ВОЗМОЖНОСТИ.....	68
10.2. ОПЦИИ ИНСТАЛЛЯЦИИ ПЛАТФОРМЫ	68
10.3. ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ	69
10.4. ТРЕБОВАНИЯ К НАДЕЖНОСТИ	72
10.5. ТРЕБОВАНИЯ К ИНСТРУМЕНТАМ ДИАГНОСТИКИ И ЛОГИРОВАНИЯ	74

ОПРЕДЕЛЕНИЯ

Таблица 1 Список определений и сокращений

Термин	Определение
Платформа (Insight low code Platform)	Платформа Insight low code™ – совокупность функциональных и технических модулей.
Конструктор (Insight low code Designer)	Функциональный модуль Платформы Insight low code, который обеспечивает возможность визуальной настройки приложений, веб-страниц и диаграмм (визуализаций) из набора виджетов.
Плеер (Insight low code Player)	Функциональный модуль Платформы Insight low code, предназначенный для просмотра (проигрывания) разработанных в конструкторе приложений.
Авторизация	Авторизация (англ. authorization «разрешение; уполномочивание») — предоставление определённому лицу или группе лиц прав на выполнение определённых действий, а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.
АСУТП	Автоматизированная система управления технологическим процессом — группа решений технических и программных средств, предназначенных для автоматизации управления технологическим оборудованием на промышленных предприятиях.
Аутентификация	Аутентификация (англ. authentication) — процедура проверки подлинности, например: <ul style="list-style-type: none"> • проверка подлинности пользователя путём сравнения введённого им пароля (для указанного логина) с паролем, сохранённым в базе данных пользовательских логинов; • подтверждение подлинности электронного письма путём проверки цифровой подписи письма по открытому ключу отправителя; • проверка контрольной суммы файла на соответствие сумме, заявленной автором этого файла.
Датасет	Набор данных (англ. Dataset) представляет собой табличный или иной объект с данными. В случае табличных данных набор данных соответствует одной или нескольким таблицам базы данных, где каждый столбец таблицы представляет определенную переменную, а каждая строка соответствует определенной записи рассматриваемого набора данных. Каждое значение набора данных определяется как данные (ячейка данных). Наборы данных также могут иметь вложенную структуру и состоять из набора документов или файлов.
Дашборд	Информационно-аналитическая панель, панель индикаторов (англ. dashboard) — это инструмент для

Термин	Определение
	визуализации и анализа информации о бизнес-процессах и их эффективности. Данные, выводимые на панель индикаторов, обычно представлены в виде ключевых показателей эффективности.
Дисклеймер	Ограничения формулировки и контекста документа
ДСП	Данные о сотрудниках и подразделениях
Кластеризация	Способ построения распределенной архитектуры с помощью кластеров — группы компьютеров, объединённых высокоскоростными каналами связи, представляющая с точки зрения пользователя единый аппаратный ресурс. Кластер - слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и представляющихся пользователю единой системой.
Контейнеризация	Метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного. Эти экземпляры (обычно называемые контейнерами или зонами) с точки зрения пользователя полностью идентичны отдельному экземпляру операционной системы.
Контур Заказчика, on-premise	Вариант размещения Платформы на серверах и оборудовании Заказчика без использования внешней по отношению к Заказчику инфраструктуры.
Облако	Вариант размещения Платформы во внешней инфраструктуре на серверах провайдера аппаратной инфраструктуры
Оркестрация	Координация взаимодействия нескольких контейнеров
ПО	Программное обеспечение
Пользователь	Пользователь Платформы, имеющий уникальный идентификатор и права доступа к приложениям
Поставка, deployment	Комплекс программного кода, файлов, процессов, документации и инструкций для развертывания Платформы и ее обновления.
Предиктивная аналитика	Предсказательная аналитика — класс методов анализа данных, концентрирующийся на прогнозировании будущего поведения объектов и субъектов с целью принятия оптимальных решений.
Проект, внедрения	Проект Комплекс мероприятий по обследованию бизнес-процессов, развертыванию, настройке, адаптации, тестированию и ввода в промышленную эксплуатацию Платформы.
Роль	Совокупность функциональных возможностей (привилегий) пользователя Платформы, позволяющая разграничивать доступ к различным функциям Платформы. Соответствие ролям пользователей определенных функциональных возможностей определяется моделью определения прав доступа.

Термин	Определение
Семантический слой	Семантический слой — это специальные средства, позволяющие бизнес-аналитику использовать данными посредством бизнес-терминов понятной ему предметной области. Поверх хранилища данных обеспечивается специальный набор терминов, заменяющих поля и колонки базы данных на многомерные понятия, такие как измерения и метрики. Дополнительные метаданные позволяют уточнить объекты (справка по значению, короткое и длинное наименование ит.п.)
СУБД	Система управления базой данных
Фреймворк	От англ. framework — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
Active Directory (AD)	LDAP-совместимая реализация интеллектуальной службы каталогов корпорации Microsoft для операционных систем семейства Windows. Active Directory используется для хранения информации о пользователях и предоставления возможности аутентификации, хранения настроек Exchange, массового распространения настроек и пр.
Alert	Уведомление (англ. alert - предупреждений) - в программировании — это сообщение, связанное с асинхронным вызовом процедур в многопоточной (мультитредовой) системе. Возможно деление таких предупреждающих сообщений на классы, например admin alerts, error alerts и printer alerts.
API	Application Programming Interface (англ.) – программный интерфейс приложения описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
Back-end	Бэкенд – программно-аппаратная часть сервиса
BI	Business Intelligence (англ.) - обозначение компьютерных методов и инструментов для организаций, обеспечивающих перевод транзакционной деловой информации в человекочитаемую форму, пригодную для бизнес-анализа, а также средства для массовой работы с такой обработанной информацией
BPMN	Business Process Management Notation (англ.)— система условных обозначений и их описания в XML для моделирования бизнес-процессов. BPMN - язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса

Термин	Определение
CMS	Content Management System (англ.), система управления содержимым — информационная система или компьютерная программа для обеспечения и организации совместного процесса создания, редактирования и управления контентом.
CRM	Customer Relationship Management (англ.), Система управления взаимоотношениями с клиентами, прикладное программное обеспечение для организаций, предназначенное для автоматизации взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.
CSS	Cascading Style Sheets (англ.)- формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
Data Lake	Озеро данных — это система или репозиторий данных, хранящихся в естественном / необработанном формате, обычно это большие двоичные объекты или файлы. Озеро данных обычно представляет собой единое хранилище данных, включая необработанные копии исходных системных данных, данных датчиков, социальных данных и другие данные.
Data Science	Наука о данных — раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме. Объединяет методы по обработке данных в условиях больших объёмов и высокого уровня параллелизма, статистические методы, методы интеллектуального анализа данных и приложения искусственного интеллекта для работы с данными, а также методы проектирования и разработки баз данных.
DMN	Decision Model and Notation (англ.)- Модель и нотация принятия решений - стандарт для описания и моделирования повторяемых решений в организациях для обеспечения взаимозаменяемости моделей решений в организациях
Drill-down	«Углубление в данные» (drill-down, англ.) - Детализация данных относится к любым из различных операций и преобразований в табличных, реляционных и многомерных данных. Этот термин широко используется в различных контекстах, но в основном связан со специализированным программным обеспечением, разработанным специально для анализа данных.

Термин	Определение
DWH	Data Warehouse (англ.), Хранилище данных — предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчетов и бизнес-анализа с целью поддержки принятия решений в организации. Строится на базе систем управления базами данных и систем поддержки принятия решений
ELT	Extract, Load and Transform (англ.)- Извлечение, загрузка, преобразование - это альтернатива извлечению, преобразованию, загрузке, используемая в реализациях озера данных. В отличие от ETL, в моделях ELT данные не преобразуются при входе в озеро данных, а сохраняются в исходном необработанном формате, что позволяет сократить время загрузки.
ERP	Enterprise Resource Planning (англ.), планирование ресурсов предприятия. Программное обеспечение для управления бизнес-процессами, которое объединяет финансы, цепочки поставок, операции, отчетность, производство, кадры и позволяет управлять ими.
ETL	Extract, Transform and Load (англ.)- один из основных процессов в управлении хранилищами данных, который включает в себя: извлечение данных из внешних источников, их трансформацию и очистка, чтобы они соответствовали потребностям бизнес-модели, а также загрузку их в хранилище данных
Front-end	Фронтэнд – клиентская сторона пользовательского интерфейса к программно-аппаратной части интерфейса
HCM	Human Capital Management , системы управления персоналом. Основные учетные системы, реализующие, как правило, базовые учетные функции – кадровый учет, штатное расписание, табельный учет и расчет заработной платы.
HTML	HyperText Markup Language (англ.) - стандартизированный язык разметки документов в Интернете. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.
HTTPS	HTTPS — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов TLS или SSL.
In-memory database	Резидентная база данных (англ. in-memory database, IMDB) — база данных, размещаемая в оперативной памяти. Резидентная СУБД — система управления резидентными базами данных, один из видов

Термин	Определение
	программных систем, работающих в парадигме резидентных вычислений (англ. in-memory computing). Резидентные СУБД за счёт оптимизаций, возможных в условиях хранения и обработки в байтоадресуемой оперативной памяти, обеспечивают лучшее быстродействие, чем СУБД, работающие с базами данных на устройствах постоянного хранения, как правило, с блочной организацией, и подключаемых по шинным или сетевым интерфейсам. При этом размер резидентной базы данных ограничен ёмкостью оперативной памяти узла.
ITIL	Самое распространённое в мире руководство по управлению ИТ-услугами.
ITSM	IT Service Management (англ.), управление ИТ-услугами — подход к управлению и организации ИТ-услуг, направленный на удовлетворение потребностей бизнеса. Управление ИТ-услугами реализуется поставщиками ИТ-услуг путём использования оптимального сочетания людей, процессов и информационных технологий.
JDBC	Платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета <code>java.sql</code> , входящего в состав Java SE. JDBC основан на концепции так называемых драйверов, позволяющих получать соединение с базой данных по специально описанному URL.
JSON	JavaScript Object Notation (англ.) — текстовый формат обмена данными, основанный на JavaScript.
LDAP	Протокол прикладного уровня для доступа к службам каталогов
Low-code development platform	Платформа разработки с небольшими потребностями в кодирование — это программное обеспечение, которое обеспечивает среду разработки, используемую для создания прикладного программного обеспечения через графический пользовательский интерфейс и конфигурацию вместо традиционного компьютерного программирования, написанного вручную. Пользователи могут создавать простые приложения почти без помощи ИТ-специалистов.
MES	Manufacturing Execution System (англ.), система управления производственными процессами — специализированное прикладное программное обеспечение, предназначенное для решения задач синхронизации, координации, анализа и оптимизации выпуска продукции в рамках какого-либо производства.
Message broker	Брокер сообщений (англ. message broker) — архитектурный паттерн в распределённых системах; приложение, которое преобразует сообщение по одному протоколу от приложения-источника, в

Термин	Определение
	сообщение протокола приложения-приемника, тем самым выступая между ними посредником.
ODBC	Программный интерфейс доступа к базам данных, разработанный компанией Microsoft в сотрудничестве с Simba Technologies на основе спецификаций Call Level Interface, который разрабатывался организациями SQL Access Group, X/Open и Microsoft. Впоследствии CLI был стандартизован ISO.
OLAP	Интерактивная аналитическая обработка (англ. online analytical processing) — технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу.
RDBMS	Реляционная система управления базами данных (англ. Relational Database Management System), реже — система управления реляционными базами данных — СУБД, управляющая реляционными базами данных.
REST API	Representational State Transfer (англ.)— «передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
SDK	Software Development Kit (англ.)- набор средств разработки, позволяющий специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ
Single Sign-On	Технология единого входа (англ. <i>Single Sign-On</i>) — технология, при использовании которой пользователь переходит из одного раздела портала в другой, либо из одной системы в другую, не связанную с первой системой, без повторной аутентификации.
SLA	Service Level Agreement (англ.), Соглашение об уровне предоставления услуги — термин методологии ITIL, обозначающий формальный договор между заказчиком услуги и её поставщиком, содержащий описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги.
SQL	Structured Query Language (англ.) - язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.
SSD	Твердотельный накопитель (англ. Solid-State Drive, SSD) — компьютерное энергонезависимое немеханическое запоминающее устройство на основе микросхем памяти, альтернатива HDD.

Термин	Определение
SSL	SSL (англ. Secure Sockets Layer — слой защищённых сокетов) — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.
Storytelling	Сторителлинг, история - метод построения аналитических панелей, при котором информация доводится до пользователя в виде последовательного набора выводов и визуализаций, раскрывая ключевые выводы.
TLS	TLS (англ. transport layer security — Протокол защиты транспортного уровня), как и его предшественник SSL — криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет. TLS и SSL используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.
UI/UX	User Inteface/ User Experience (англ.) – совокупность пользовательского интерфейса и дизайна взаимодействия Системы с пользователем (пользовательского опыта)
Workflow	Поток работ — графическое представление потока задач в процессе и связанных с ним подпроцессах, включая специфические работы, информационные зависимости последовательность решений и работ.

НАЗНАЧЕНИЕ ДОКУМЕНТА

Настоящий документ описывает функциональные и технические возможности Платформы создания аналитических приложений Insight low code™ («Инсайт»), далее Платформа.

Дисклеймер. Вся информация, приведенная в настоящем документе, является предварительной и служит исключительно для целей ознакомления и оценки. Подробное описание Платформы является предметом Технической документации на Платформу, поставляемой с программным обеспечением.

ОСНОВНЫЕ СВЕДЕНИЯ О ПЛАТФОРМЕ INSIGHT LOW CODE™

Платформа предназначена для:

- Разработки интерактивных информационно-аналитических панелей (дашбордов);
- Разработки интерактивных приложений, использующих аналитику для поддержки процессов принятия решений;
- Обеспечения и поддержки работы ситуационных центров;
- Создания аналитических и статистических сайтов и разделов корпоративных сайтов без программирования;
- Создания мобильных приложений;
- Создания приложений под нестандартные разрешения экранов (например, больших медиа-панелей);
- Создания приложений, предназначенных как для вывода данных, так и для записи данных (социальные реакции, учет времени, проведение анкетирования и т.д.);
- Создания приложений, в которых специалисты могут самостоятельно выполнять запросы к нужным данным и генерировать необходимые отчеты без привлечения дата-специалистов (Self-service аналитики);
- Публикации интерактивных приложений со встроенной аналитикой на порталах, в средствах массовой информации и социальных сетях;
- Создания современных интерфейсов к различным приложениям и унификация стилей и точки входа сотрудника.

Платформа Insight low code™ предоставляет следующие сервисы:

- Business Intelligence (BI) - инструменты построения аналитических дашбордов и приложений для преобразования данных в форму, пригодную для бизнес-анализа (графики, таблицы, диаграммы);
- Средства работы с хранилищами данных (Data warehouse, DW) и озерами данных (Data lake) - инструменты хранения и управления хранением различной информации для аналитических целей
- Аналитическая база данных (in-memory СУБД, современные технологии, приходящие на смену OLAP)

- Управление созданием веб-приложений (CMS);
- Средства визуального проектирования приложений с минимальным программированием («low code development platform»).

Платформа содержит в составе следующие инструменты:

- Сбора, подготовки и трансформации данных (ETL);
- Встроенное in-memory аналитическое хранилище данных;
- Средства быстрой самостоятельной разработки веб-приложений и бизнес-аналитики с минимальным привлечением разработчиков (low-code development platform);
- Встроенные инструменты прогнозной и предиктивной (predictive) аналитики на базе математических алгоритмов и нейросетей;
- Бесшовную интеграцию с инструментами специалистов по работе с большими данными (data science);
- Встроенные средства моделирования бизнес-процессов (workflow);
- Встроенные средства автоматизированных уведомлений с помощью электронной почты, мессенджеров и иных средств коммуникации;
- Инструменты расширения функционала Платформы средствами сторонних разработчиков.

Типовые пользователи Платформы:

- Топ-менеджмент компаний;
- Линейный менеджмент компаний вплоть до нижних уровней управления;
- Бизнес-аналитики;
- Анонимные пользователи – пользователи интранет-порталов, читатели СМИ и другие (ограниченные информацией, не относящейся к персональным или коммерческим данным в соответствии с политиками компании)

Ключевые особенности Платформы:

- Представление данных в формате интерактивных презентаций (storytelling);
- Богатые возможности по дизайну внешнего вида, интерфейса, бизнес-логики и способов представления информации;
- Большая и расширяемая библиотека виджетов;
- Визуальный конструктор приложений с поддержкой самостоятельной настройки;
- Встраивание в процессы принятия решений – комментарии, поручения, согласования;
- Возможность работать не только «на чтение», но и на запись данных в системы;
- Высокопроизводительная in-memory база данных, поддерживающая работу с огромными массивами данных;
- Подключение любых источников данных;
- Возможности для Self-service аналитики;
- Математические и прогнозные модели с возможностью разработки собственных алгоритмов.

Платформа поддерживает работу на рабочих станциях (десктопах), мобильных устройствах, больших экранах и медиапанелях любого размера и конфигурации.

Платформа поддерживает работу на всех основных операционных системах. Взаимодействие с Платформой осуществляется через тонкий клиент (браузер). Не требуется установки дополнительного программного обеспечения на устройства конечных пользователей.

Платформа полностью построена на собственных разработках компании и технологиях с открытым кодом. Платформа не содержит проприетарных компонентов и закрытого кода. Для работы Платформы не требуется приобретения лицензий третьих лиц.

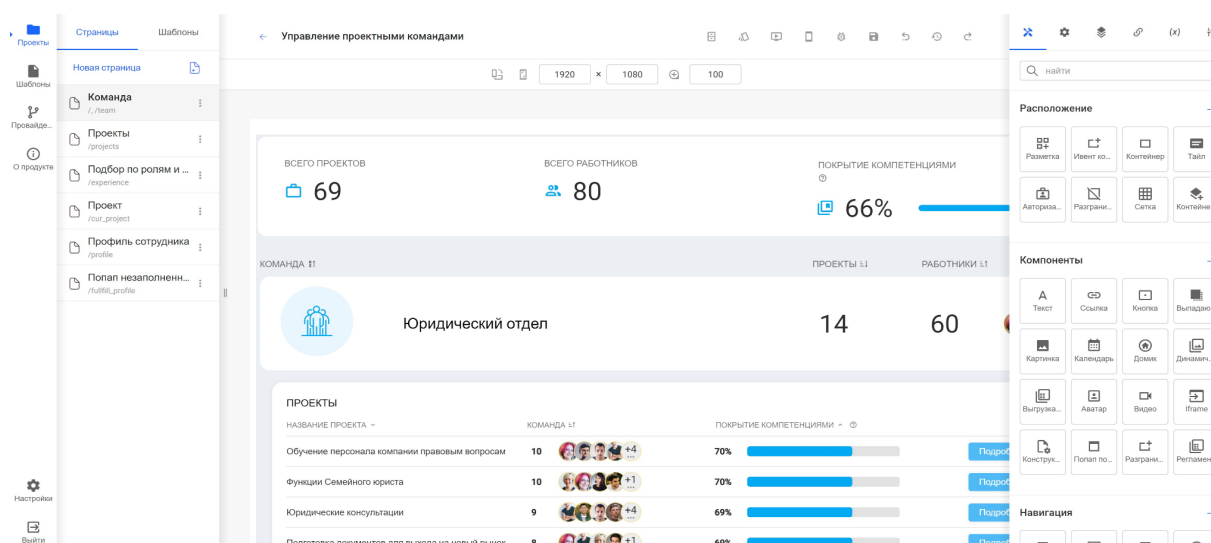
Платформа использует в качестве фундамента широко распространенные языки программирования и фреймворки, что позволяет упростить процесс поддержки и самостоятельного развития.

Платформа поддерживает необходимые требования информационной безопасности в части защиты от несанкционированного использования информации, обеспечения режимов защиты персональных и коммерческих данных. Поддерживаются необходимые стандарты аутентификации и авторизации, управление ролевой моделью полномочий на уровне функциональных полномочий и отдельных строк данных. Возможна установка Платформы в закрытый контур, не имеющий доступа в сеть Интернет. Платформа поддерживает логирование действий пользователей.

Платформа поддерживает механизмы виртуализации и поддерживает средства контейнеризации и оркестрации. Платформа обеспечивает опции вертикального и горизонтального масштабирования (кластеризацию).

Платформа обеспечивает необходимые функции мониторинга работы, интеграцию в промышленный ландшафт и возможность развертывания множественных сред для обеспечения бесперебойной работы и цикла разработки и тестирования.

Рисунок 1 Пример интерфейса Платформы



ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПЛАТФОРМЫ

1. ОБЩАЯ АРХИТЕКТУРА РЕШЕНИЯ

Функциональная архитектура решения включает несколько основных компонентов:

- Подключение данных:
 - прямые подключения к БД;
 - менеджер подключений;
 - загрузка файлов xlsx, csv;
 - коннекторы к источникам данных.
- Подготовка данных:
 - оркестраторы и etl/elt инструменты;
 - SDK - семантический слой работы с данными;
 - каталог метрик и измерений;
 - создание датасетов в Редакторе.
- Исследование данных:
 - Сводная аналитика;
 - Выгрузка витрин данных.
- Конструктор приложений
 - композиционные и навигационные виджеты, компоненты;
 - стилизация виджетов;
 - адаптивная верстка;
 - тема проекта;
 - многоуровневая навигация в приложении;
 - шина событий;
 - логирование действий пользователей;
 - формы ввода
 - публикация приложений.
- Визуализация данных:
 - аналитические виджеты, таблицы и карты;
 - разработка собственных виджетов.
- Self-service отчётность:
 - регламентная отчетность;
 - выгрузка витрин данных;
 - выгрузка страниц проекта;
 - выгрузка виджетов;
 - рассылка отчетных форм.
- Управление данными и безопасность:
 - ролевая модель.
- Дополнительные компоненты:
 - математические модели;
 - работа с бизнес-процессами (workflow);
 - модуль организационной структуры;

- шина уведомлений.

Схематично архитектура Платформы представлена на Рисунке 2.

Рисунок 2 Функциональная архитектура Платформы



В последующих разделах представлено функциональное и техническое описание основных блоков Платформы.

Платформа реализована в виде микросервисной архитектуры. Отдельные компоненты Платформы взаимодействуют между собой и с внешним окружением с помощью механизмов REST API и иных программных интерфейсов. Интеграционные потоки реализуются с помощью интеграционной шины и (при необходимости) встроенных инструментов подготовки, загрузки и трансформации данных (ETL).

Выбранная архитектура решения предоставляет возможности:

- эксплуатации решения с минимальными затратами на обслуживание;
- обеспечения максимального уровня отказоустойчивости и безопасности порталных систем;
- обеспечения репликации данных на уровне СУБД;
- оперативной реакции на возникающие задачи и ситуации в рамках порталной платформы;
- обеспечения предсказуемости в управлении всем комплексом порталных решений в процессе его развития;
- обеспечение архитектурной целостности поставляемого решения и решения по модернизации внутрикорпоративного Портала Заказчика;
- при возникновении потребности в том или ином бизнес-решении адекватно и четко выбрать способ его реализации.

Платформа учитывает планы и стратегию развития, а также модернизации в перспективе:

- на веб-уровне обеспечивается горизонтальная масштабируемость серверов приложений, балансировки, кэширования, файлового хранения, СУБД и т.д. Эти серверы могут быть настроены как обычные веб-серверы для обработки запросов пользователей, как выделенные серверы запросов или как компоненты служб;
- для обеспечения целевых значений нагрузки и требуемой функциональности системы используется in-memory СУБД (в оперативной памяти) для выполнения агрегаций данных и аналитических запросов;
- на уровне пользовательских интерфейсов используются наиболее распространенные фреймворки (библиотеки) с открытым исходным кодом;
- система поддерживает модель разработки виджетов с использованием JavaScript и библиотек и встраивание их в пользовательский интерфейс с возможностью многократного использования;
- в системе реализованы механизмы управления очередями и фоновыми задачами на технологиях с открытым кодом;
- в качестве основной СУБД используется реляционная СУБД с открытым исходным кодом;
- Платформа может размещаться на серверах в ЦОД Заказчика;
- Система поддерживает поставку и оркестрацию в виде контейнеров (Kubernetes);
- компоненты Платформы поддерживают кластеризацию для масштабирования и распределения нагрузки.

1.1. КОНЦЕПТУАЛЬНАЯ СХЕМА РАБОТЫ ПЛАТФОРМЫ В СРЕДЕ РАЗРАБОТКИ (DEV)

В среде DEV компоненты Платформы распределены между двумя контурами:

- 1) Конструктор размещается за пределами клиентского контура и доступен в качестве облачного сервиса;
- 2) Все остальные сервисы Платформы, включая сервисы для работы с данными, размещаются в закрытом контуре клиента.

Таким образом, данные клиента обрабатываются только внутри клиентского контура и не передаются за его пределы.

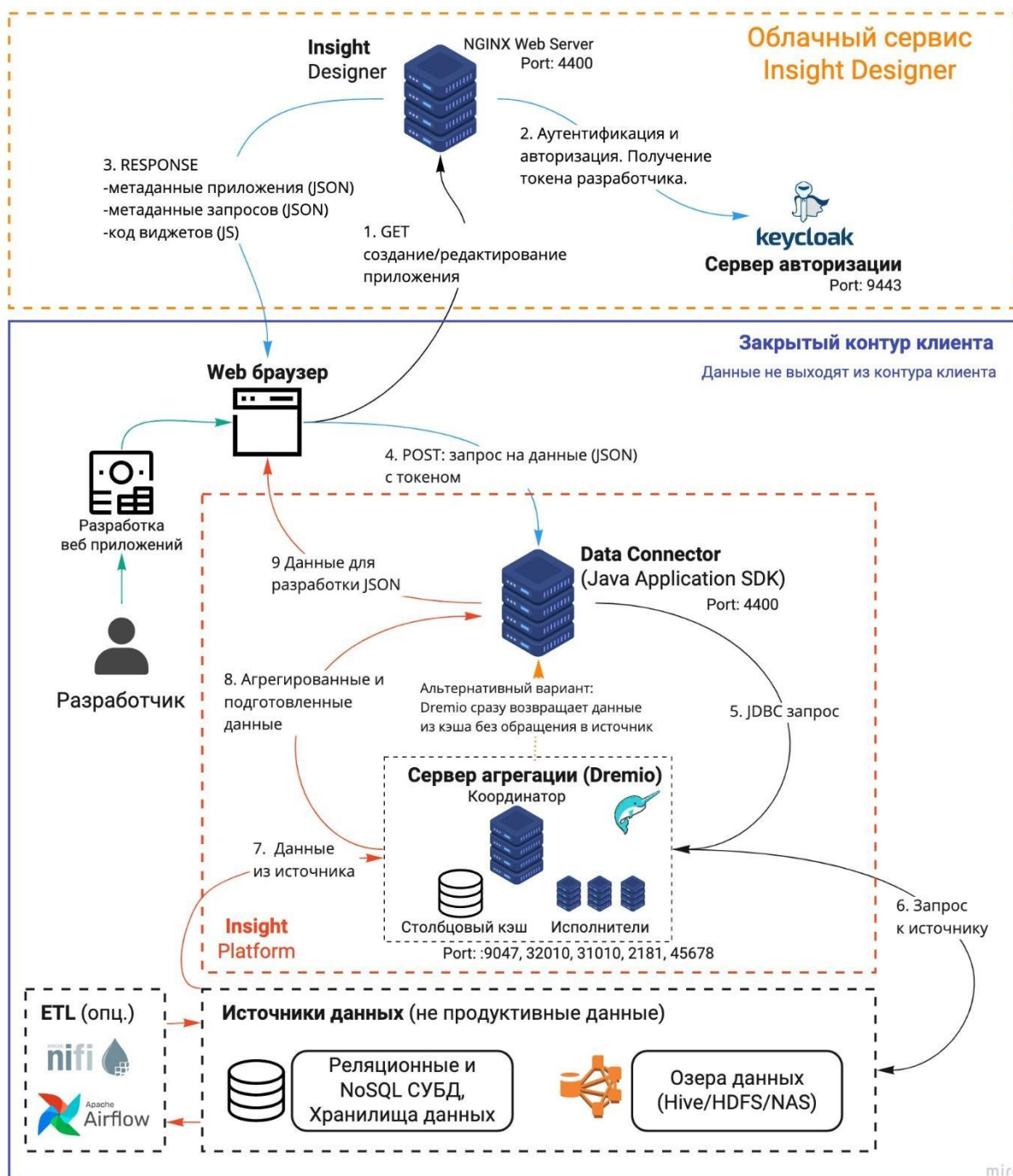
Схема процесса выглядит следующим образом:

1. Разработчик через Web браузер обращается к конструктору приложений посредством HTTPS запроса (GET).
2. Для получения доступа разработчик проходит аутентификацию путем ввода пары логин и пароль. Аутентификация и авторизация выполняются сервисом keusloak. После успешного входа в систему пользователь получает токен со всеми необходимыми атрибутами для дальнейшего доступа к компонентам платформы и данным.

3. Платформа возвращает пользователю метаданные аналитического приложения со всей конфигурацией, а также метаданные с описанием запроса к источникам данными, которые требуются для отчетности. Метаданные структурированы в формате JSON.
4. Клиентское приложение обращается к сервису Data Connector методом POST с указанием запроса к данным в формате JSON.
5. Data Connector, используя стандарт Java Database Connectivity (JDBC), открывает соединение с сервером агрегации данных Dremio и передает ему информацию для выборки данных.
6. Сервер сбора и агрегации данных может сразу вернуть данные из кэша (при их наличии) или сформировать последующие запросы непосредственно к источникам данных клиента, используя стандартные или разработанные коннекторы.
7. В результате обработки запроса, полученного от сервера сбора и агрегации, источник возвращает необходимые данные.
8. Сервер сбора и агрегации данных возвращает данные в Data Connector.
9. Data Connector в ответ на HTTP запрос в шаге 4 возвращает выборку данных в формате JSON для последующей визуализации в Web браузере. Стоит отметить, что Web браузер находится в закрытом периметре клиента и поэтому клиентские данные остаются в защищенном периметре и не уходят в открытый контур.

Подробная графическая схема процесса представлена на Рисунке 3.

Рисунок 3 Схема работы платформы в DEV



1.2. КОНЦЕПТУАЛЬНАЯ СХЕМА РАБОТЫ ПЛАТФОРМЫ В ТЕСТОВЫХ И ПРОДУКТИВНЫХ СРЕДАХ (TST, PRD)

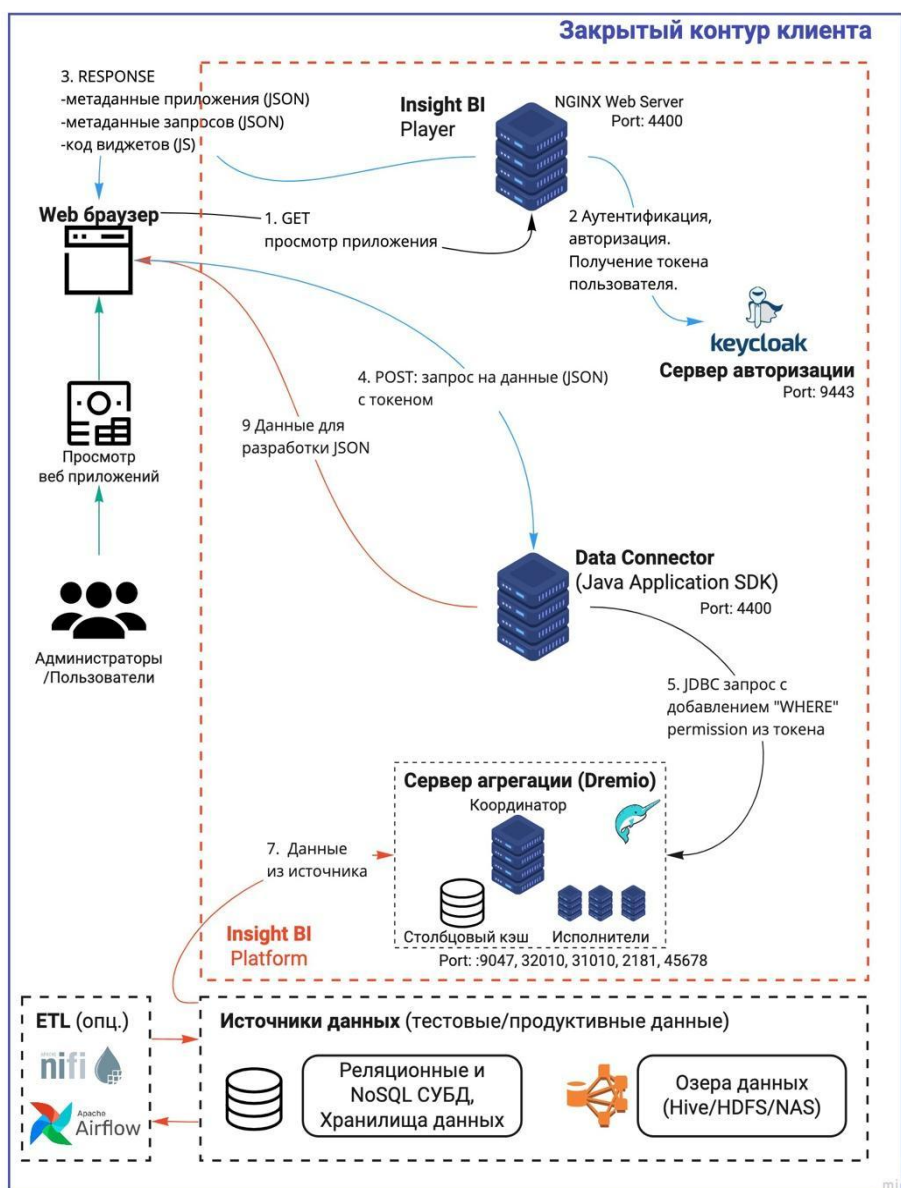
Принцип работы Платформы в тестовой и продуктивной средах (далее TST и PRD) незначительно отличается от среды DEV:

1. TST и PRD не использует облачный сервис для конструирования приложения (Designer).

2. Все сервисы Платформы Insight low code для TST и PRD размещаются в закрытом контуре клиента.
3. В средах TST и PRD работает механизм распределения доступа между пользователями на уровне данных: на шаге кроме фильтров, определенных разработчиком на уровне конструктора, также передается условный оператор WHERE, в котором фиксируются ограничения к данным с учетом роли, полученной разработчиком на этапе авторизации.

В остальном общий принцип извлечения данных из источника и их последующая обработка остается аналогичным тому, что применяется для DEV.

Рисунок 4 Схема работы платформы в TST/PRD



1.3. ПОДХОД К ТРАНСПОРТИРОВКЕ КОНФИГУРАЦИИ ПЛАТФОРМЫ МЕЖДУ DEV, TST, PRD

В данном разделе представлен принцип синхронизации конфигурации Платформы между средами DEV, TST и PRD.

Платформа также поддерживает другие варианты:

- двухзвенная конфигурация ландшафта (DEV+TST, PRD), если такой ландшафт был выбран клиентом;
- четырехзвенный вариант с prePRD для тестирования на продуктивных данных ограниченным числом пользователей.

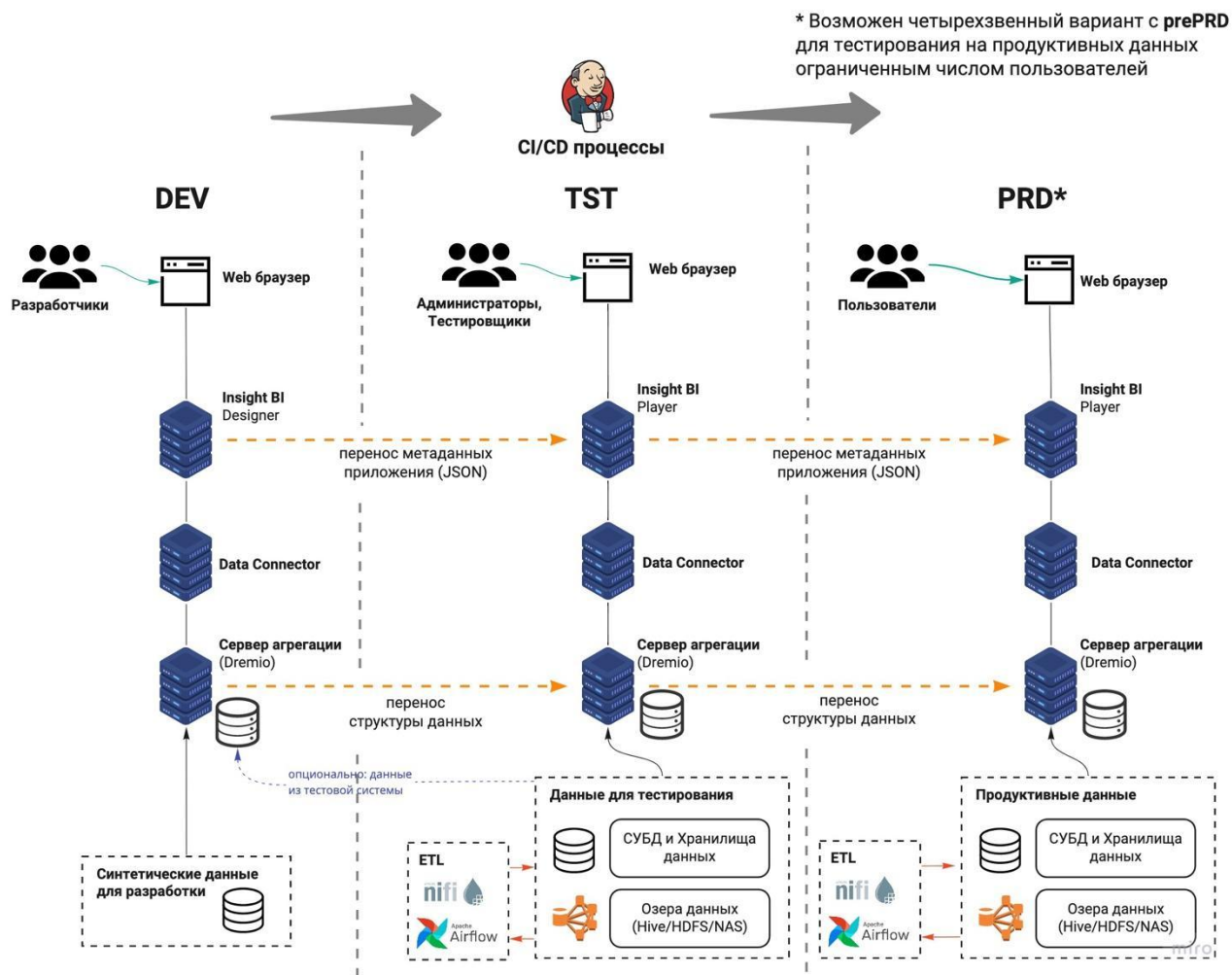
Транспортная система организуется с помощью автоматизации процессов непрерывной интеграции программного обеспечения (CI/CD) при использовании вспомогательных сервисов (например Jenkins).

Перенос конфигурации включает в себя передачу следующих типов объектов Платформы:

1. **Конфигурация приложений**, которая описывается в формате JSON и включает в себя: разметки страниц, настройки виджетов, доступы к данным, настройка запросов к данным, полномочия доступа к элементам приложения и другие компоненты, необходимые для работы аналитических приложений.
2. **Сопутствующие файлы**, используемые для приложений: изображения, файлы с описанием css стилей и прочее.
3. **Метаданные** модели данных, используемой в приложении. Перенос метаданных выполняется на уровне системы агрегации (Dremio).

Концептуальная схема синхронизации конфигурации Платформы между DEV, TST, PRD представлена на схеме (Рисунок 5).

Рисунок 5 Схема синхронизации конфигурации Платформы между DEV-TST-PRD



2. ПОДКЛЮЧЕНИЕ ДАННЫХ

Для подключения данных на Платформе есть 3 опции:

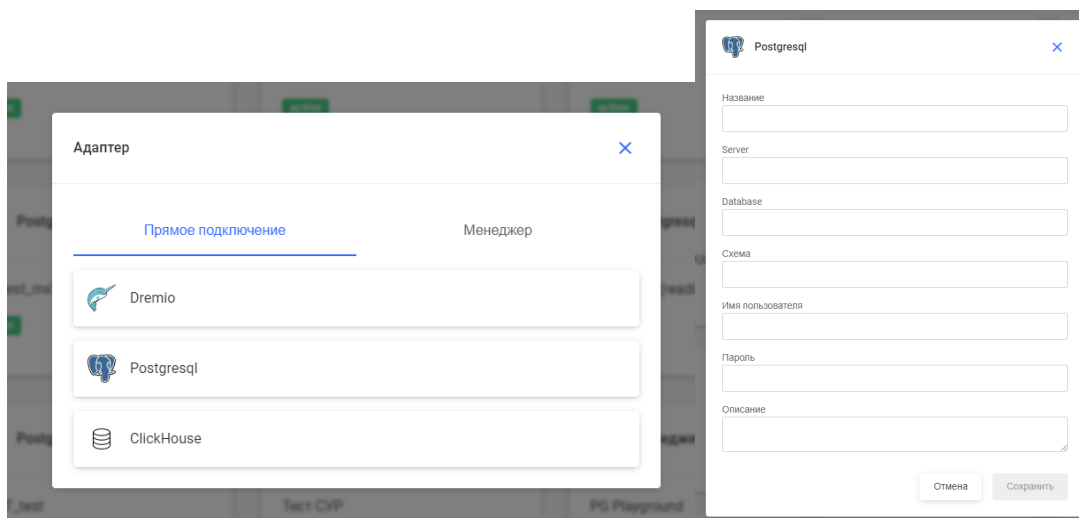
1. Подключение напрямую к БД;
2. Использование менеджера подключений;
3. Загрузка файлов формата excel, csv.

Прямые подключения предполагают использование коннектора Insight low code.

Подключение возможно к:

- PostgreSQL;
- Dremio;
- ClickHouse;
- Loginom.

Рисунок 6 Интерфейс настройки прямого подключения



Также существует возможность разработать собственный коннектор.

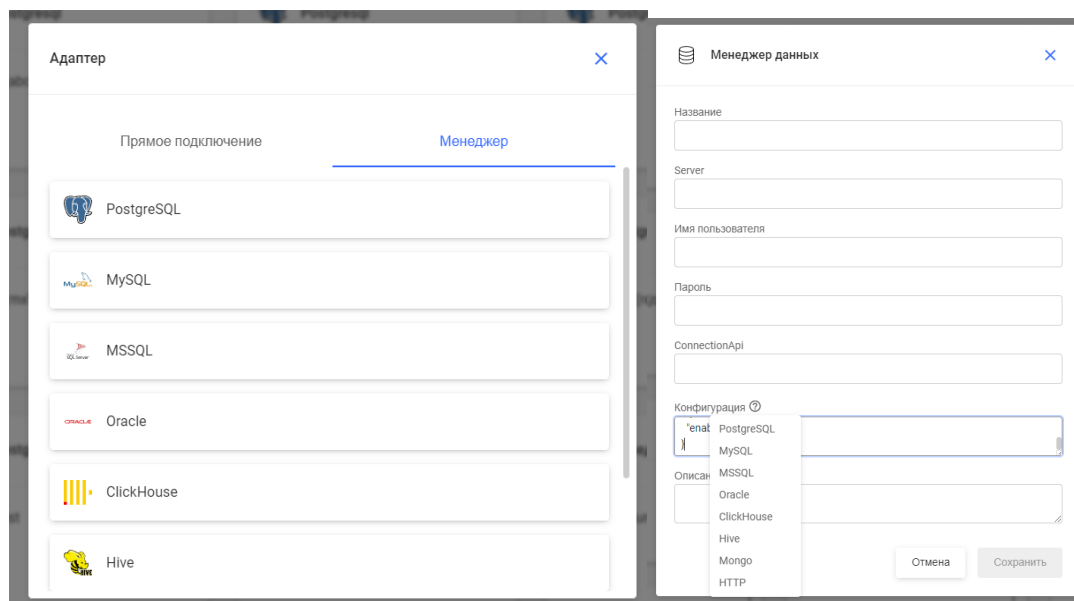
Менеджер подключений - это подключения, настроенные с помощью плагинов Apache drill.

Плагин Drill — это программный модуль для подключения Drill к источникам данных, таким как базы данных и локальные или распределенные файловые системы.

С помощью Менеджера подключений можно обратиться к:

- PostgreSQL
- MySQL
- MSSQL
- Oracle
- ClickHouse
- GreenPlum
- Arenadata
- Hive
- Mongo
- HTTP

Рисунок 7 Интерфейс настройки менеджера подключений



На Платформе существует возможность загрузки файлов формата `xlsx`, `xls`, `csv`:

1. Через интерфейс S3 хранилища (например, MiniO);
2. Через интерфейс Dremio;
3. Через интерфейс редактора Insight low code.

Ключевые особенности решения с точки зрения работы с данными:

- Высокая производительность – быстрые запросы благодаря встроенным in-memory алгоритмам;
- Слой самостоятельной работы с данными и семантики (self service);
- Объединение данных из различных источников «на лету»;
- Унифицированная работа с данными через стандарт SQL – простота освоения и мощные возможности;
- Возможность расширения коннекторов к любым источникам данных.

Платформа обеспечивает непосредственную работу с данными из различных источников – с озерами данных (data lake, реляционными базами данных (RDBMS), файлами и программными интерфейсами внешних приложений (REST API).

Современная архитектура решения позволяет избежать необходимости объединять все эти данные в сложные и дорогостоящие хранилища данных. Данные могут оставаться в исходном формате в существующих системах. При этом если в компании уже внедрены и используются хранилища данных (data warehouse) или реализованы решения на базе BI инструментов (таких как Qlik), такие источники также могут быть подключены к Платформе наряду с другими.

3. ПОДГОТОВКА ДАННЫХ

3.1. ОРКЕСТРАТОРЫ И ETL/ELT ИНСТРУМЕНТЫ

В поставку Платформы могут входить следующие ETL/ELT и оркестраторы подготовки данных инструмента:

1. Loginom;
2. Apache Ni-Fi;
3. Apache AirFlow.

3.2. SDK INSIGHT LOW CODE

SDK Insight low code обеспечивает семантический слой работы бизнес-аналитика с данными.

Он преобразует таблицы данных в понятные аналитику термины предметной области.

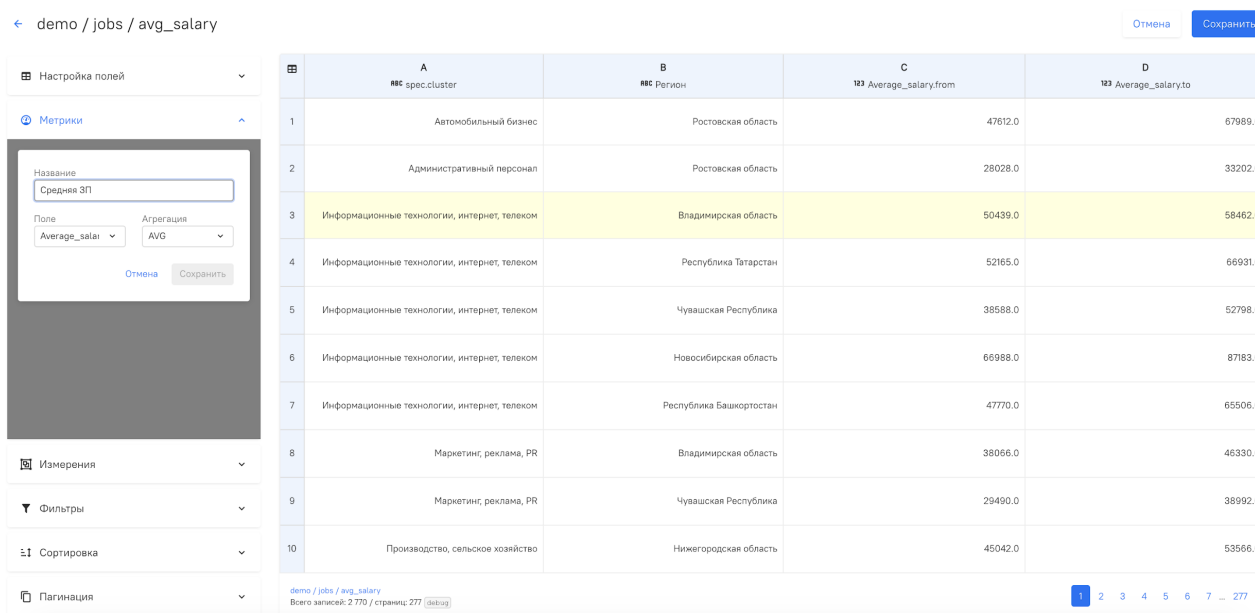
SDK (software development kit) обеспечивает автоматическую генерацию SQL запроса на основании:

- выбранного сохраненного запроса (виртуального датасета), выступающего в качестве источника данных;
- выбранных полей запроса;
- выбранных способов агрегации данных (сумма, среднее значение и др.);
- заранее определенного фильтра данных (по значению поля, по маске и др.);
- динамического фильтра данных (например, при выборе значения поля в диаграмме при «проваливании» на нижний уровень иерархии, либо при выборе параметра в связанном элементе или диаграмме);
- фиксированных или динамических условий сортировки;
- вычисляемого поля (произвольной формулы или запроса SQL);
- скользящих периодов.

Сохраненные датасеты доступны для использования в качестве источников данных для построения интерактивных динамических дашбордов и сложных веб-приложениях.

Параметры запроса описываются в виде конструкции SDK Insight low code ML (Modeling Language), представляющем собой особую конструкцию в формате JSON.

Рисунок 8 Интерфейс настройки динамического запроса в визуализации (SDK)



	A #RC spec.cluster	B #RC Регион	C #3 Average_salary.from	D #3 Average_salary.to
1	Автомобильный бизнес	Ростовская область	47612.0	67989.0
2	Административный персонал	Ростовская область	28028.0	33202.0
3	Информационные технологии, интернет, телеком	Владимирская область	50439.0	58462.0
4	Информационные технологии, интернет, телеком	Республика Татарстан	52165.0	66931.0
5	Информационные технологии, интернет, телеком	Чувашская Республика	38588.0	52798.0
6	Информационные технологии, интернет, телеком	Новосибирская область	66988.0	87183.0
7	Информационные технологии, интернет, телеком	Республика Башкортостан	47770.0	65506.0
8	Маркетинг; реклама, PR	Владимирская область	38066.0	46330.0
9	Маркетинг; реклама, PR	Чувашская Республика	29490.0	38992.0
10	Производство, сельское хозяйство	Нижегородская область	45042.0	53566.0

Таким образом реализуется бесшовный интерфейс между:

- исходными данными (физический датасет);
- последовательными преобразованиями данных на уровне подготовки данных (виртуальный датасет);
- динамическими запросами со стороны потребителя информации с уточнением критериев запроса – вычисления и агрегаты, фильтры, сортировка (динамические запросы из визуализации).

SDK Insight low code может также использоваться в качестве универсального средства обмена информацией между внешними приложениями и слоем данных (виртуальных запросов).

SDK Insight low code поддерживает следующие объекты:

- **Источник** – ссылка на сохраненный SQL запрос (виртуальный или физический датасет). Выбор источника данных определяет, какие данные и какие поля из таблицы-источника будут использоваться в визуализации. Платформа поддерживает возможность подключать или отключать поля таблицы запроса, переименовывать их для удобства использования;
- **Метрика** – вычисляемый показатель, используемый в визуализации. В качестве примера метрики могут отображаться в виде значений точек на линейной диаграмме по оси Y. Метрика имеет название и описание для отображения в диаграмме (например, «средняя выручка»);
- **Измерение** – определяет разрез данных и группировки, по которым отображаются и группируются данные (метрики) в визуализациях. Например, измерение «Дата» является разрезом данных для средней выручки. Измерения поддерживают иерархии (вложенные измерения) (например,

«Год/Квартал/Месяц» или «Регион/Город/Магазин»). Вложенные измерения настраиваются путем выбора полей источника с указанием порядка отображения. Вложенные измерения используются для реализации механизма детализации данных (drill-down).

- **Фильтр** – условие отбора значений метрики или измерения. Поддерживаются математические операнды (>, >=, <, <=, =, !=, between, not between), символные (like для выбора по маске), операторы множеств (in, not in). Для полей с типом «Дата» поддерживаются специальные **константы**, с помощью которых можно легко настраивать выбор периода отображения и использовать для расчета метрик нарастающим итогом за период:
 - #year-start# - начало года;
 - #year-end# - конец года;
 - #year-prev-start# - начало предыдущего года;
 - #year-prev-end# - конец предыдущего года;
 - #year-rolling# - скользящий год;
 - #quarter-start# - начало квартала;
 - #quarter-end# - конец квартала;
 - #quarter-py-start# - начало аналогичного квартала предыдущего года;
 - #quarter-py-end# -конец аналогичного квартала предыдущего года;
 - #quarter-prev-start# - начало предыдущего квартала;
 - #quarter-prev-end# - конец предыдущего квартала;
 - #month-start# - начало месяца;
 - #month-end# - конец месяца;
 - #month-rolling# - скользящий месяц;
 - #month-py-start# - начало аналогичного месяца предыдущего года;
 - #month-py-end# - конец аналогичного месяца предыдущего года;
 - #month-prev-start# - начало предыдущего месяца;
 - #month-prev-end# - конец предыдущего месяца;
 - #week-rolling# - скользящая неделя;
 - #week-start# - начало недели;
 - #week-end# - конец недели;
 - #week-py-start# - начало аналогичной недели предыдущего года;
 - #week-py-end# - конец аналогичной недели предыдущего года;
 - #week-prev-start# - начало предыдущей недели;
 - #week-prev-end# - конец предыдущей недели;
 - #day-start# - начало дня;
 - #day-end# - конец дня;
 - #today# - сегодня;
 - #yesterday# - вчера.

Константы позволяют массово изменять значения стилей/текстовых выносок и иных текстовых значений при использовании в виджете. На Платформе реализован механизм хранения и навигации по константам, а также импорт/экспорт.

- **Сортировка** – способ сортировки (по возрастанию/ по убыванию). Поддерживается множественная сортировка и управление очередностью;
- **Пагинация** - управляет количеством записей, возвращаемых запросом с возможностью дозагрузки по мере необходимости (по кнопке «Далее» или номеру страницы). Пагинация позволяет избежать запроса чрезмерно большого количества данных.

Платформа поддерживает следующие виды расчетов для метрик:

- Агрегации данных:
 - SUM – сумма;
 - AVG – среднее;
 - MAX – максимум;
 - MIN – минимум;
 - COUNT – количество записей;
 - GROUP_CONCAT – конкатенация значений;
 - GROUP_CONCAT_UNIQUE – конкатенация уникальных наборов значений;
 - VALUE - значение строки;
- Произвольное SQL выражение (expression) с поддержкой функций:
 - Математические функции - арифметические и тригонометрические функции, прочие математические функции;
 - Функции даты/ времени - расчет интервалов времени, округление дат, текущее время, часовой пояс и др.;
 - Функции форматирования даты/времени;
 - Функции преобразования типов данных;
 - Текстовые функции - длина, преобразование текста, поиск, замена, вычисление регулярных выражений и др.;
 - Агрегирующие функции - стандартные агрегирующие функции, статистические функции;
 - Условные функции – «case ... when ... then» и др.;
 - Оконные функции;
 - Функции работы с вложенными структурами.

Полный перечень SQL функций приводится в технической документации на Платформу. Платформа поддерживает расширение перечня функций с помощью пользовательских функций, написанных на Java.

Платформа поддерживает разнообразные типы данных, автоматически определяя при загрузке данных, либо с помощью функций преобразования типов данных: INTEGER, FLOAT, DOUBLE, DECIMAL, BOOLEAN, VARCHAR, TIMESTAMP, DATE, TIME, STRUCT, LIST.

3.3. СОЗДАНИЕ ДАТАСЕТОВ В РЕДАКТОРЕ

Платформа поддерживает технологию массово-параллельного чтения данных из источников, применяя механизм динамического определения оптимального плана

выполнения запросов и технологию кэширования часто используемых данных из озер данных в формате колоночной структуры БД, оптимизированной для работы с оперативной памятью и SSD.

Механизм аналитических вычислений, основанный на открытом фреймворке Apache Arrow для колоночных in-memory структур данных, оптимизирован для векторных вычислений для эффективного использования современных CPU. Платформа поддерживает эластичное масштабирования и распределение нагрузки между нодами кластера.

Платформа поддерживает создание виртуальных SQL запросов, выстраиваемых в цепочки, позволяя пользователям легко проектировать архитектуру данных для аналитики с помощью визуальных механизмов (объединения таблиц, сортировки, фильтры, формулы) или с помощью стандартного синтаксиса языка запросов SQL. Это позволяет снизить требования к специалистам по подготовке данных и обеспечить быстрый старт работы с Платформой.

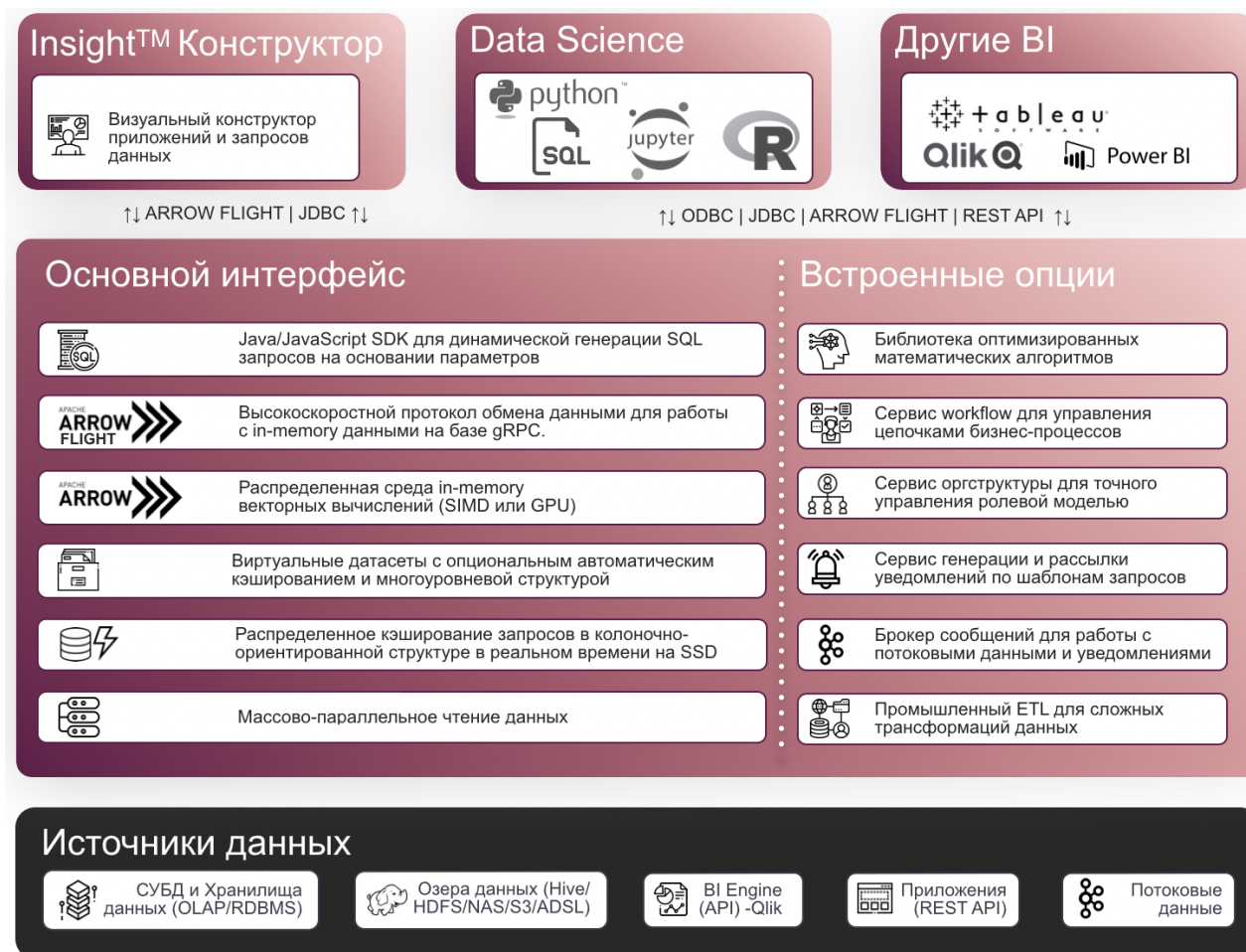
Оptionальный механизм принудительного кэширования данных запросов и агрегированных предрасчитанных данных в in-memory кэше позволяет снизить нагрузку на источники данных, обеспечивая при этом огромную производительность.

Интегрированный в Insight low code с помощью собственных разработок открытый фреймворк Apache Arrow, не зависящий от языка программирования вместе с эффективным бинарным колоночно-ориентированным форматом хранения данных Apache Parquet становятся в настоящее время наиболее популярным стандартом работы с большими данными и вычислениями в памяти (in-memory). Колоночный формат хранения также обеспечивает многократное сжатие данных по сравнению с традиционным форматом. Сохраненный таким образом кэш данных мгновенно загружается в память в случае обращения без необходимости декодировать или разархивировать сжатые данные.

С появлением применяемых новых технологий отпадает необходимость в построении кубов данных (OLAP), требующих больших ресурсов и времени на обновление, реконфигурацию структур кубов и специализированных средств и языков работы с такого рода данными. Работа с источниками данных может производиться в едином стандарте SQL запросов с беспрецедентной простотой и скоростью внесения изменений.

Платформа поддерживает автоматическое обновление кэшированных данных по расписанию и механизм определения «дельты» (загрузки только изменившихся данных). Планировщик-оптимизатор выполнения SQL запросов автоматически определяет наилучший вариант по дереву запросов с точки зрения наилучшей производительности.

Рисунок 9 Ключевые возможности Платформы в части работы с данными



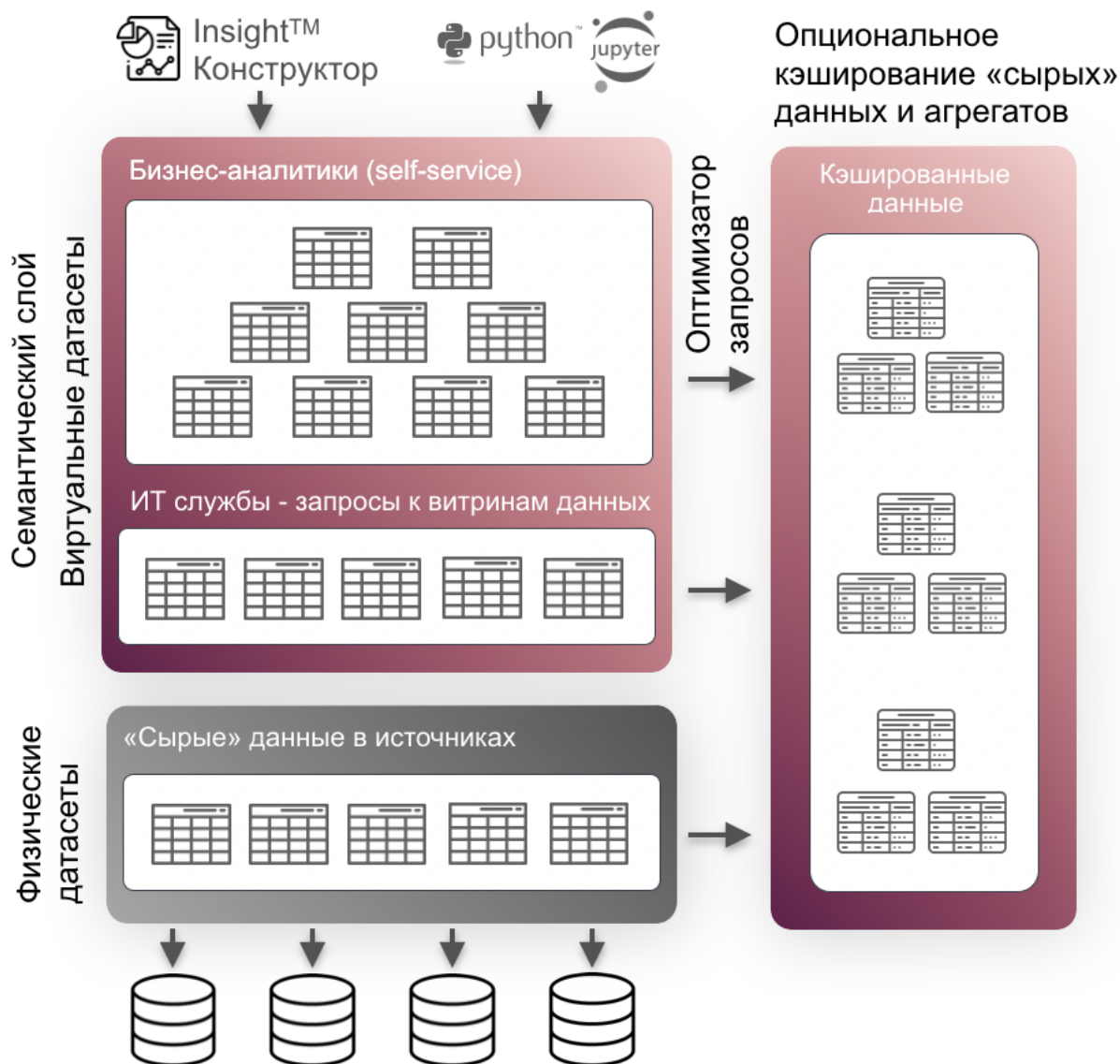
Наряду с технологическими решениями в области извлечения, обработки и хранения данных, Платформа также обеспечивает семантический слой для удобной работы с данными на уровне аналитиков и ИТ специалистов. Платформа содержит средства ведения метаданных, каталогизации запросов с поддержкой вики-формата описания, каталогов метрик и измерений, поддерживает механизмы тегирования и встроенный поиск.

Логика работы с данными предполагает наличие физических наборов данных (*датасетов*) и виртуальных наборов (не предполагающих создания физической копии данных). Виртуальные датасеты могут рассматриваться наряду с физическими в SQL запросах в операциях объединения (*join*), фильтрации, расчетах. Таким образом могут выстраиваться длинные цепочки запросов, упрощающих написание и поддержку сложных преобразований без влияния на объемы хранения и производительность выполнения запросов.

Использование виртуальных датасетов позволяет решить важную задачу в части разделения зон ответственности в части работы с данными, разделив уровень управления физическими датасетами и коннекторов (уровень ИТ поддержки) и работы аналитиков и специалистов data science (уровень самостоятельной «self-service» работы с данным), не нарушая целостность и синхронизированность

данных. Фактически виртуальные датасеты вместе с метаданными на уровне датасетов и полей, встроенной вики-справкой образуют семантический слой для аналитика по работе с данными.

Рисунок 10 Семантический и физический слой работы с данными



4. ИССЛЕДОВАНИЕ ДАННЫХ

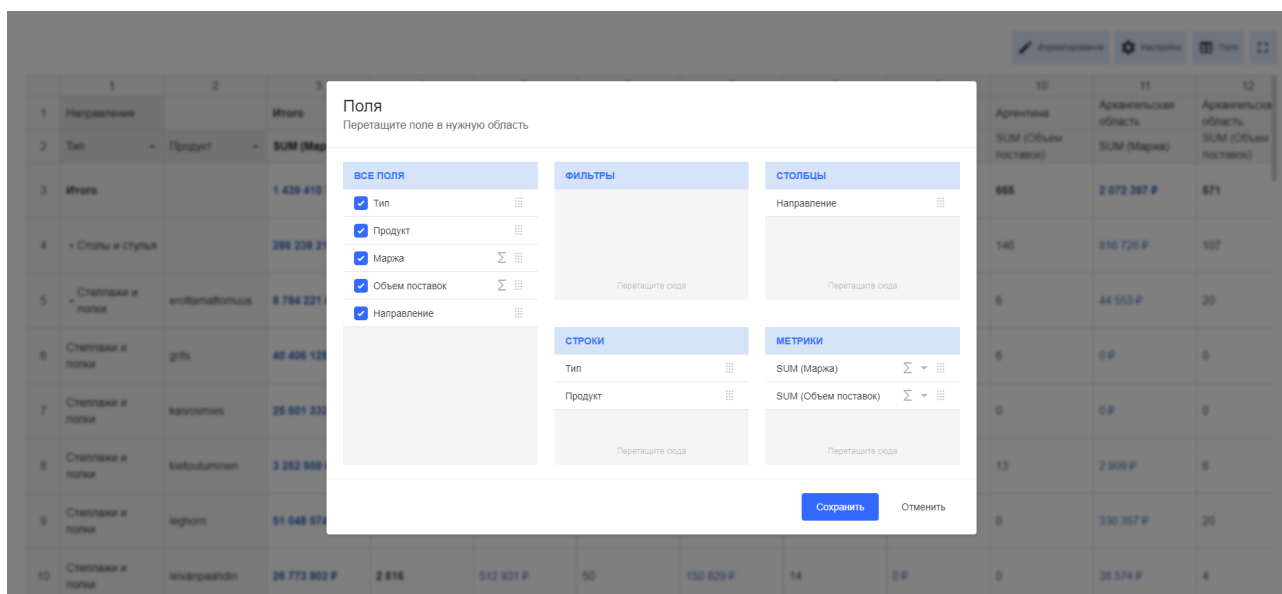
4.1. СВОДНАЯ АНАЛИТИКА

Платформа обеспечивает создание приложений, в которых специалисты могут самостоятельно выполнять запросы к нужным данным и генерировать необходимые отчеты без привлечения дата-специалистов.

Сводная таблица - Многофункциональный инструмент для анализа данных. Работа с таблицей предполагает активные действия пользователя в Плеере и позволяет:

1. "На лету" менять набор отображаемых в Таблице Метрик и Измерений;
2. Выводить Измерения в формате строк и столбцов;
3. Изменять тип агрегации Метрик;
4. Анализировать общие и промежуточные итоги по срезам данных;
5. Применять фильтрацию к выводимым значениям;
6. Управлять форматом отображаемых Метрик;
7. Сортировать значения в порядке увеличения и убывания;
8. Управлять иерархией измерений с помощью режима «Дриллдаун»;
9. Выводить нумерацию строк;
10. Работать в полноэкранный режиме;
11. Выбирать Тип отображения — Табличный, Компактный или Структуру;
12. Добавлять расчетные метрики;
13. Управлять индикацией значений по условию.

Рисунок 11 Интерфейс настройки Сводной таблицы



4.2. ВЫГРУЗКА ВИТРИН ДАННЫХ

Платформа обеспечивает возможность скачивания среза данных из витрины в формате **xlsx**, **csv** с учетом примененных фильтров, сортировки и пагинации для дальнейшего проведения анализа Пользователем.

5. КОНСТРУКТОР ПРИЛОЖЕНИЙ

Платформа поддерживает возможность визуальной настройки приложений с помощью встроенного конструктора веб-страниц и диаграмм (визуализаций) из набора виджетов, далее Конструктор или Insight low code Designer.

Встроенный Конструктор содержит все необходимые для верстки веб-страниц компоненты, поддерживает тонкую настройку каждого компонента с точки зрения требований верстки в соответствии со стандартами HTML. Платформа полностью соответствует последним стандартам – HTML5, CSS3.

Конструктор построен по принципу WYSIWYG. С его помощью пользователь самостоятельно располагает все элементы на странице, указывая положение на странице и относительно других элементов, настраивает бизнес-логику, визуализации и анимацию. Платформа автоматически преобразует созданное визуальное представление в HTML код.

Конструктор поддерживает блочную верстку и работу со слоями.

Создаваемые приложения поддерживают автоматическое масштабирование (scaling) для различных разрешений экранов как для размеров страниц, контейнеров и элементов, так и для типографики (размера шрифтов).

Платформа полностью поддерживает работу с мобильными устройствами путем создания приложений, предназначенных для работы на небольших экранах. При создании такого рода приложений необходимо учитывать специфику UX для небольших экранов – компоновку элементов, логику навигации.

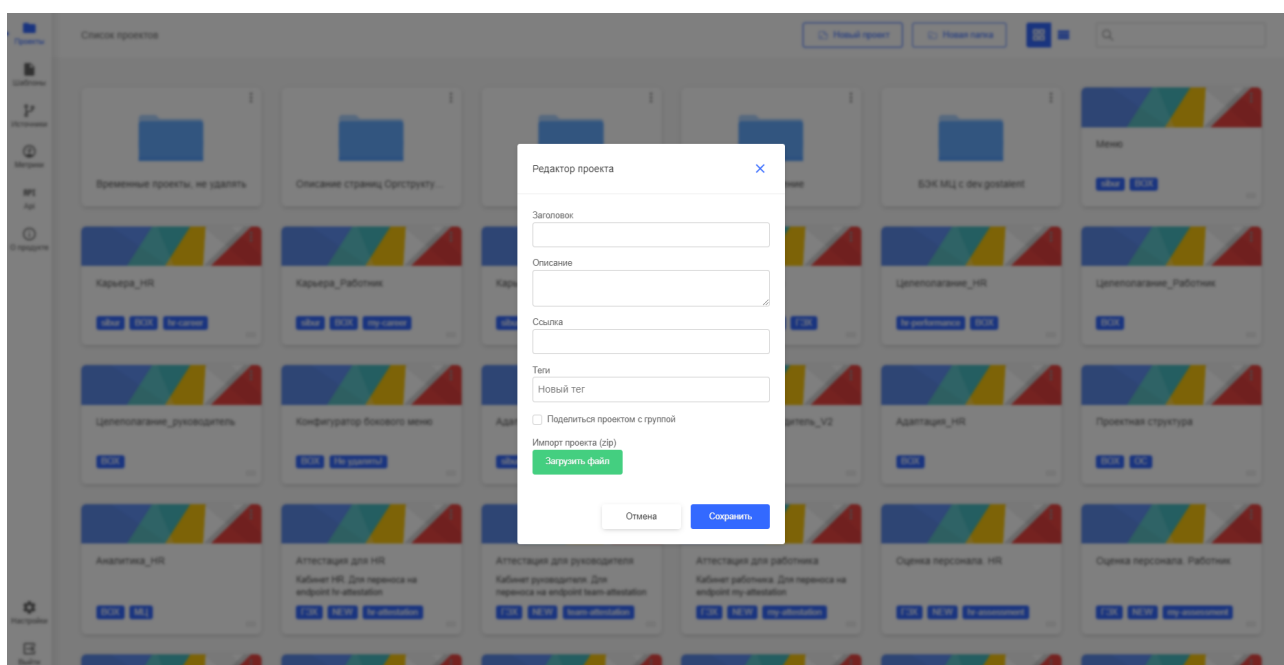
Платформа содержит встроенный CSS-фреймворк – библиотеку стилей, позволяющей централизованно управлять стилем отображения компонентов (цветами, градиентами, отступами, анимацией и др.) и типографикой (шрифтами, абзацами и др.). Фреймворк содержит все необходимые веб-компоненты, позволяет сильно упростить создание страницы. С помощью фреймворка можно централизованно управлять корпоративным стилем, различными константами (например, задать константу цвета «#success» («успех»), присвоив оттенок зеленого цвета в соответствии с брендбуком).

Платформа поддерживает передачу параметров между различными объектами страницы (визуализациями) с помощью **СОБЫТИЙНОЙ ШИНЫ**. Наличие front-end шины позволяет организовать создание реактивных приложений в соответствии с современными стандартами разработки веб-приложений. С помощью событийной шины реализуется возможность drill-down, обновления визуализаций по событию или посредством передачи параметра. Простыми примерами использования является обновление диаграмм после выбора даты в веб-компоненте «Календарь», либо смена визуализации на альтернативную после нажатия кнопки. Событийная шина работает полностью на стороне веб-приложения без участия и вызова back-end сервисов. В случае необходимости запроса к back-end веб-

виджеты самостоятельно отправляют соответствующие запросы к back-end сервисам.

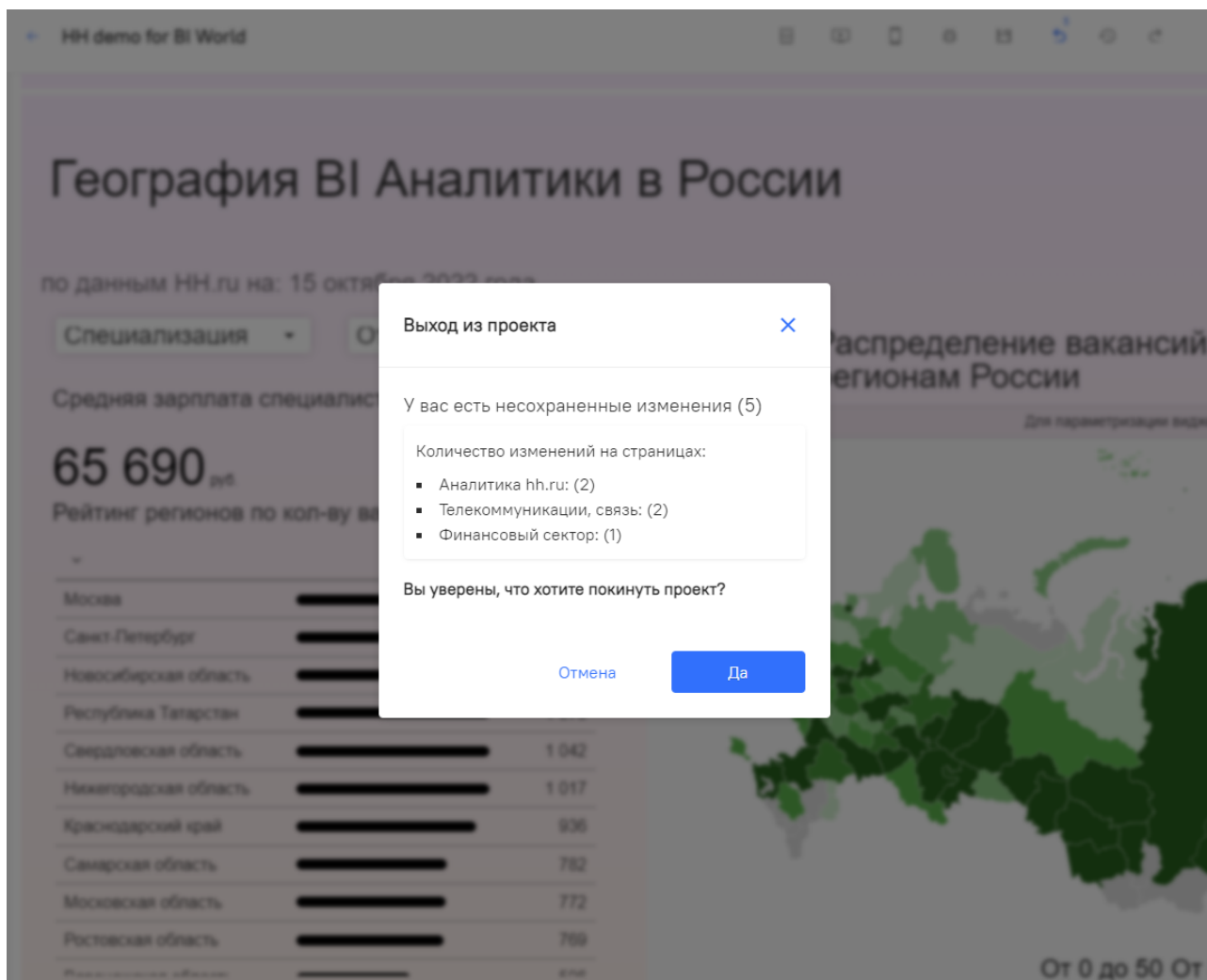
Основной элемент Платформы - **приложение** – объект, описывающий набор страниц, визуализаций и бизнес-логики, включая подключения к источникам данных. Приложения представляют собой законченную логическую группировку **страниц** и бизнес-логики. Пользователь самостоятельно принимает решение о выделении отдельных приложений. В дальнейшем доступ к каждому приложению обеспечивается с помощью уникальной веб-ссылки. С помощью полученных ссылок можно организовать на портале вызов различных приложений. Приложения имеют метаданные, описывающие их назначение - наименование, теги, картинку. Также, доступна опция клонирования приложений - новое приложение будет полной копией со всеми страницами и вложенными файлами.

Рисунок 12 Интерфейс внесения изменений



При выходе из приложения пользователь получит предупреждение с количеством несохраненных изменений на каждой странице.

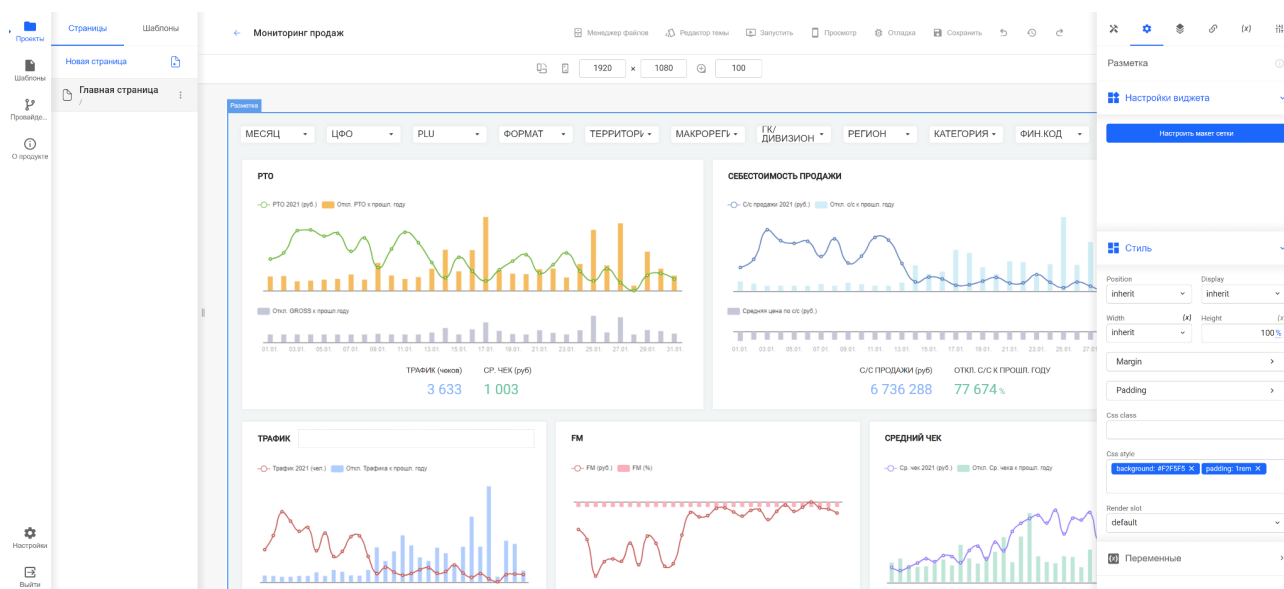
Рисунок 13 Предупреждение о несохраненных изменениях



Страница представляет собой веб-страницу с размерами, возможностью прокрутки и иными настройками стиля. Приложение может включать произвольное количество страниц, логика навигации между которыми может осуществляться с помощью настройки компонентов меню и элементов страницы. На уровне Конструктора задается стартовая страница приложения (root, “/”), а также задаются символьные ссылки для каждой страницы. Платформа поддерживает отображение страниц:

- в виде страницы фиксированного размера без возможности прокрутки;
- в виде страницы с вертикальной и/или горизонтальной прокруткой;
- в виде всплывающего окна.

Рисунок 14 Пример настройки страницы приложения

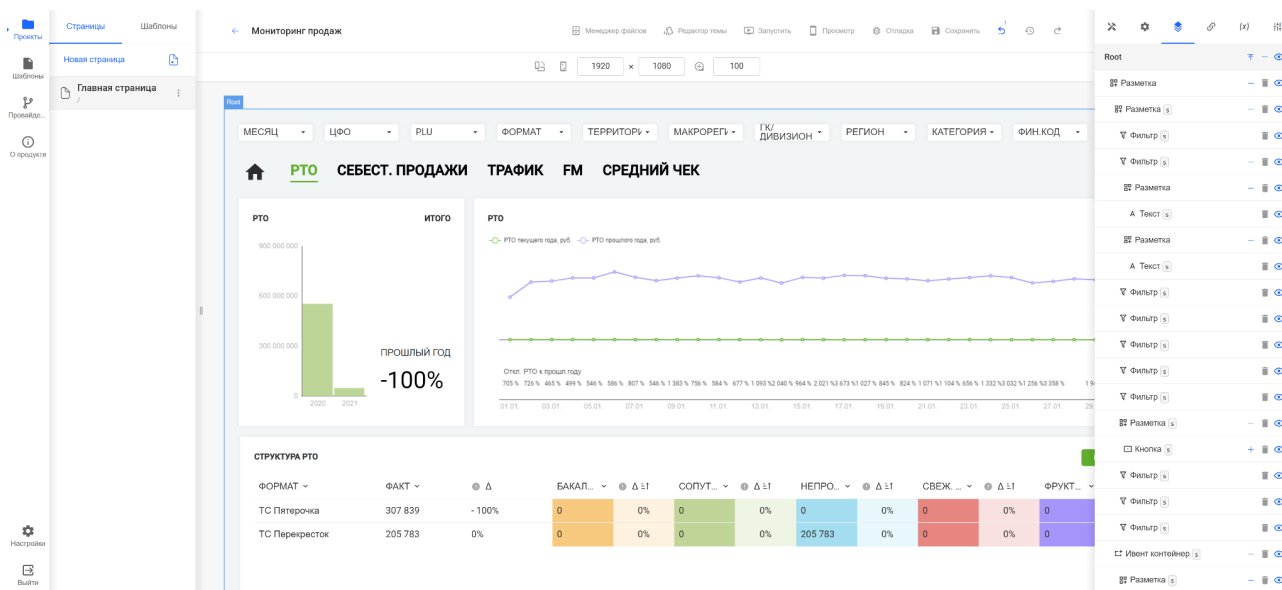


Платформа поддерживает гибкую настройку **разметки**. Можно настроить фиксированную сетку в абсолютных значениях, либо гибкую сетку в виде произвольной комбинации колонок и строк с динамически вычисляемым размером, заданным в процентах от размера страницы.

Платформа поддерживает слои и управление наложением элементов друг на друга (z-index). В частности, виджет Контейнер иерархии позволяет без использования css стилей и классов управлять иерархией отображаемых виджетов и накладывать их друг на друга.

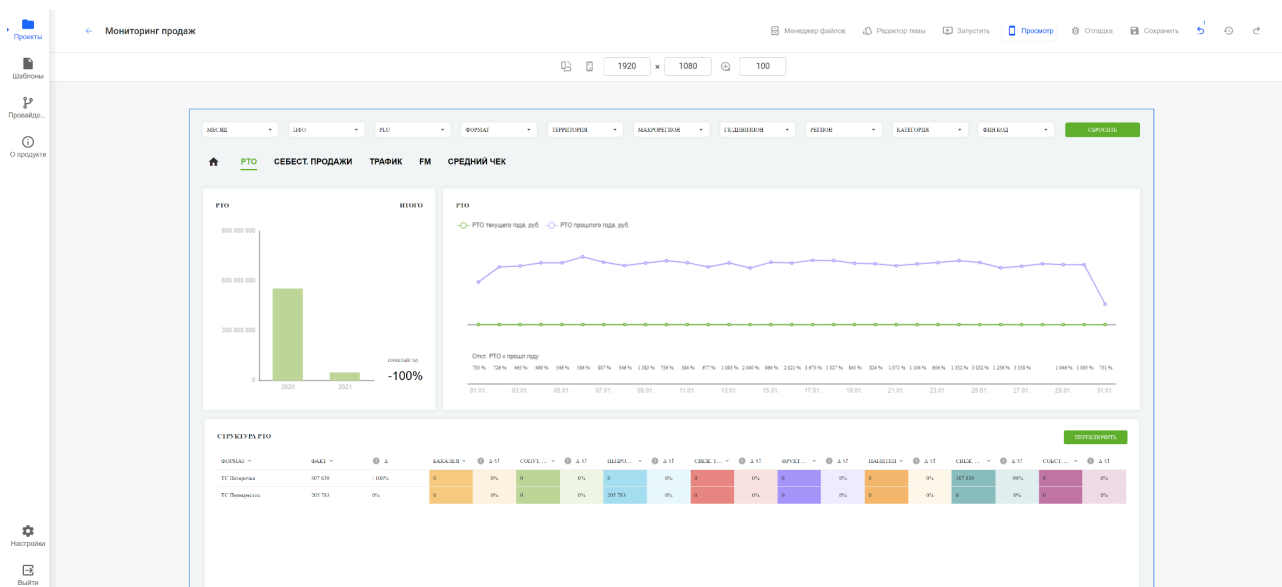
Элементы страницы выстраиваются в виде дерева, эквивалентного структуре дерева DOM элементов. Конструктор поддерживает возможность навигации как с помощью выбора элемента с помощью клика, так и с помощью навигации по дереву элементов.

Рисунок 15 Пример дерева элементов страницы



Конструктор содержит средства удобного отображения/скрытия технических элементов (режим отладки, режим просмотра). В режиме отладки выделяются все скрытые элементы.

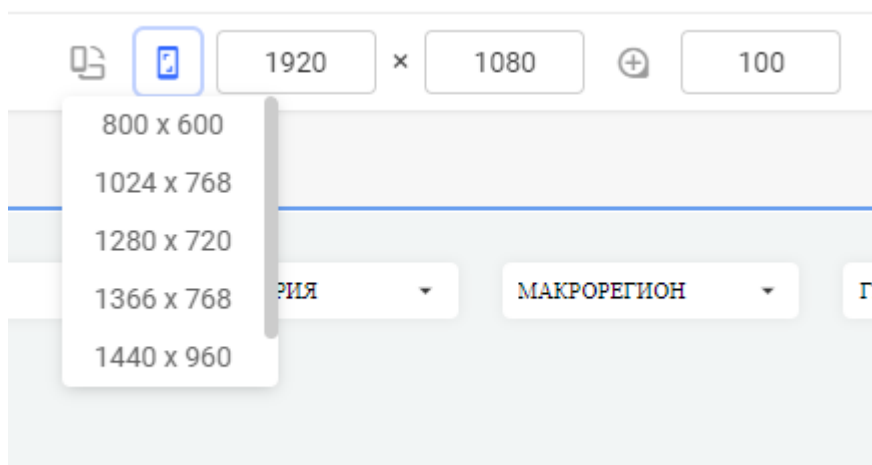
Рисунок 16 Пример отображения страницы в режиме отладки



В режиме просмотра можно предварительно посмотреть на вид страницы в браузере.

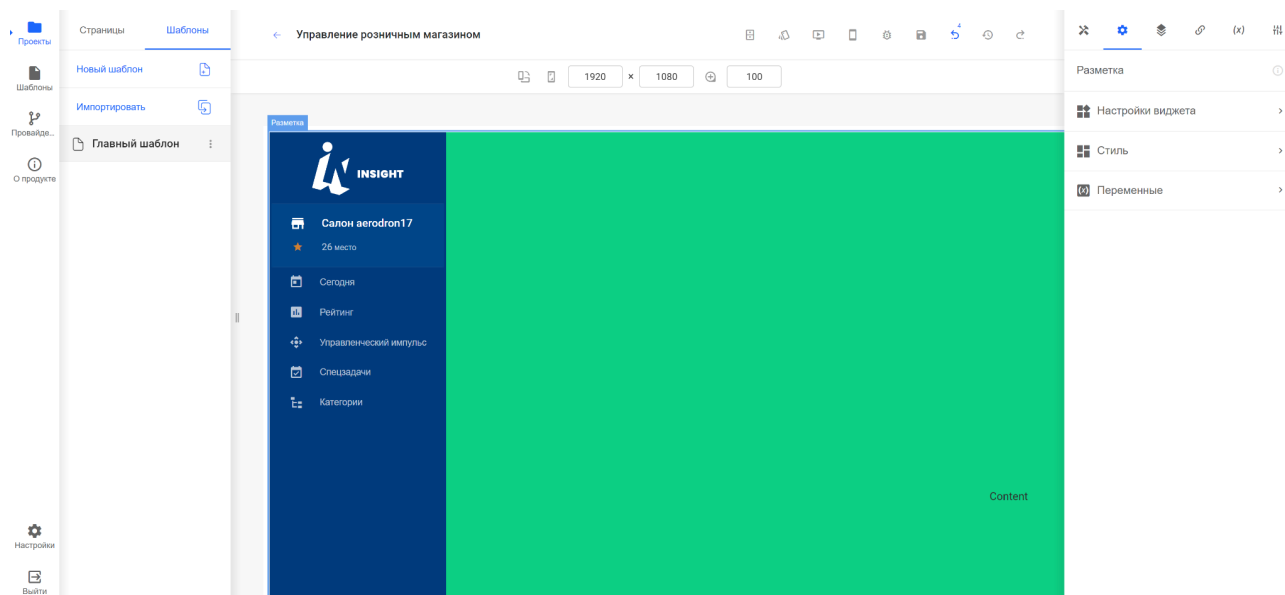
Для использования доступны предустановленные размеры устройств, а также ручной ввод параметров.

Рисунок 17 Режим Просмотр



Страницы могут ссылаться на **шаблоны**. С помощью шаблона можно создавать структуру страницы – компоновку элементов навигации и меню. Например, в качестве шаблона страницы можно реализовать горизонтальную строку меню вверху страницы или расположить меню слева в виде выезжающей панели. Шаблоны также могут использоваться для управления брендбуком – например, цвета заливки фона или логотипы. Шаблоны содержат специальный элемент «Контент шаблона», в который будет размещаться настраиваемый контент страницы. Внесение изменений в шаблон автоматически изменит все зависящие от него страницы, что облегчает управление изменениями.

Рисунок 18 Пример настройки шаблона страницы с меню в левой части

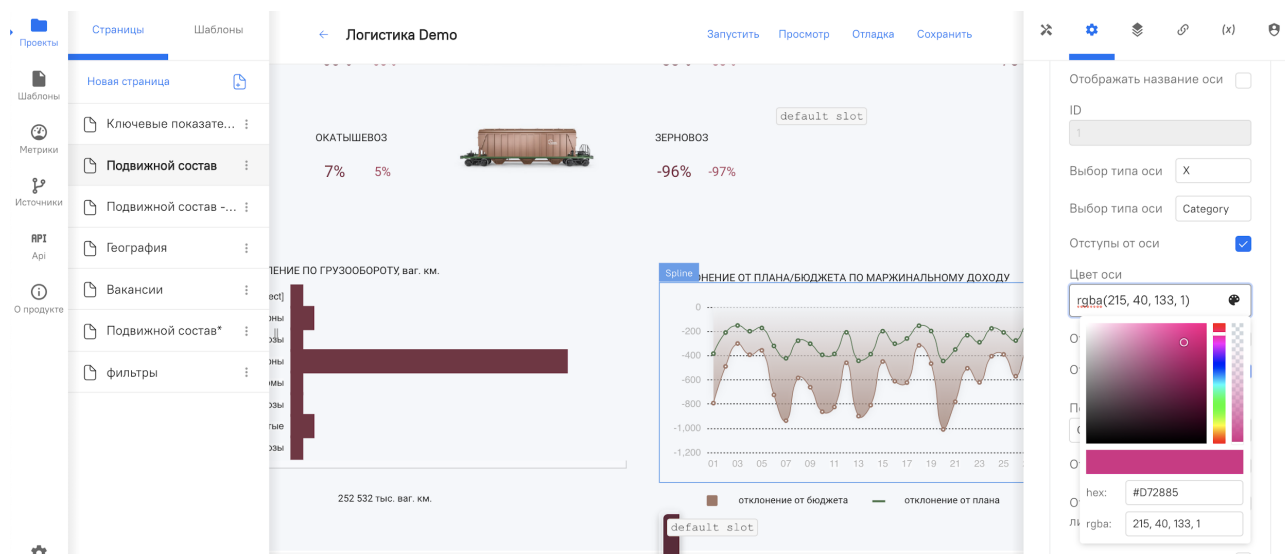


Ключевым элементом Конструктора Платформы являются **виджеты**. Они представляют собой веб-компоненты, визуализации (диаграммы) или законченные мини-приложения. Примеры веб-компонентов – кнопка, картинка, календарь. В качестве визуализаций могут быть таблицы, линейные и столбчатые диаграммы, карты и т.п. Отдельным примером виджетов могут являться мини-приложения,

выполняющие комплекс задач как по отображению, так и вводу и сохранению данных со встроенной бизнес-логикой.

Конструктор позволяет распределить виджеты по странице с помощью технологии drag-n-drop. На уровне каждого виджета настраиваются свойства виджетов, такие как стиль, параметры шины, которые слушает или изменяет виджет, специфичные для виджета настройки.

Рисунок 19 Пример настройки виджета



Конструктор содержит также ряд специализированных виджетов для управления аутентификацией и ролевым доступом. С помощью контейнера авторизации (“auth container”) и контейнера ролей (“permission container”) возможна настройка авторизации и видимости отдельных виджетов в зависимости от доступных пользователю полномочий. С помощью настройки можно реализовать сценарий, при котором пользователи разного уровня доступа будут видеть одно и то же приложение по-разному. Например, территориальный руководитель магазинов будет видеть все нижестоящие магазины и навигацию по ним и сводные таблицы и диаграммы по магазинам, а менеджер магазина будет видеть только свой магазин и отдельный набор диаграмм с упрощенной навигацией.

Специализированные виджеты («event container» и «event stack») позволяют в зависимости от событий в шине управлять отображением отдельных элементов. Например, по нажатию кнопки убрать с экрана таблицу и заменить ее на две круговых диаграммы. В качестве сценария использования может рассматриваться вариант расширенного drill-down, когда при достижении нижнего уровня иерархии измерения вместо отображения единственного значения диаграмма заменяется на другую, показывающую необходимые детали.

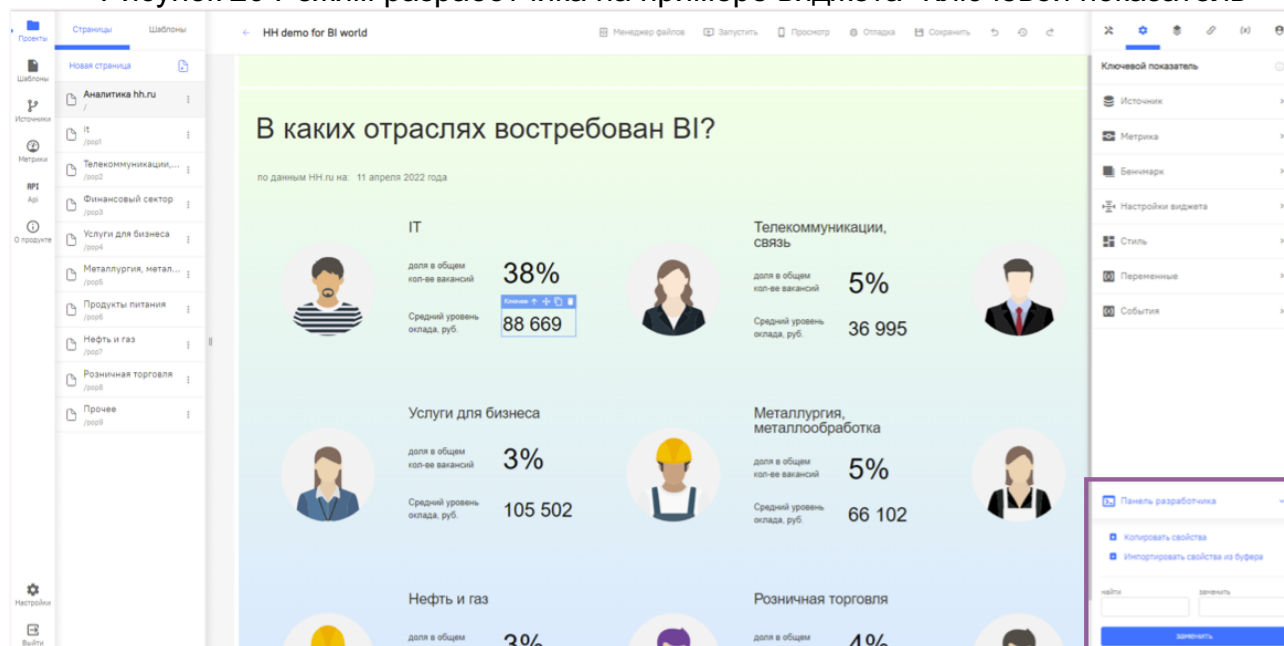
Оба описанных выше сценария позволяет решить виджет конструктор бизнес-логики (“smart gallery”), который регулирует отображение контента в зависимости от событий в шине и настроенной ролевой модели.

Режим разработчика содержит в себе расширенный функционал для продвинутых пользователей (прошли дополнительное обучение).

На текущий момент функционал состоит из:

- Модуля, позволяющего копировать полную настройку элемента системы (подключение к данным и стилистические настройки) с дальнейшим импортом;
- Модуля, позволяющего производить замену json страницы.

Рисунок 20 Режим разработчика на примере виджета “Ключевой показатель”



5.1. ПАРАМЕТРЫ ВИДЖЕТОВ

Виджеты поддерживают следующий обязательный набор настраиваемых параметров:

- Параметры стиля в соответствии со стандартом CSS3:
 - Позиция (position) со значениями inherit, static, absolute, relative, fixed, sticky;
 - Отображение (display) со значениями inherit, block, inline, inline-block, flex, inline-flex;
 - Ширина (width) и высота (height) со значениями в процентах, абсолютных и относительных величинах - size, %, px, em, rem, vw;
 - Внешний отступ (margin) с параметрами top, bottom, left, right;
 - Внутренний отступ (padding) с параметрами top, bottom, left, right;
 - CSS-класс (CSS class) для присвоения виджету класса, описанного в CSS фреймворке;
 - CSS стиль (CSS style) для произвольной настройки стиля с помощью стандарта CSS.
- Параметры рендера (отображения):

- Выбор **слота** для отображения;
- Переименование компонента в слоях редактора

Слот позволяет определить в какую область будет выводиться виджет – для управления слоями (z-index) или скрытыми областями, выводимыми в случае наступления события.

В зависимости от типа виджета для них доступны следующие наборы параметров:

- Параметры виджета – специфичные для каждого виджета наборы параметров, которые также могут быть сгруппированы по смысловым категориям. Например, параметры отображения осей графика (цвет оси, шрифт, цвет надписей на оси, скрыть ось и др.), параметры отображения точек данных (цвет, шрифт).
- Источник данных (для виджетов-диаграмм) – настроенная с помощью SDK Insight low code комбинация источника данных, метрик, измерений, фильтров, сортировки и параметров пагинации.
- Переменные:
 - Переменные, которые виджет получает («слушает») из шины;
 - Переменные, которые виджет отправляет в шину.

5.2. КОМПОЗИЦИОННЫЕ ВИДЖЕТЫ

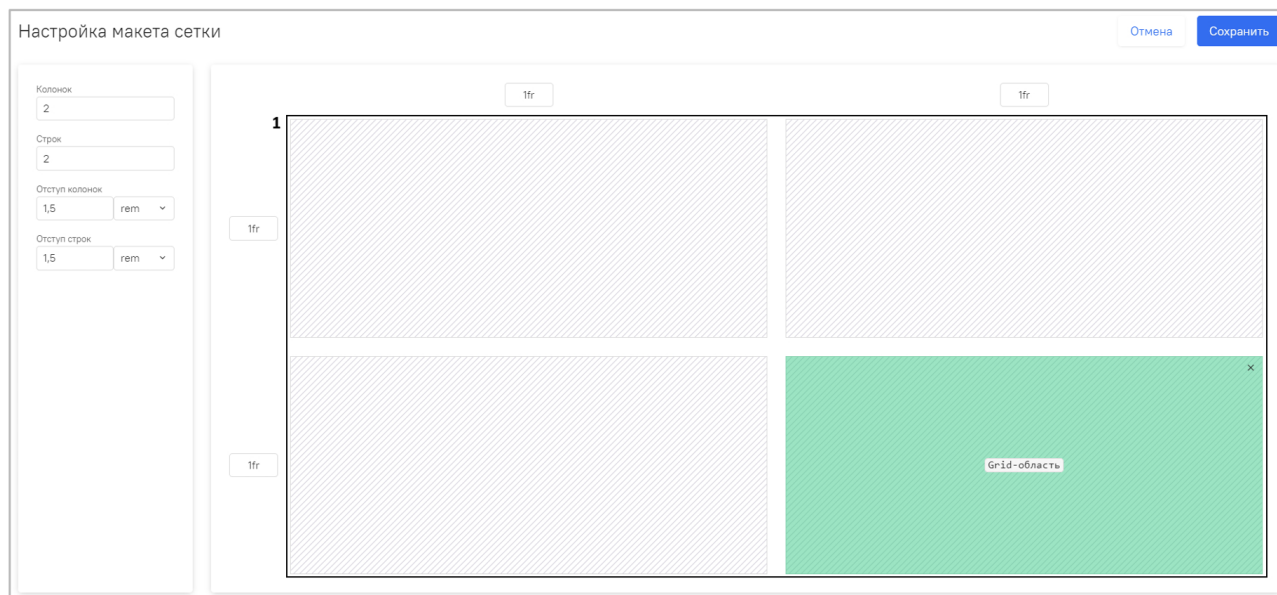
Платформа поддерживает гибкую верстку страницы с помощью следующих компонентов:

- Разметка;
- Ивент контейнер;
- Контейнер;
- Тайл – прямоугольная плитка;
- Авторизационный контейнер;
- Разграничение контента;
- Контейнер иерархии;
- Конструктор бизнес-логики.

С помощью указанных элементов можно создать любую конфигурацию элементов на странице, включая вложенные элементы.

Разметка (grid layout) – виджет, который позволяет создать двумерную сетку для формирования структуры страницы и размещения элементов.

Рисунок 21 Пример настройки макета сетки виджета “Разметка”



Виджет представляет собой контейнер, содержащий строки и колонки(полосы), количество которых задается в одноименных полях меню настройки слева.

Между полосами можно задать отступ. Строки и колонки формируют ячейки сетки, для использования которых необходимо задать Grid-область. Виджет, расположенный в Grid-области, является Grid-элементом. Особенностью работы с виджетом “Разметка” является возможность размещения одной Разметки в другой для создания страниц с большим количеством элементов.

Существует пул виджетов, позволяющих управлять контентом страницы и настройкой авторизации.

Ивент контейнер (event stack) – позволяет управлять содержимым в зависимости от состояния в шине событий. Например, необходимо в зависимости от значения кнопки выводить разные диаграммы. В этом случае для каждой кнопки устанавливается свое событие, которое она отправляет в шину событий. Компонент «ивент контейнер» определяет все возможные состояния (слоты) для входящих в него элементов, а также состояние по умолчанию. Для каждого из набора виджетов устанавливается свой слот отображения, который в свою очередь привязан к состоянию кнопки.

Контейнер авторизации (auth container) – позволяет настроить параметры авторизации для приложения или отдельной его страницы (сервер авторизации, параметры авторизации). Как правило, контейнер настраивается на уровне шаблона.

Разграничение контента (permission stack) – специальный контейнер, который позволяет для всех элементов внутри него настроить параметры отображения в зависимости от ролевой модели. В настройке может быть указано несколько состояний, для каждого из которых определяется свой объект полномочий (и

значение). Например, можно настроить два состояния – «младший менеджер» и «старший менеджер», определяемых объектом полномочий `manager_role = 1` и `manager_role = 2`. Элементы (диаграммы, кнопки и др.) вложенного контейнера со статусом «младший менеджер» будут отображаться если у пользователя в модели полномочий прописан объект полномочий `manager_role = 1`. С помощью такой модели можно гибко управлять ролевой моделью на уровне отдельных визуализаций, заменяя всю страницу или ее часть, скрывая отдельные разделы для разных бизнес-ролей.

5.3. НАВИГАЦИОННЫЕ ВИДЖЕТЫ

Платформа содержит необходимые динамические компоненты навигации (на основе данных). Таким образом навигация в интерфейсе приложения может осуществляться как по статическим маршрутам с возможностью программной обработки событий интерфейса (меню, кнопки, переход по ссылкам и окнам), так и по элементам, чье наполнение зависит от подключенного источника данных.

Таблица 2 Динамические навигационные компоненты на основе данных

Компонент	Ключевые особенности
Фильтр	<ul style="list-style-type: none"> • Специализированный компонент на основе выпадающего списка, динамически заполняемый данными из источника • Настройка вида отображения – выпадающий список, либо отображение контента в окне; • Поддержка вывода нескольких измерений; • Поддержка drill-down по измерению; • Поддержка сортировки по метрике или измерению; • Возможность встраивания картинок, ссылок и тегов; • Настройка цветовых схем и шрифтов.
Кнопочный фильтр	<ul style="list-style-type: none"> • Специализированный фильтр для переключения событий • Поддержка настройки без подключения датасета - с помощью Ручного ввода • Стилизация кнопок (активная/неактивная) с помощью интегрированных в виджет настроек, а также кастомных CSS-свойств; • Настройка цветовых схем и шрифтов; • Возможность встраивания картинок, ссылок и тегов; • Поддержка сортировки по метрике или измерению; • Поддержка drill-down по измерению; • Поддержка вывода нескольких измерений; • Поддержка режима Радио - автоматическое отключение действия прошлой кнопки при выборе новой кнопки; • Поддержка режима Мультиселект; • Возможность вывода первого значения из Датасета (настраивается в SDK).
Расширенный фильтр	<ul style="list-style-type: none"> • Специализированный виджет для фильтрации по измерению • Возможность встраивания бейджей с кастомной стилизацией и значений измерения в поле фильтра; • Настройка цветовых схем и шрифтов.
Мульти-фильтр	<ul style="list-style-type: none"> • Специализированный фильтр, работающий на нескольких источниках; • Возможность добавления и стилизации бейджей; • Настройка цветовых схем и шрифтов.

Компонент	Ключевые особенности
Меню	<ul style="list-style-type: none"> ● Меню в виде “гармошки” ● Поддержка свертывания вложенных элементов ● Поддержка навигации посредством ссылок
Бургер меню	<ul style="list-style-type: none"> ● Специализированный виджет для открытия меню в виде выезжающего списка для перехода на другие страницы внутри аппа; ● Поддержка сортировки по метрике или измерению; ● Настройка цветовых схем и шрифтов.
Боковое меню	<ul style="list-style-type: none"> ● Специализированный виджет для отображения меню в виде перечня пунктов меню для перехода на другую страницу; ● Поддержка настройки без подключения датасета; ● Возможность встраивания слотов для добавления других виджетов (например, картинку); ● Настройка цветовых схем и шрифтов.
Выезжающий контейнер	<ul style="list-style-type: none"> ● Специализированный виджет для создания дополнительного контента, который будет появляться при наведении или нажатии; ● Поддержка наполнения виджета любым контентом, доступным в библиотеке виджетов; ● Возможность настройки анимации - тип, длительность, задержка; ● Настройка цветовых схем и шрифтов.
Галерея	<ul style="list-style-type: none"> ● Специализированный виджет для отображения проектов через превью; ● Поддержка вывода тегов и подсказок; ● Поддержка перехода по ссылкам; ● Параметризация элементов виджета (ссылки картинок, ссылки аватаров, ссылки для перехода, теги и подсказки).
Чекбокс/Радио-кнопки	<ul style="list-style-type: none"> ● Специализированный формат фильтра в виде динамического чекбокса, радиокнопки или кнопки; ● Настройка вида отображения: <ul style="list-style-type: none"> ○ Чекбокс (checkbox); ○ Радиокнопка (radio button); ○ Кнопка (button); ● Поддержка сортировки по метрике или измерению; ● Поддержка drill-down по измерению; ● Настройка цветовых схем и шрифтов.
Навигационная цепочка («хлебные крошки», breadcrumbs)	<ul style="list-style-type: none"> ● Визуальное отображение пути от начального элемента к выбранному через все элементы иерархии; ● Поддержка drill-down по измерению; ● Настройка цветовых схем и шрифтов.
Поиск	<ul style="list-style-type: none"> ● Настраиваемый элемент поиска с поддержкой вывода нескольких измерений или метрик (таблицы); ● Поиск по нескольким полям; ● Моментальный поиск – при вводе символов; ● Встроенная пагинация вывода данных (дозапрос в случае большого количества найденных записей); ● Поддержка сортировки по метрике или измерению; ● Настройка цветовых схем и шрифтов.
Иерархический список	<ul style="list-style-type: none"> ● Подсчёт вложенных элементов ● Переход по ссылкам ● Отправка события ● Гибкая стилизация каждой строки списка (в т.ч. использование иконок)

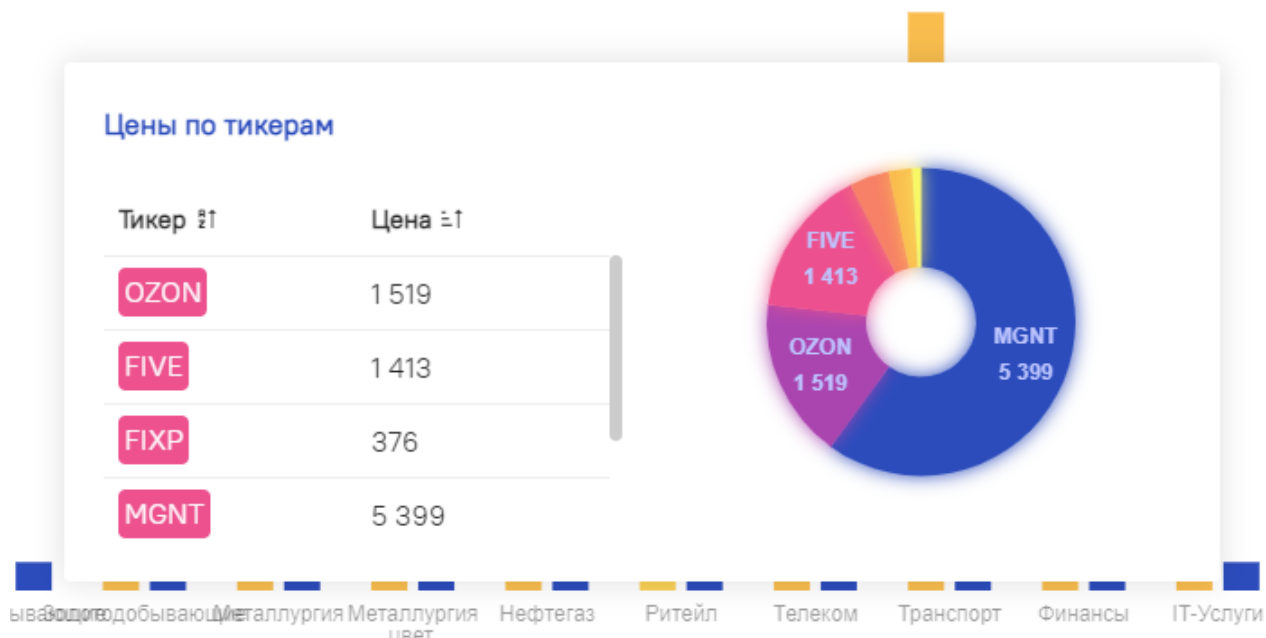
5.4. ПРОЧИЕ КОМПОНЕНТЫ

В состав Платформы включены все необходимые для создания веб-приложения компоненты:

- **Текст** – компонент для отображения текста (в формате HTML);
- **Ссылка** – компонент для отображения гиперссылки;
- **Выпадающий список** – компонент для отображения в виде выпадающего списка. В качестве элементов списка могут быть другие веб-компоненты;
- **Картинка** – компонент для вывода изображения (по URL изображения);
- **Кнопка** – компонент для отображения кнопки действия (отправки события в шину). Кнопка может являться контейнером для других элементов (например, текст, картинка);
- **Календарь** – компонент для вывода выпадающего календаря для выбора даты или интервала дат;
- **Домик** - компонент для настройки и фильтрации контента на странице путем отправки в остальные виджеты (или часть виджетов) переменной из датасета, текущей даты или набора конкретных значений в соответствующие переменные. Домик является невидимым и не занимает места на странице;
- **Динамическая картинка** - компонент для вывода изображений, загруженных в Менеджер файлов. Виджет позволяет, с помощью переменной, передаваемой от другого виджета, подставлять соответствующую картинку;
- **Аватар** - компонент для отображения данных о пользователе (лого, ФИО, должность и т.д.) с возможностью параметризации;
- **Видео** - компонент для отображения видео;
- **IFrame** - компонент для отображения контента других приложений и внешних ресурсов с помощью ссылки;
- **Попап по условию** - компонент для вывода всплывающего окна при первой загрузке страницы за период с возможностью настройки этого периода;
- **Кнопка скачивания данных** - компонент для скачивания информации из датасета в формате excel-файла;
- **Регламентный отчет** - сервис для формирования регламентной отчетности.
- **Скриншот** - компонент для скачивания страницы или области страницы в форматах pdf, png, jpeg, webp.
- **Автообновление** - компонент для обновления виджетов на странице по таймеру или событию (в том числе, от сокет-сервера).
- **Расширенный тултип** - полностью кастомизируемая подсказка.
 - Поддержка произвольного расположения контента в границах тултипа;
 - Поддержка стилизации элемента (фон, шрифты, цветосхемы, границы, размер, расположение);

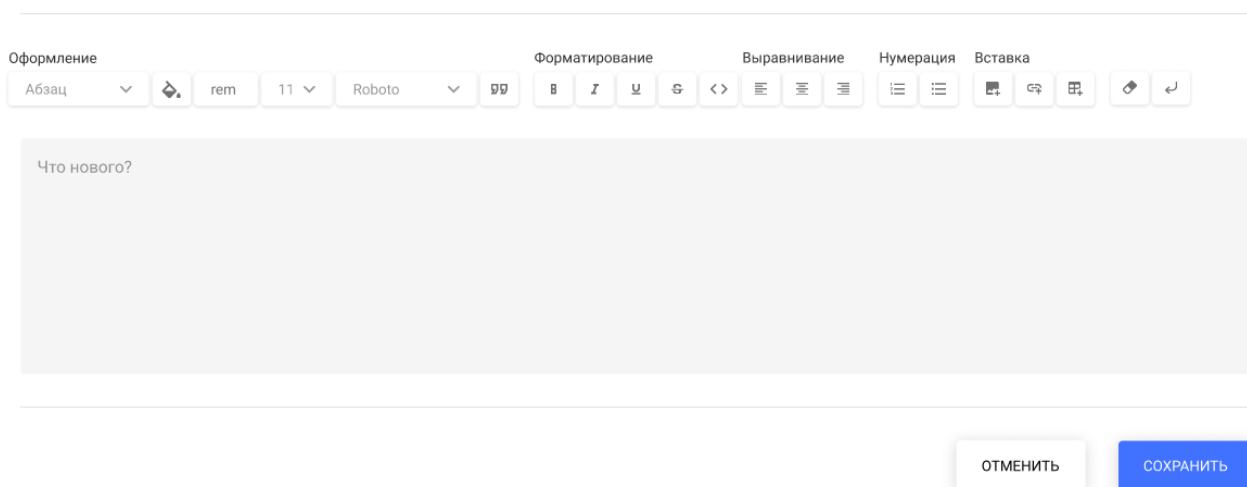
- Элемент может включать в себя неограниченный набор значений метрик и измерений, а также слоты с любым контентом библиотеки виджетов Insight low code.

Рисунок 22 Пример настроенного Расширенного тултипа



- **WYSIWYG** - визуальный редактор для текстовых элементов. Компонент поддерживает большое количество html и css настроек, вынесенных в отдельные ui-компоненты.

Рисунок 23 Панель визуального редактора WYSIWYG



Набор веб-компонентов Платформы постоянно расширяется. Кроме того, есть возможность написания собственных веб-компонентов с помощью JavaScript, HTML, CSS и встраивания их в перечень доступных в Конструкторе компонентов.

5.5. CSS-ФРЕЙМВОРК INSIGHT LOW CODE™

Платформа содержит специально разработанную библиотеку стилей (CSS-framework Insight low code), описывающую стандарты и веб-компоненты, используемые при разработке приложений и дашбордов. Данный подход обеспечивает возможность централизованного управления стилями на уровне отдельных компонентов для всех проектов в рамках конкретного Заказчика – управление цветами, шрифтами, отступами и иными элементами веб-верстки.

Изменение в шаблоне CSS автоматически меняет все проекты, использующие классы из библиотеки стилей.

В состав библиотеки стилей входят:

- Типографика (управление шрифтами, абзацами);
- Управление таблицами;
- Описание и управление сеткой;
- Компоненты:
 - предупреждения (alerts);
 - аватары (avatars);
 - бейджи (badges);
 - хлебные крошки (breadcrumb);
 - кнопки (buttons);
 - разделители (dividers);
 - выпадающие списки (dropdown);
 - заголовки (headers);
 - меню (menu);
 - классы навигации (nav);
 - канва (Canvas);
 - пагинация (pagination);
 - панели (panel);
 - всплывающие окна (popup);
 - загрузчики (preloader);
 - ширмы (shim);
 - скелетоны (skeletons) ;
 - боковые панели (sidebar);
 - табы (tabs);
 - плитки (tiles);
 - всплывающие сообщения (toast);
 - подсказки (tooltip);
- Формы ввода:

- поле ввода (input);
- текстовый блок (textarea);
- выбор (select);
- чекбокс (checkbox);
- радиокнопка (radio);
- переключатель (switch);
- иконки (icon).

В Insight low code CSS-framework 2.0 были добавлены утилиты для стандартизации, кастомизации объектов системы, которые включают в себя:

- расположение (#Float) - .pull-left, .pull-right и т.д.;
- текст (#Text) - .text-bold, .text-justify и т.д.;
- цвет (#Color) - .color-green, .color-transparent и т.д.;
- фон (#Background) - .bg-green, .bg-transparent и т.д.;
- вертикальное выравнивание (#Vertical align) - .v-top, .v-base и т.д.;
- внутренний отступ (#Padding) - .pad-none, .pad-l-{1-7}, .pad-h-{1-9} и т.д.;
- внешний отступ (#Margin) - .mar-auto, .mar-{1-9} и т.д.;
- тип отображения (#Display) - .d-flex, .show и т.д.;
- тип выравнивания (#Flex) - .flex-col, .flex-center и т.д.;
- позиционирование (#Position) - .pos-abs, .pos-fixed и т.л.;
- ширина (#Width) - .w-auto, .w-minс и т.д.;
- высота (#Height) - .h-auto, .h-fitс и т.д.;
- преднастроенные отступы (#Layout) - .section, .p, .block и т.д.;
- скролл (#Scroll) - .scroll-x, .scroll-y, .noscroll, .scroll-hide;
- функциональность курсора (#Events) - .events-none, .events-all;
- внешний вид курсора (#Cursor) - .cursor-auto, .cursor-default, .cursor-pointer, .cursor-help;
- визуальный фильтр (#Filter) - .filter-blur, .filter-grayscale, .filter-invert;
- формат заполнения содержимым (#Object) - .obj-contain, .obj-cover;
- визуальные эффекты для содержимого (#Misc) - .thumb, .round и т.д.

Платформа также поддерживает управление шаблонами страниц приложений, обеспечивая единство стилей, навигационных элементов, меню для различных проектов. Изменение шаблона автоматически изменяет все связанные с ним страницы и приложения.

Для компонентов определены классы CSS, позволяющие легко настраивать веб-компоненты с соблюдением утвержденного стиля. Полное описание CSS фреймворка Insight low code приводится в технической документации. Ниже приведены некоторые примеры.

Рисунок 24 Примеры определения стилей в CSS фреймворке

Alert

[preview](#) [code](#)

Title
Body content

Стили

Для изменения стиля используйте классы `.alert-success` `.alert-warn` `.alert-error`

[preview](#) [code](#)

Title
Body content

Title
Body content

Title
Body content

Стили

Для изменения стиля кнопки используйте классы `.btn-primary` `.btn-secondary` `.btn-success` `.btn-warn` `.btn-error` `.btn-misc` `.btn-outline` `.btn-ghost` `.btn-inline`

[preview](#) [code](#)

`.btn-primary` `.btn-secondary` `.btn-ghost` `.btn-inline`

`.btn-primary` `.btn-success` `.btn-warn` `.btn-error` `.btn-misc`

`.btn-primary` `.btn-success` `.btn-warn` `.btn-error` `.btn-misc`

Состояния

Для удобства btn имеет спец. классы `.hover` `.active` `.focus` `.disabled`, повторяющие состояния `hover` `active` `focus` `disabled`

Если вы используете для `.btn` отличные элементы от `<button>`, не забудьте, что для работы `.focus` состояния, необходимо указывать атрибут `tabindex="0"`

[preview](#) [code](#)

normal hover active focus disabled

Badge

[preview](#) [code](#)

badge

Стили

Для изменения стиля кнопки используйте классы `.badge-primary` `.badge-success` `.badge-warn` `.badge-error` `.badge-misc`

Для закругления краев используйте утилиту `.round` (см. [утилиты](#))

[preview](#) [code](#)

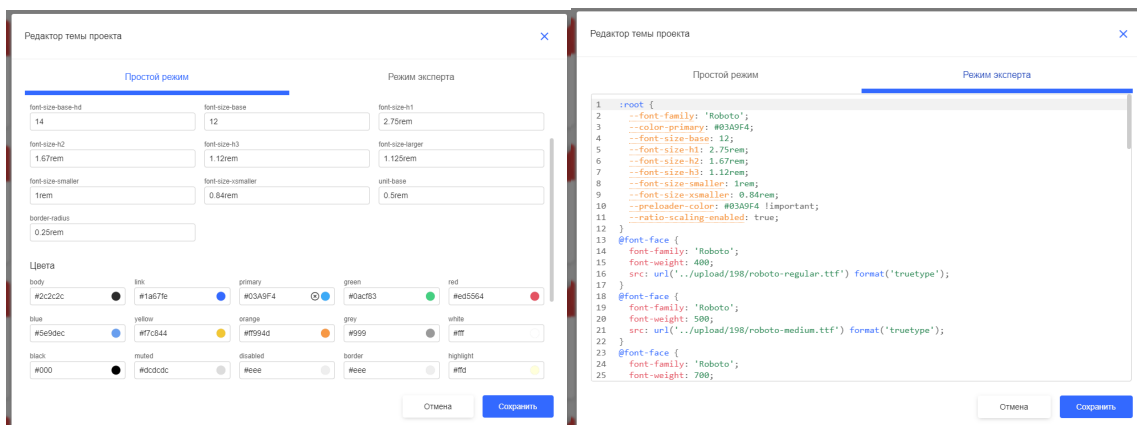
badge `.badge` `.badge` `.badge` `.badge` `.badge` `.badge`

При необходимости можно доопределить или переопределить стиль на уровне приложения в настройках темы приложения. Таким образом, с помощью **Темы проекта**, можно задавать специализированные шрифты и иные элементы на уровне конкретного проекта в дополнение к централизованным стилям на уровне фреймворка.

Тема проекта - панель настройки типографики, цветов и прочих компонентов CSS Framework на уровне конкретного проекта.

Опционально доступен "Режим эксперта", который позволяет вручную дополнять тему проекта на языке `css.u`

Рисунок 25 Панель настройки Темы проекта



5.6. ЛОГИРОВАНИЕ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ

Платформа поддерживает подмодуль, позволяющий анализировать действия пользователей.

Данный механизм осуществляет мониторинг рабочих операций пользователя для анализа информации о посещении приложений и конкретных страниц, а также для сбора статистики в разрезе авторизованных пользователей.

5.7. ФОРМЫ ВВОДА

Виджеты для настройки форм ввода в редакторе.

Для настройки и подключения форм ввода вначале необходимо использовать виджет Форма (elem form), который позволяет параметризовать тип операции (создать, изменить и т.д.), а также отображает список полей с индикацией обязательности заполнения после настройки ссылки на метод.

Виджет позволяет встроить любой элемент системы внутри Формы, а также использовать ui-компоненты для ввода:

- **Кнопка (elem button)** - виджет для ввода данных в формате кнопки;
- **Чекбокс (elem checkbox)** - виджет для ввода данных, где пользователь может выбрать один вариант;
- **Чекбокс группа (elem checkbox group)** - виджет для ввода данных, где пользователь может выбрать один или несколько вариантов;
- **Строка (elem input)** - виджет для ввода однострочных данных в формате текста, например, сообщение;
- **Дата (elem input date)** - виджет для ввода даты кастомным способом или при помощи календаря;

- **Радио кнопки (elem radio group)** - виджет для ввода данных, где пользователь может выбрать только один вариант
- **Выпадающий список (elem select)** - виджет для ввода данных в формате списка опций для выбора;
- **Абзац (elem textarea)** - виджет для ввода многострочных данных в формате текста.

В каждом ui-компоненте ввода настраивается связь с полями, необходимыми для заполнения, а также специфические настройки в зависимости от логики использования.

Форма ввода позволяет настроить пост-действия после выполнения операции (нажатие кнопки):

- Отправка события;
- Отправка переменных в шину событий;
- Переход по ссылке.

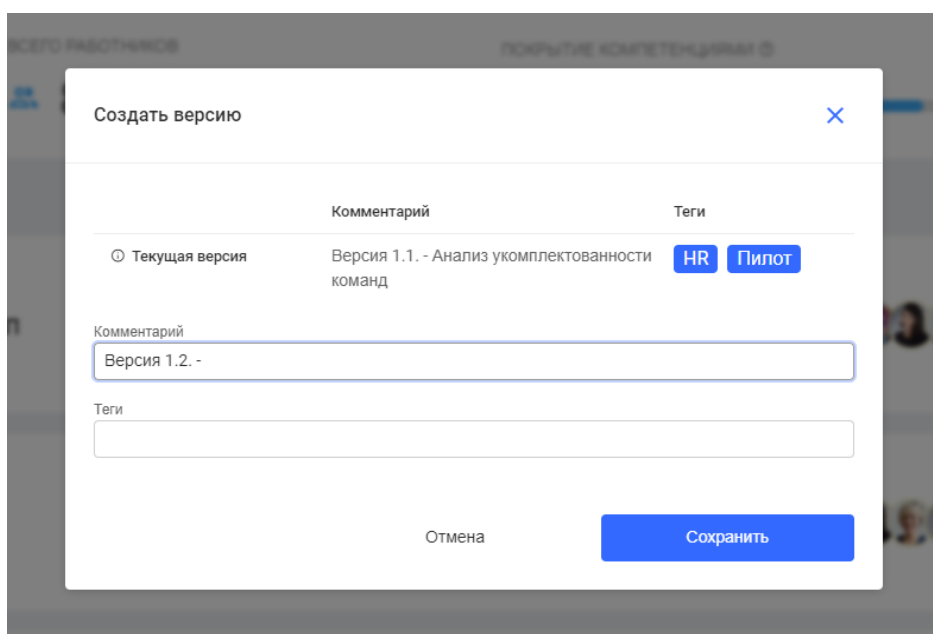
5.8. СОЗДАНИЕ ВЕРСИЙ ПРОЕКТА

В Insight low code есть возможность:

- Создавать;
- Хранить;
- Скачивать версии проекта в формате zip-архива.

При создании новой версии поддерживается комментирование и тегирование версии:

Рисунок 26 Создание новой версии проекта



Создать версию

Комментарий Теги

Текущая версия

Версия 1.1. - Анализ укомплектованности команд **HR** **Пилот**

Комментарий

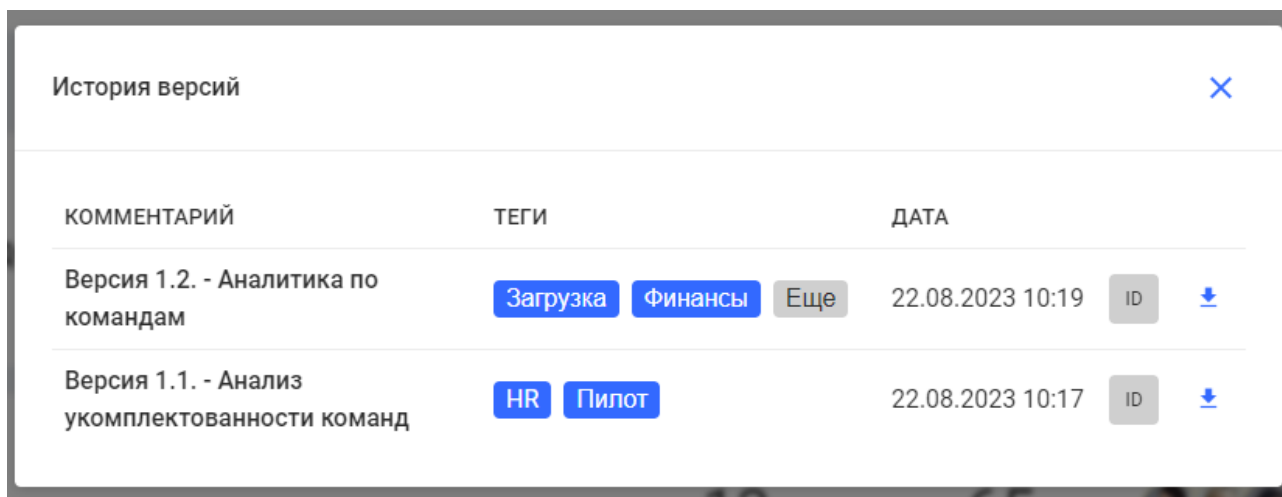
Версия 1.2. -

Теги

Отмена Сохранить

В панели настроек проекта доступна История версий, в которой можно скачать необходимую версию в формате zip-архива:

Рисунок 27 История версий проекта



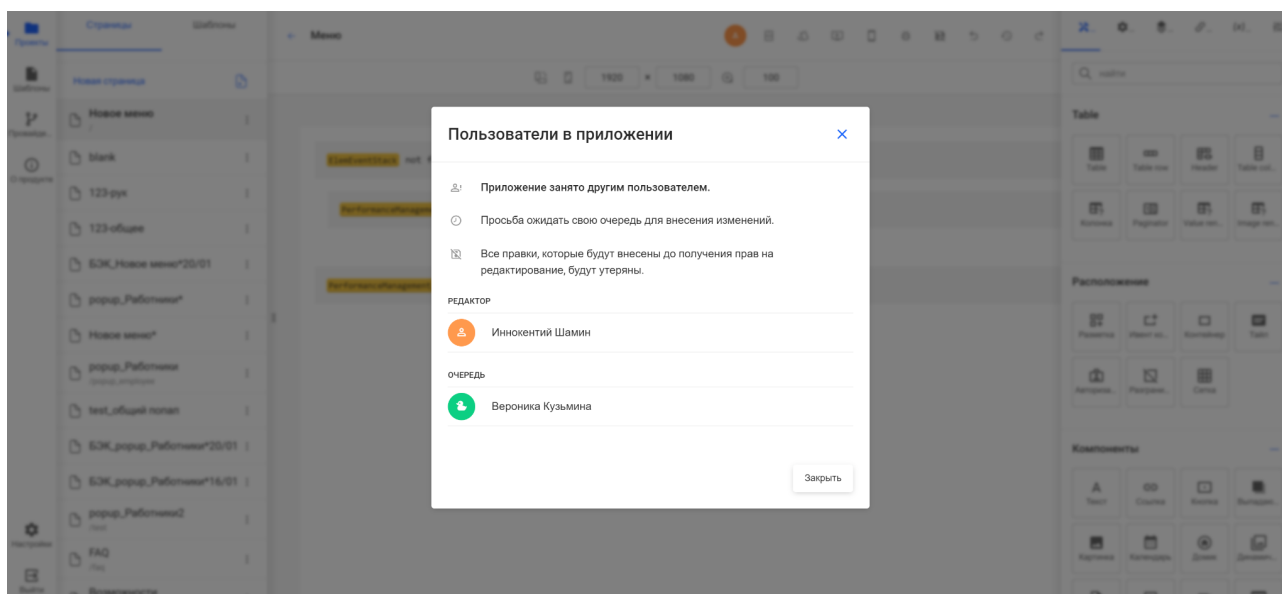
5.9. СОВМЕСТНАЯ РАБОТА

Функционал "Очередь" позволяет избежать ситуаций, когда в проекте работают несколько человек и "перезатирают" изменения друг друга.

В случае входа разработчика в проект, в котором находится другой пользователь, появится уведомление о:

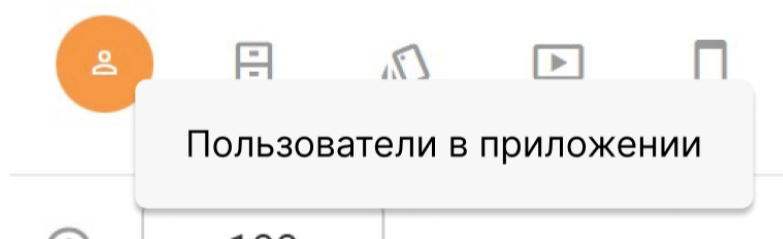
1. Пользователе, который является Редактором и может вносить правки в проект.
2. Пользователях, которые находятся в очереди и могут лишь просматривать проект:

Рисунок 28 Уведомление о пользователях в приложении



В верхней панели останется уведомление о первом пользователе, вошедшем в проект. Именно он сможет сохранять внесенные им изменения в проект, а остальные разработчики будут предупреждены о необходимости дождаться своей очереди для внесения правок:

Рисунок 29 Уведомление о пользователях в приложении



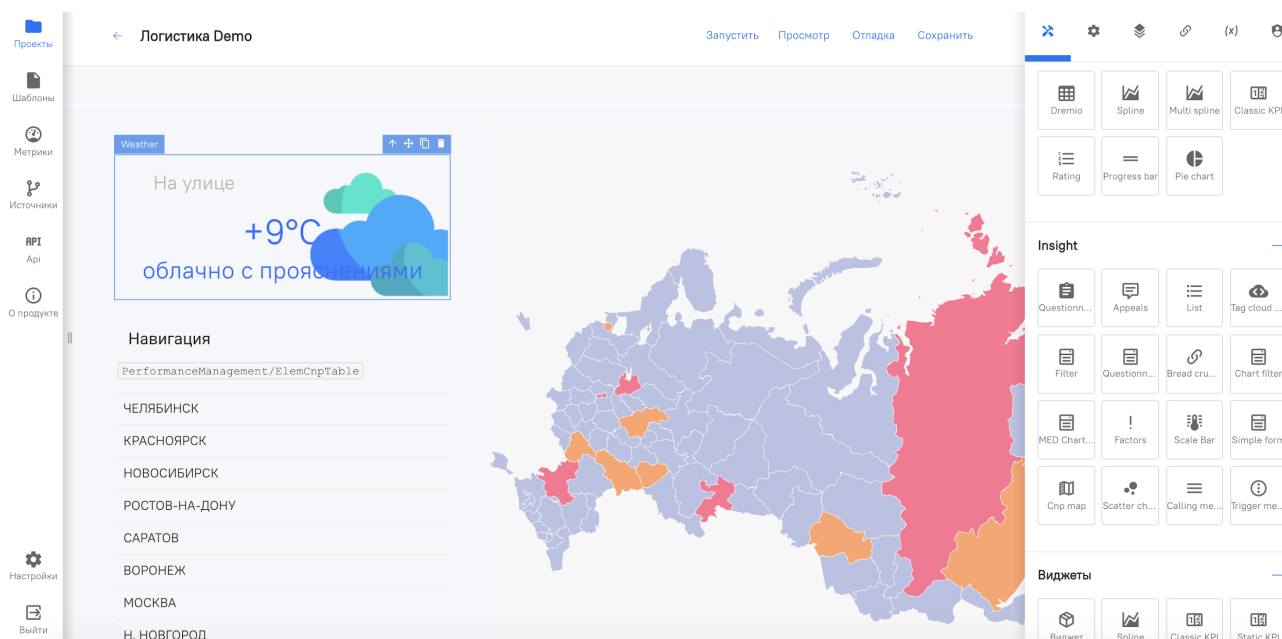
5.10. ПУБЛИКАЦИЯ ПРИЛОЖЕНИЙ

5.10.1. МИКРО-ПРИЛОЖЕНИЯ

Отдельный класс виджетов в Платформе – это микроприложения. С их помощью может реализовываться комплексная сложная логика работы с данными. Микроприложения позволяют не только визуализировать данные (в табличной форме, в виде интерактивных диаграмм), но и работать на запись данных в приложения и базы данных (формы ввода, запуск внешних процессов). С помощью микроприложений можно реализовать сколь угодно сложную логику работы с бэк-енд сервисами, внешними приложениями через вызов API, использовать все возможности аналитического SDK Insight low code и единой библиотеки стилей.

На рисунке представлен пример простого микроприложения «Погода», получающего информацию с сайта погоды и представляющего в виде стилизованной анимации.

Рисунок 30 Пример виджета микроприложения «Погода»



Как и другие типы виджетов, микроприложения могут иметь собственные параметры настройки – подключение к данным SDK Insight low code, внешним API (на чтение или запуск процессов), параметры конфигурации внешнего вида и бизнес-логики.

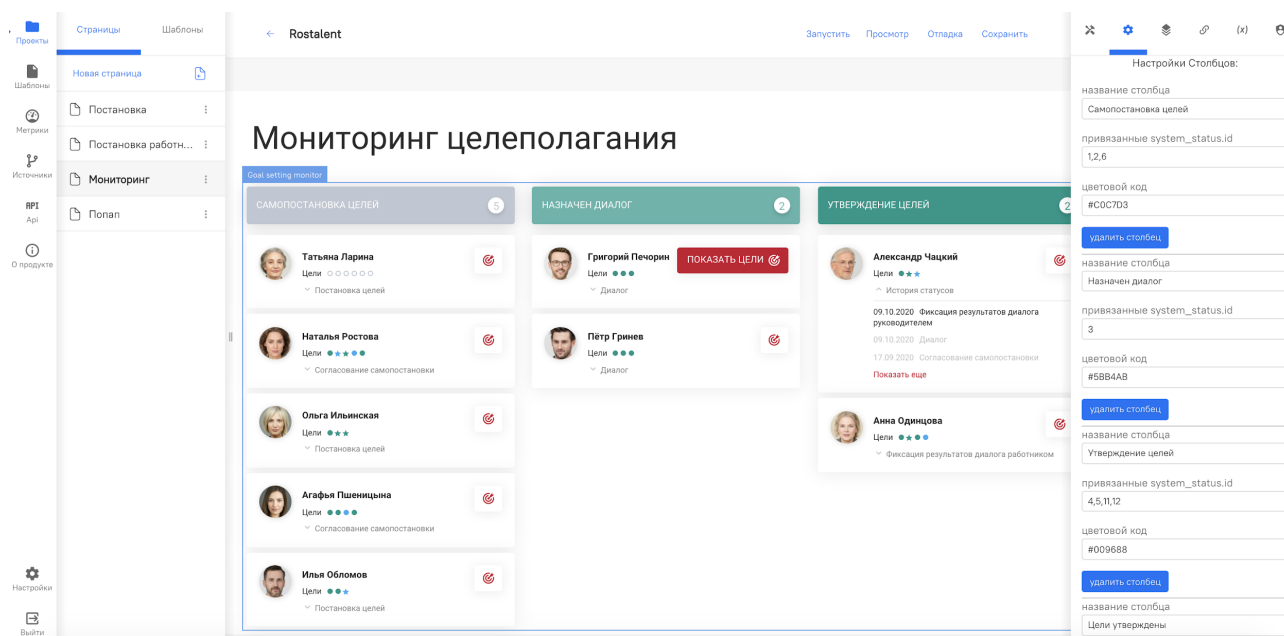
Платформа построена по модульному принципу, таким образом, что разработчикам виджетов микроприложений не нужно описывать подключение к данным, достаточно указать, что виджет использует SDK Insight low code. Соответствующие диалог настройки будет автоматически добавлен к виджету.

Как и диаграммы, микроприложения могут использовать шину событий для чтения параметров и передачи параметров в шину. Таким образом, микроприложения могут бесшовно интегрировать бизнес-логику внешних приложений в создаваемое веб-приложение.

Например, в примере выше, по клику на регион на карте в шину событий будет передаваться параметр региона и виджет погоды может быть обновлен. Аналогично, по клику на кнопку в микроприложении может быть вызван API внешнего приложения, которое направит уведомление ответственного за указанный регион.

Ниже на рисунке приведен пример сложного микроприложения, позволяющего управлять карточками задач в зависимости от их статуса, проваливаться в детали и изменять данные в базе данных путем перетаскивания карточек между столбцами (drag-n-drop).

Рисунок 31 Пример сложного микроприложения



5.10.2. ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ

В процессе разработки приложения аналитик может запустить приложение на просмотр и проверить работу, логику и представление данных с учетом параметров доступа авторизованного пользователя.

После завершения редактирования разработанное приложение может быть выгружено в виде набора файлов, описывающих структуру приложения и все необходимые инструкции для выполнения в проигрывателе (плеере). В файлах в виде JSON описываются все метаданные, инструкции SDK Insight low code ML, настройки виджетов, структуру страниц и стилей, логику перехода и переменные. Файлы не содержат каких-либо реальных данных, кроме метаданных.

Помимо файлов приложений может быть выгружена в файл схема данных, описывающая структуру используемых запросов (физических и виртуальных датасетов), параметры подключений к источникам и др. Выгружаемые файлы содержат только метаданные и не содержат строки самих данных.

Файлы приложений и схемы данных могут быть перенесены и загружены на тестовый и продуктивный ландшафт с помощью интерфейса администратора. Как правило, при разработке применяется трехуровневый ландшафт – среда разработки, среда тестирования и продуктивная среда.

Файлы схемы данных переносятся на тестовый, затем на продуктивный ландшафт, где производится настройка параметров подключения запросов к источникам данных (пути и параметры соединения), настраивается сервер авторизации и таблица ролевой модели.

Конструктор приложений не применим для тестового и продуктивного ландшафта.

Доступ пользователей к приложениям осуществляется по ссылке. В продуктивном ландшафте файлы приложений поступают на вход проигрывателю (плееру).

Плеер приложений представляет собой небольшой JavaScript код, который по инструкциям в файле выводит в браузере готовое приложение. Вместе с плеером запускаются все необходимые библиотеки и виджеты JavaScript, работает SDK Insight low code, шина событий. Пользователь видит реальные и обновляющиеся данные, может запускать и инициировать действия в API внешних систем т.д.

Приложения могут быть встроены в виде компонентов в другие порталы и веб-страницы с помощью специализированной JavaScript фабрики Insight low code Factory.

Фабрика Insight low code Factory позволяет коду приложений выполняться в любой среде и фреймворке JavaScript. При этом отдельные компоненты Insight low code, размещенные на странице с помощью Insight low code Factory могут взаимодействовать между собой через шину событий, как если бы они были размещены в одном приложении.

Безопасность на уровне приложений определяется следующими факторами:

- Только авторизованный пользователь имеет доступ к приложению. Пользователи, не прошедшие авторизацию, увидят либо специализированное уведомление, либо пустой элемент.
- Время доступа определяется параметрами токена сессии авторизации и может регулироваться централизованно администратором;
- Все соединения осуществляются с помощью защищенного протокола HTTPS;
- Ролевой доступ регулируется объектами полномочий на уровне компонентов приложений и на уровне данных (до уровня строк таблиц данных)

5.10.3. ПРОЧИЕ ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Платформа обеспечивает реализацию следующих функциональных возможностей в части моделирования:

- Моделирование (проведение анализа «что если») – изменение параметров и пересчет всех связанных с ними формул и визуализаций «на лету»;
- Возможность сравнения произвольно сформированных групп данных – сравнительный анализ;
- Возможность интерактивной детализации до первичных данных и возврата на вышестоящие уровни (drill-down/up);
- Расширенный drill-down - изменение контента (например, типа виджета), отображение слотов с микро-приложениями, трансформация drill-down в попап.

Платформа содержит встроенные средства поиска:

- Настраиваемые виджеты поиска, обеспечивающие полнотекстовый поиск и фильтрацию по полям данных (измерениям и метрикам);

- Поиск среди датасетов (физических и виртуальных) включая метаданные и теги.
- Поиск по приложениям, включая метаданные и теги.
- Поиск по виджетам.

Платформа поддерживает настраиваемую сортировку и фильтрацию данных.

6. ВИЗУАЛИЗАЦИЯ ДАННЫХ

6.1. АНАЛИТИЧЕСКИЕ ВИДЖЕТЫ

Платформа содержит следующие стандартные виды диаграмм и визуализации для целей анализа данных:

Таблица 3 Перечень поддерживаемых стандартных визуализаций

Тип визуализации	Ключевые особенности
Аналитические диаграммы	
Столбчатая вертикальная и горизонтальная диаграмма (bar chart) и диаграмма с накоплением (stacked bar chart)	<ul style="list-style-type: none"> • Настройка вывода меток значений итогов столбцов и подитогов внутри столбцов; • Настройка вывода/скрытия осей, подписей и заголовков; • Настройка цветов, размера и стиля шрифтов для показателей, осей, легенды, фона; • Настройка цветов для отдельных элементов измерений; • Настройка анимации при изменении выбора значения; • Поддержка drill-down по клику на значение; • Поддержка drill-down по клику на элемент диаграммы (бар).
Линейная диаграмма	<ul style="list-style-type: none"> • Настройки, аналогичные столбчатой диаграмме.
Комбинированная диаграмма	<ul style="list-style-type: none"> • совмещенная столбчатая, линейная, точечная – с настройками аналогичными столбчатой.
Столбчатая диаграмма сравнения	<ul style="list-style-type: none"> • Специализированная диаграмма, показывающая совмещенные отклонения в одном столбце. Например, прирост или падение показателя. • Настройки, аналогичные столбчатой диаграмме.
Ключевой показатель	<ul style="list-style-type: none"> • настройка отображения произвольного текста, размера, цвета; • вывод произвольных иконок-индикаторов и их цвета по условию (например, стрелка вниз/вверх, иконка человека, валюты и т.п.); • Настройка бенчмарков
Круговая диаграмма и Кольцевая диаграмма	<ul style="list-style-type: none"> • Настройка цветовых схем, шрифтов; • Настройка индивидуальных цветов для отдельных значений меры и измерений; • Настройка отображения легенды и отметок значений; • Настройка анимации; • Поддержка drill-down по клику на значение.
Сегментная диаграмма	<ul style="list-style-type: none"> • Вывод информации по сегментам в виде круговой диаграммы; • Настройка цветовых схем; • Настройка всплывающей подсказки при наведении на область виджета.
Каскадная диаграмма	<ul style="list-style-type: none"> • Диаграмма из последовательно отображаемых значений (прибыль и убыток), где конечная позиция одного элемента определяет начало следующего;

Тип визуализации	Ключевые особенности
	<ul style="list-style-type: none"> ● Представление в двух видах - в виде баров (bar-chart) и пузырьков (scatter-chart); ● Поддержка двух типов ориентации - горизонтальная и вертикальная; ● Поддержка drill-down по измерению; ● Окрашивание по условию (<0 и >0).
Пузырьковая диаграмма и Точечная диаграмма	<ul style="list-style-type: none"> ● Настройка размеров и цвета точек (пузырьков) в зависимости от значения метрики; ● Настройка индивидуальных цветов для отдельных значений измерений; ● Поддержка drill-down по клику на значение; ● Возможность группировки точек (пузырьков) по признаку Измерения для формирования Точечной диаграммы
Древовидная карта	<ul style="list-style-type: none"> ● Плиточное отображение данных, сгруппированных в категории; ● Поддержка drill-down по измерению; ● Окрашивание по условию; ● Возможность анимации; ● Поддержка трансформации карты в Лучевую диаграмму.
Поточная диаграмма	<ul style="list-style-type: none"> ● Настройка цветовых схем; ● Настройка всплывающей подсказки при наведении на ячейку; ● Поддержка drill-down по измерению.
Лепестковая диаграмма («Паутинка»)	<ul style="list-style-type: none"> ● Настройка цветовых схем; ● Настройка всплывающей подсказки при наведении на ячейку; ● Поддержка drill-down по измерению.
Радарная диаграмма	<ul style="list-style-type: none"> ● Настройка цветовых схем; ● Настройка всплывающей подсказки при наведении на ячейку; ● Поддержка drill-down по измерению.
Спидометр	<ul style="list-style-type: none"> ● Отражение хода достижения цели или значение ключевого показателя; ● Кольцевой формат виджета; ● Настройка цветовых схем.
Прогресс бар	<ul style="list-style-type: none"> ● Комплексная диаграмма для отслеживания динамики исполнения показателей (например, «Факт»/ «План») с возможностью отслеживания вспомогательных метрик (например, «Прогноз», «Значение предыдущего периода») ● Настройка цветовых схем и параметров отображения.
Простой таймлайн	<ul style="list-style-type: none"> ● Вывод информации в виде прогресса по этапам; ● Настройка цветовых схем.
Индикатор	<ul style="list-style-type: none"> ● настройка вида отображения: <ul style="list-style-type: none"> ○ спидометр; ○ термометр; ● задание диапазонов и цветовых схем для отдельных диапазонов.
Градусник	<ul style="list-style-type: none"> ● Поддержка нескольких режимов: классический, план/факт, проценты, батарейка; ● Настройка цветовых схем; ● Вывод изображений; ● Вывод Легенды; ● Ограничение на количество объектов в ряду; ● Окрашивание по условию.
Облако тегов	<ul style="list-style-type: none"> ● Настройка количества отображаемых тегов; ● Настройка цветов для разных значений метрики;
Факторы	<ul style="list-style-type: none"> ● Переключение событий; ● Поддержка иконок; ● Вывод прогресс-бара.

Тип визуализации	Ключевые особенности
SVG	<p>Виджет SVG позволяет создать интерактивный объект на основе изображения в формате SVG.</p> <p>Поддерживает:</p> <ul style="list-style-type: none"> ● Работу с несколькими объектами SVG- переименование объектов, скрытие/отображение; ● Связку датасета с объектами SVG через ID; ● Параметризацию объектов (в т.ч. по условию) - цвет элементов, цвет фона, прозрачность, тени, радиус, блюр; ● Параметризацию при клике и/или наведении на конкретный объект SVG; ● Отправку переменной другим виджетам и внутри виджета; ● Вывод значений Метрики и Измерения внутри объекта SVG.
Диаграмма Ганта	<ul style="list-style-type: none"> ● Настраиваемый формат отображения элементов; ● Поддержка свертывания вложенных элементов; ● Поддержка выделения просроченных элементов с помощью цвета и иных параметров стиля; ● Поддержка всплывающих подсказок при наведении на элементы; ● Поддержка кастомных периодов; ● Параметризация элементов диаграммы (бары) - цветовые схемы, величина, расположение); ● Отображение иконок, тегов; ● Вывод наименьшей даты начала и наибольшей даты окончания из датасета; ● Возможность стилизации выходных дней.
Тепловой календарь	<ul style="list-style-type: none"> ● Календарное отображение данных; ● Поддержка drill-down по измерению; ● Отображение иконок, тегов; ● Возможность стилизации выходных дней.
Дерево организационной структуры	<ul style="list-style-type: none"> ● Специализированный виджет для отображения организационной структуры - состава, расположения и взаимосвязи отдельных подсистем организации; ● Поддержка структуры произвольной глубины вложенности; ● Динамическое разворачивание/ сворачивание нижестоящих уровней по нажатию кнопки; ● Вывод изображений и настройки отображения отдельных элементов (стилей, текстов и других параметров); ● Вывод произвольных символов-иконок (например, индикаторы роста/падения, символы валют, иерархии и т.п.); ● Настройка цвета, размера текста и его отдельных элементов (выделить цветом, задать жирный текст или подчеркнуть и т.п.) – статично или по условию. ● Поддержка вертикального и горизонтального (разворот на 90°) отображения виджета.
Плитка с аватаром	<ul style="list-style-type: none"> ● Отображение карточки с информацией (заголовок, рассчитываемое значение) ● Отображение аватаров (картинки, фото сотрудников, инициалы - при отсутствии фото).

6.2. ТАБЛИЦЫ

Таблица 4 Перечень поддерживаемых видов таблиц

Тип визуализации	Ключевые особенности
Конструктор таблиц	<ul style="list-style-type: none"> ● Таблица представляет собой комплекс виджетов, которые позволяют собирать гибкие и разнообразные конструкции на основе табличного представления.

Тип визуализации	Ключевые особенности
	<ul style="list-style-type: none"> Использование разметки для конструирования строк и столбцов таблицы; Использование ui-компонентов из набора: список, изображение, бейдж, прогресс бар, чек-бокс, пагинация, заголовок; Упрощение работы с настройками стилей по сравнению с стандартным виджетом Таблица; Автоматическая генерация Таблицы содержимым Источника; Общая и индивидуальная настройка каждой колонки/строки/компонента для отправки событий, триггера переменных и перехода по ссылкам; Чрезвычайно гибкая кастомизация каждого из представлений.
Классическая таблица	<ul style="list-style-type: none"> Настройка формата отображения строк и столбцов (условное форматирование) - цвета, шрифты, стили CSS в зависимости от значения в ячейке; Динамическая сортировка по клику на иконку сортировки; Пагинация (постраничное отображение больших таблиц); Поиск в таблице; Отображение значений, иконок, иконок со значением, ссылок; Поддержка drill-down по клику на значение; Отображение только тех полей (столбцов) таблицы, которые есть в датасете (Транспонированная таблица); Формирование слотов (вложенностей) для строк Таблицы, которые позволяют размещать внутри себя другие виджеты; Формирование результирующих строк с различными типами агрегации; Формирование сложных (двухуровневых) заголовков Возможность вывода отклонений в абсолютном и процентном выражении.
Транспонированная таблица	<ul style="list-style-type: none"> Формирование Измерений-столбцов Настройка формата отображения строк и столбцов (условное форматирование) - цвета, шрифты, стили CSS в зависимости от значения в ячейке; Динамическая сортировка по клику на иконку сортировки; Пагинация (постраничное отображение больших таблиц);
Иерархическая таблица	<ul style="list-style-type: none"> Использование шаблонов для вывода информации в табличном виде в виде иерархии Автоматическое определение уровня иерархии родительских и вложенных таблиц
Канбан	<ul style="list-style-type: none"> Поддержка свертывания вложенных элементов; Регуляция отображаемого контента - количество отображаемых записей в колонках по умолчанию.
Список	<ul style="list-style-type: none"> Вывод созданного измерения списком; Вывод рассчитанной метрики в абсолютном или относительном значении для каждой позиции; Возможность функционирования в качестве фильтра; Группировка значений для каждой позиции с помощью отдельной метрики; Настройка цветовой схем;
Рейтинг	<p>Специализированная таблица, предназначенная для отображения рейтингов:</p> <ul style="list-style-type: none"> Визуальное отображение места в рейтинге – виде иконки с номером (местом в рейтинге); Отображение иконки-индикатора изменения места по сравнению с предыдущим периодом (рост/ падение в рейтинге, предыдущее место), значения показателя; Настройка цветовой схемы и стиля (шрифты, размер, структура);

Тип визуализации	Ключевые особенности
Сводная таблица	<ul style="list-style-type: none"> • Вывод изображений вместе с номером позиции. • Активные действия Пользователя в плеере: <ul style="list-style-type: none"> ○ Управление набором и порядком отображаемых Метрик и Измерений; ○ Вывод Измерений в формате строк и столбцов; ○ Изменение типа агрегации Метрик; ○ Вывод общих итогов — по строкам и по столбцам; ○ Управление форматом отображаемых Метрик; ○ Сортировка Измерений; ○ - Управлять иерархией измерений с помощью режима «Дриллдаун»; ○ - Выводить нумерацию строк; ○ - Работать в полноэкранном режиме; ○ - Выбирать Тип отображения — Табличный, Компактный или Структуру; ○ - Выводить промежуточных итогов — по строкам и столбцам; ○ - Добавлять расчетные метрики; ○ - Применять условное форматирование для Метрик.

6.3. КАРТЫ

Таблица 5 Перечень поддерживаемых видов карт

Тип визуализации	Ключевые особенности
Точечная карта	<ul style="list-style-type: none"> • Детализация до населенных пунктов, улиц и домов с использованием картографических сервисов (Open Street Map и др.); • Возможность нанесения пользовательских слоев на карту и использования картографических сервисов клиента • Кластеризация точек с возможностью детализации (drill-down) до уровня региональных и муниципальных образований; • Возможность масштабирования и центрирования виджета ; • Возможность “полета” к точкам карты ; • Настройка отображения легенды ; • Возможность настройки попапа ; • Поддержка всплывающих подсказок при наведении на элементы.
Матричная карта	<ul style="list-style-type: none"> • актуальная административная тепловая карта регионов Российской Федерации в формате матричной карты, схематично отражающей расположение регионов в виде ячеек одинакового размера • возможность смены представления в виде отсортированной тепловой карты по значению метрики
Карта «Хороплет»	<ul style="list-style-type: none"> • актуальная административная карта регионов Российской Федерации; • возможность использования произвольной карты или изображения в формате SVG; • возможность цветовой раскраски по градиенту (от худшего к лучшему и наоборот – автоматически в зависимости от выбранного показателя); • возможность цветовой раскраски по выполнению условия (например, зеленым регионы, выполнившие план, красным - не выполнившие);

Тип визуализации	Ключевые особенности
	<ul style="list-style-type: none"> • возможность вывода при наведении на регион подсказки со значением показателя с возможностью настройки формата и стиля подсказки и отображаемой информации.
Тепловая карта	<ul style="list-style-type: none"> • Настройка цветовых схем; • Настройка всплывающей подсказки при наведении на ячейку; • Поддержка drill-down по измерению.

6.4. ВОЗМОЖНОСТЬ РАЗРАБАТЫВАТЬ СОБСТВЕННЫЕ ВИДЖЕТЫ

Платформа позволяет пользователям разрабатывать новые виджеты, а также дорабатывать существующие.

7. SELF-SERVICE ОТЧЕТНОСТЬ

7.1. РЕГЛАМЕНТНАЯ ОТЧЕТНОСТЬ

Платформа поддерживает функционал работы с регламентированной отчетностью через виджет "Регламентная отчетность".

Система позволяет загружать размеченные шаблоны Excel и Word для последующей настройки выгрузки данных в заранее определенные места. При этом поддерживается выгрузка массивов данных и стилизация выгружаемой информации на уровне шаблонов.

Сервис поддерживает:

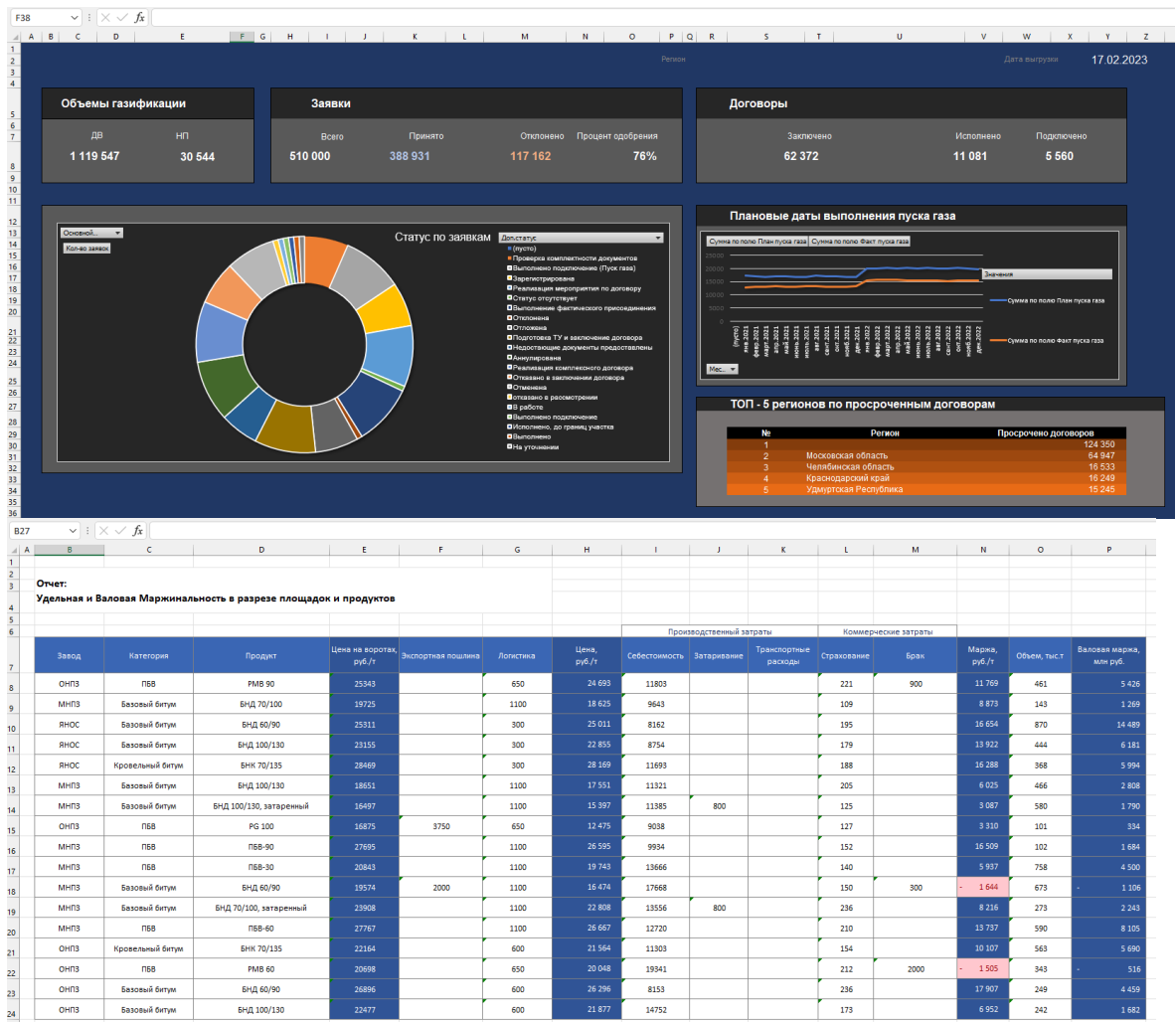
- Разметку шаблона — заполнение статическими данными (заголовки, текст, сноски и др.) и переменными. При выгрузке отчета из системы вместо переменных будут подставляться данные из источников — показатели.
- Форматирование и подготовку к печати средствами Excel и Word
- Расчетные показатели
- Настройку сводной аналитики в Excel — сводные таблицы, диаграммы и условное форматирование данных
- Единое хранилище шаблонов в системе.
- Настройку отчета с помощью параметризируемого виджета: подключение к данным.
- Возможность настройки разграничения доступа.

Регламентированная отчетность, как и любой виджет системы, работает с шиной событий и позволяет конкретизировать набор информации, который поступает в выгрузку.

Работа с регламентированной отчетностью заключается в нескольких шагах:

- 1) Создание шаблона и разметка под метрики/измерения;
- 2) Загрузка шаблона в систему;
- 3) Настройка источников данных/метрик/измерений для последующей выгрузки в шаблон.

Рисунок 32 Примеры готовых отчетов



7.2. ВЫГРУЗКА ВИТРИНЫ ДАННЫХ

Платформа обеспечивает возможность скачивания произвольного среза данных из витрины в формате **xlsx**, **csv** с учетом примененных фильтров, сортировки и пагинации для дальнейшего проведения анализа Пользователем.

7.3. ВЫГРУЗКА СТРАНИЦЫ ПРОЕКТА

Платформа обеспечивает возможность выгрузки страниц приложения в форматах:

- png
- jpeg
- webp
- pdf

Опционально доступно добавление даты к названию файла.

7.4. ВЫГРУЗКА ВИДЖЕТА

Платформа обеспечивает возможность выгрузки виджетов, а также областей страницы приложения в форматах:

- png
- jpeg
- webp
- pdf

Опционально доступно добавление даты к названию файла.

8. УПРАВЛЕНИЕ ДАННЫМИ И БЕЗОПАСНОСТЬ

Платформа поддерживает единую модель авторизации, аутентификации и управления полномочиями.

Все соединения между конечным пользователем и серверными компонентами, между самими серверными компонентами осуществляются строго по защищенному протоколу HTTPS с шифрованием данных SSL.

Параметры авторизации, включая срок жизни сессии и веб-токена определяются централизованно. Доступ к данным неавторизованным пользователям запрещен на уровне всех компонентов системы.

В случае необходимости обеспечения доступа анонимных пользователей к определенным приложениям и данным (например, в случае размещения на портале), работа пользователей ведется также с помощью токенов. На уровне сервера авторизации может быть полностью запрещен анонимный доступ. На уровне сервера данных, доступ анонимных пользователей также может быть запрещен. Как правило, для обеспечения безопасности работы такого сценария, выделяется отдельный сервер данных или схема данных, надежно разделяющая авторизованный доступ и анонимных пользователей, которым доступен только ограниченный набор данных.

Прямой доступ к серверу данных извне запрещен. Все запросы, направляемые через приложения пользователями к серверу данных, отправляются с помощью SDK Insight low code.

Обеспечение разделения ролей и полномочий осуществляется с помощью таблицы ролей и полномочий на уровне доступа к запросу данных, а также на уровне отдельных строк с помощью **объектов полномочий**.

Объект полномочий через пару ключ/значение определяет, какие значения полей запросов доступны пользователю.

Например, объект полномочий `dept_id = 1` определяет, что пользователю доступны только данные с подразделением с кодом 1 (`dept_id = 1`). Все запросы, содержащие поле `dept_id` будут автоматически отфильтрованы по полю `dept_id`. Пользователи,

которым не присвоен объект полномочий `dept_id` не получат записей из таблиц, в которых содержится указанное поле.

Пользователи, работающие с проектированием данных, которым даны полномочия на создание запросов в Платформе, также разделены ролями. Каждый такой пользователь видит только доступные ему источники данных и запросы.

Ролевая модель позволяет настраивать ограничение доступа к:

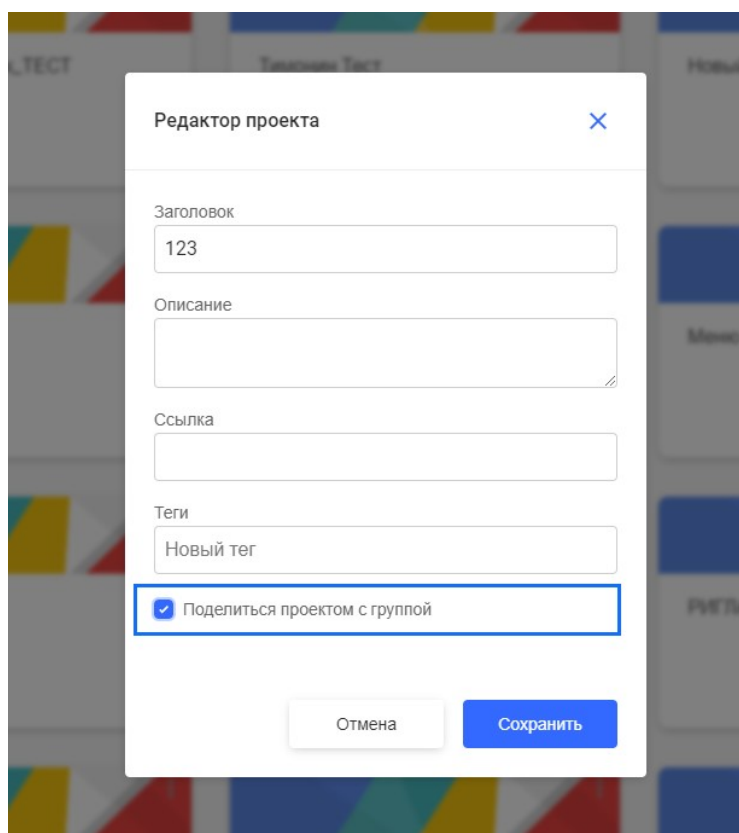
- Подключениям;
- Витринам данных (RLS);
- Приложениям;
- Областям приложения;
- Визуализациям

в зависимости от роли.

У каждого разработчика Insight low code существует возможность предоставить доступ к своему приложению другим разработчикам.

Доступ предоставляется к приложению с правами на просмотр, редактирование и удаление для всех пользователей из группы:

Рисунок 33 Интерфейс настройки предоставления прав группе



9. ДОПОЛНИТЕЛЬНЫЕ ОПЦИОНАЛЬНЫЕ КОМПОНЕНТЫ

9.1. Модуль БИЗНЕС-ПРАВИЛ WORKFLOW

Модуль Бизнес-правил (workflow) является опциональным модулем Платформы и используется в случае необходимости настройки уведомлений и согласований, настройки сложных моделей и формул принятия решений, координации взаимодействия приложений и ролевой модели в рамках организационной структуры предприятия.

Модуль Бизнес-правил предназначен для:

- настройка диаграмм бизнес-процессов (workflow) для цепочек согласования и уведомления – BPMN диаграмма
- настройка алгоритмов проверок и вычислений – таблиц DMN;
- использование данных оргструктуры для отдельной конфигурации бизнес-процессов на уровне отдельных структурных подразделений.

С помощью BPMN диаграмм можно описать бизнес-логику процесс рассылки уведомлений в случае наступления события (alert), порядок и текст уведомлений. Платформа позволяет автоматически определить соответствующие бизнес-роли, чтобы найти адресата уведомлений.

BPMN диаграмма может также использоваться для создания приложений, поддерживающих логику согласования каких-либо процессов по матрице согласования, с рассылкой соответствующих уведомлений и напоминаний. Основанием для инициации бизнес-процесса могут выступать объекты платформы – данные, метрики, отдельные визуализации.

С помощью DMN диаграммы можно организовать гибкую настройку сложных правил и моделей расчета показателей. Например, при наступлении нескольких событий (значения метрик данных выше порога), вычисляется выражение на основании этих метрик, сравнивается с неким порогом и рассчитывается значение метрики-флага. Использование DMN диаграммы вместо жестко зашитых формул метрик позволяет на уровне приложений организовать возможность бизнес-пользователям самостоятельно настраивать логику работы приложения без привлечения бизнес-аналитика и ИТ специалиста.

9.2. Модуль УВЕДОМЛЕНИЙ

Встроенная функциональность уведомлений позволяет организовать автоматическую рассылку уведомлений о наступлении какого-либо события. В качестве события может выступать достижение некоторого порога метрики.

С помощью брокера сообщений, подключенного к модулю Сбора и агрегации данных, модулю Бизнес-правил, при выполнении заданных условий может быть сгенерировано уведомление (alert). Опрос выполняется в режиме таймера (пакетная обработка) или непрерывно (streaming).

Уведомление автоматически рассылается по заданному в модуле workflow маршруту и выбранному средству коммуникации – электронная почта, push-уведомление на мобильное устройство, сообщение в мессенджере.

9.3. Модуль ОРГАНИЗАЦИОННОЙ СТРУКТУРЫ

Модуль Организационной структуры (Оргструктуры) является опциональным модулем, упрощающим работу модуле Уведомлений и Бизнес-правил. С помощью модуля Оргструктуры можно также упростить ведение ролевой модели на основе реальной системы управления предприятием.

В основе модуля Оргструктуры лежит встроенная готовая модель данных, описывающая различные варианты организационной структуры, перечень сотрудников и физических лиц с необходимыми атрибутами. Источником для наполнения данных модуля могут служить системы учета кадров (HCM), файлы XLS и иные источники. Платформа имеет предварительно разработанные коннекторы (средства, упрощающие интеграцию) с основными HCM системами: SAP HCM, Oracle E-Business Suite, 1C: ЗУП, MS Dynamics и др.

Модуль собирает основную информацию из интеграционного интерфейса учетной кадровой системы (HCM). При этом недостающая в учетной системе информация может быть введена непосредственно через экранные формы в модуле ДСП.

Модуль Оргструктуры поддерживает следующие виды организационной структуры:

- Юридическая оргструктура;
- Управленческая оргструктура;
- Функциональная (матричная оргструктура);
- Проектная оргструктура;
- Продуктовая (сегментная) оргструктура;
- Географическая оргструктура;
- Структура по местам возникновения затрат (МВЗ);
- Структура по местам возникновения прибыли (МВП).

Недостающая в системе HCM информация может загружаться в Платформу из различных источников и файлов.

Модуль Оргструктуры обеспечивает унификацию объектов в случае использования множества ИТ систем (например, единый идентификатор физического лица).

Модуль Оргструктуры обеспечивает API, необходимые для работы модуля Бизнес-правил, модуля Уведомлений, ролевой модели и микроприложений и внешних приложений:

- Руководителей сотрудника по выбранному виду оргструктуры (линейный, функциональный, проектный и др.);
- Перечень подчиненных по выбранному виду оргструктуры;
- Учет временных замещений и отсутствий;
- Иные API.

10. НЕФУНКЦИОНАЛЬНЫЕ ТЕХНИЧЕСКИЕ ВОЗМОЖНОСТИ

10.1. ОБЩИЕ ВОЗМОЖНОСТИ

Приложения и системы, создаваемые на основе Платформы, удовлетворяют следующим общим требованиям:

- Масштабируемость;
- Открытость;
- Модифицируемость;
- Надежность и безопасность.

Под **масштабируемостью** понимается возможность систем «расти» вместе с Заказчиком. Масштабируемость приложений и систем обеспечивается техническими средствами вертикальной и горизонтальной масштабируемости и функциональными (инструментальными) средствами разработки Платформы.

Под **открытостью** понимается возможность самостоятельного развития возможностей Платформы и наращивания ее функциональности. Это достигается за счет модульной структуры и возможности создания самостоятельных компонентов на ее основе – как на уровне визуального представления (виджеты), так и на уровне бэк-энда (встраиваемые микроприложения, сложная логика с использованием BPMN и DMN диаграмм модуля Бизнес-правил. Платформа также предоставляет ряд открытых API и возможности SDK Insight low code™ для работы внешних приложений.

Модифицируемость создаваемых систем обеспечивается возможностями Платформы, которые позволяют адаптировать решение под конкретные требования Заказчика и гибко изменять их по запросу.

Безопасность систем обеспечивается совокупностью средств Платформы, позволяющих разграничивать права доступа пользователей, исключая возможность несанкционированного доступа к создаваемым на базе Платформы системам.

Надежность систем определяется как свойство сохранять во времени в установленных пределах значения всех параметров, характеризующих способность систем выполнять свои функции в заданных режимах и условиях эксплуатации. Платформа содержит необходимые компоненты для обеспечения максимальной надежности и связанные с этим рекомендации к внедрению систем.

10.2. ОПЦИИ ИНСТАЛЛЯЦИИ ПЛАТФОРМЫ

Платформа устанавливается в двух вариантах поставки в зависимости от требований Заказчика:

- установка на серверах в контуре Заказчика;
- установка в защищенном частном облаке (private cloud) на территории РФ.

В случае установки в контуре Заказчика поставка (deployment) ПО осуществляется с помощью технологии контейнеризации/оркестрации Docker/Kubernetes.

Последующее обновление программного обеспечения также осуществляется путем обновления соответствующих контейнеров.

В случае установки в защищенном облаке, поддержка обновлений и инфраструктуры осуществляется централизованно провайдером услуги. Доступ к данным (инсталляции Платформы) строго ограничен конкретным Заказчиком в виде выделенного экземпляра Платформы. Предоставляемое облако защищено с точки зрения Федерального Закона 152 о защите персональных данных, иных требований законодательства и требований информационной безопасности, предоставляет настроенные сценарии резервного копирования и восстановления данных.

10.3. ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

Технические средства, на базе которых планируется эксплуатация Платформы, должны включать следующие виды технических средств:

- Серверы (баз данных и приложений) и их внешние системы хранения данных;
- Рабочие станции;
- Средства вывода информации (дисплеи, медиапанели)
- Мобильные устройства при наличии защищенного соединения с обязательным шифрованием, обязательных средств защиты информации в соответствии с действующими корпоративными стандартами и политиками информационной безопасности;
- Системы копирования и долговременного хранения данных;
- Средства ЛВС (активное и пассивное сетевое оборудование) и телекоммуникационное оборудование;
- Средства обеспечения бесперебойного питания и выделенной электрической сети;
- Средства защиты информации.

Предпочтительно использование виртуальных серверов с поддержкой масштабирования. Поставка и развертывание ПО осуществляется с использованием технологии контейнеризации/оркестрации Docker/Kubernetes.

Приведенные требования к серверному обеспечению являются предварительными и подлежат уточнению с учетом деталей конкретного Проекта внедрения.

Таблица 6 Рекомендуемая продуктивная конфигурация - до ~1 000 пользователей.

Наименование сервера	CPU	RAM, GB	SSD, GB
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	4	50
Сервис авторизации Сервис авторизации с поддержкой SSO и встроенными средствами интеграции с внешними системами авторизации типа Microsoft Active Directory и т.д. <i>Keycloak</i>	2	4	50
Сервис приложения Сервис для web/html и серверного приложений. <i>Nginx & Java applications</i>	7	32	100
Сервис СУБД Сервис поддержки хранения транзакционных данных на уровне бизнес-процессов платформы. Отдельные хранилища для хранения данных, полученных в результате интеграции, поддержки потоковой обработки данных и аналитических операций. <i>PostgreSQL</i>	4	8	2000
Сервис интеграции. Репозиторий ETL процессов. Сервисы экстракции и передачи данных из внешних источников, включая сервисы трансформации и преобразования данных. <i>Apache NFi & Apache Airflow</i>	8	16	50
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	4	50

Таблица 7 Рекомендуемая продуктивная конфигурация до ~ 10 000 пользователей.

Наименование сервера	CPU	RAM, GB	SSD, GB
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	4	50
Сервис авторизации Сервис авторизации с поддержкой SSO и встроенными средствами интеграции с внешними системами авторизации типа Microsoft Active Directory и т.д. <i>Keycloak</i>	2	4	50
Сервис приложения Сервис для web/html и серверного приложений. <i>Nginx & Java applications</i>	13.2	38.4	50
Сервис СУБД Сервис поддержки хранения транзакционных данных на уровне бизнес-процессов платформы. Отдельные хранилища для хранения данных, полученных в результате интеграции, поддержки потоковой обработки данных и аналитических операций. <i>PostgreSQL</i>	4	16	6000
Сервис интеграции. Репозиторий ETL процессов. Сервисы экстракции и передачи данных из внешних источников, включая сервисы трансформации и преобразования данных. <i>Apache NFi & Apache Airflow</i>	8	16	50
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	4	50

Таблица 8 Рекомендуемая продуктивная конфигурация до ~ 50 000 пользователей.

Наименование сервера	CPU	RAM, GB	SSD, GB
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	8	50
Сервис авторизации Сервис авторизации с поддержкой SSO и встроенными средствами интеграции с внешними системами авторизации типа Microsoft Active Directory и т.д. <i>Keycloak</i>	2	8	50
Сервис приложения Сервис для web/html и серверного приложений. <i>Nginx & Java applications</i>	17.6	51.2	160
Сервис СУБД Сервис поддержки хранения транзакционных данных на уровне бизнес-процессов платформы. Отдельные хранилища для хранения данных, полученных в результате интеграции, поддержки потоковой обработки данных и аналитических операций. <i>PostgreSQL</i>	8	32	12000
Сервис интеграции. Репозиторий ETL процессов. Сервисы экстракции и передачи данных из внешних источников, включая сервисы трансформации и преобразования данных. <i>Apache NFi & Apache Airflow</i>	16	32	50
Сервис точки входа Балансировщик входящих запросов <i>Nginx</i>	2	8	50

Представленные требования к конфигурации платформы являются **рекомендуемыми** с учетом опыта реальных проектов класса Enterprise. Возможность применения **минимальных** требований потребует более детального анализа в ходе проектирования Платформы.

Необходимая производительность технических средств и каналов связи должна быть уточнена в ходе проектирования Платформы расчетным путем, исходя из заданных показателей назначения Платформы в целом.

В качестве серверной операционной системы могут использоваться дистрибутивы, основанные на Linux. Рекомендуемая конфигурация Платформы – ALT Linux или Astra Linux.

Платформа не требует использования дополнительного программного обеспечения.

Рабочие станции должны обеспечивать:

- Отображение визуальной информации;
- Возможность отображения алфавитно-цифровой и графической информации в цветном режиме;

- Печать (вывод) информации на принтер и получение копий документов в текстовом и цветном графическом исполнении с разрешением не менее 300 точек на дюйм для графического режима;
- Запись и хранение информации на жестком диске.

Рабочие станции и серверы Платформы должны быть объединены в локальную сеть.

Платформа разработана и оптимизирована в соответствии со стандартом W3C и стабильно работает в последних версиях браузеров на момент согласования технической спецификации (с настройками браузеров по умолчанию).

Таблица 9 Типовые требования к рабочему месту пользователя

Техническая платформа	PC (Windows) MAC (MacOS) Операционные системы на базе Linux
Программное обеспечение	Тонкий клиент: Google Chrome версии 74 и выше; Microsoft Edge версии 85 и выше; Chromium 91 версия и выше; Mozilla Firefox версии 68 и выше; Safari версии 12.1 и выше; Яндекс.Браузер версии 20.3.0.1223 и выше; Спутник - версия 5.3.5380.0 и выше.
Минимальное разрешение экрана	Не ограничивается.
Максимальное разрешение экрана	Не ограничивается. Поддерживается работа на сверхбольших экранах и медиапанелях нестандартного разрешения.
Мобильные платформы (iOS, Android)	Google Chrome Safari

При использовании мобильной архитектуры предъявляются дополнительные требования к безопасности (на уровне доступа к определенным метрикам или организационным элементам). Средства доступа к мобильным приложениям (средства защиты соединения, дополнительные средства шифрования) не входят в состав Платформы и обеспечиваются Заказчиком в соответствии с корпоративными стандартами в области информационной безопасности.

10.4. ТРЕБОВАНИЯ К НАДЕЖНОСТИ

При разработке и внедрении систем на базе Платформы Insight low code необходимо обеспечивать:

- Доступность в рабочее время, установленное у Заказчика;
- Выполнение регламентных процедур вне рабочего времени;
- Возможность полного восстановления системы на конец предыдущего дня;
- Устранение ошибок, включая ошибки обмена данными, в течение зафиксированного в SLA времени.

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС;
- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Платформа представляет все необходимые программные инструменты для обеспечения максимальной надежности создаваемых на ее основе систем:

- контроля целостности данных на уровне СУБД;
- сохранения целостности базы данных при нештатном завершении программы (отказ рабочей станции и т.п.);
- сохранения работоспособности программного обеспечения при некорректных действиях пользователя;
- средства контроля корректности входных данных;
- средства оповещения пользователя об ошибках входных/выходных и противоречивых данных.

Непрерывность работы и отказоустойчивость системы должны быть обеспечены средствами инфраструктуры Заказчика в соответствии с требованиями к отказоустойчивости и режиму работы пользователей Платформы, а также к времени запуска и продолжительности работы операций, выполняемых в фоновом режиме.

Для достижения заданных параметров надежности рекомендуется использовать:

- Трехуровневый ландшафт развертывания Платформы (среда разработки, среда тестирования, продуктивная среда);
- Средства мониторинга работы Платформы – Платформа предоставляет в режиме реального времени логи для обработки внешними системами и встроенные средства мониторинга отдельных компонентов;
- Систему резервного копирования для обеспечения отказоустойчивости (не входит в состав Платформы в версии размещения у Заказчика). Резервное копирование реализуется в конкретной инфраструктуре Заказчика с помощью автоматической или ручной процедуры на основе регламентов резервного копирования и восстановления средствами программно-аппаратного комплекса Заказчика. В случае использования облачной версии Платформы, в состав включены средства резервного копирования;

- Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры;
- Трехуровневую службу поддержки;

Рекомендуемые параметры надежности, применяемые к серверному оборудованию:

- технические средства серверов СУБД должны обеспечивать возможность горячей замены компонентов аппаратного обеспечения сервера, возможность резервирования путей взаимодействия серверов;
- надежность активного сетевого оборудования должна обеспечивать время однократного простоя не более 10 мин, суммарного времени на регламентное обслуживание не более 32 часов в год;
- надежность серверов СУБД должна обеспечивать время однократного простоя не более 20 мин, суммарного времени на регламентное обслуживание не более 32 часов в год.

10.5. ТРЕБОВАНИЯ К ИНСТРУМЕНТАМ ДИАГНОСТИКИ И ЛОГИРОВАНИЯ

Системы, создаваемые на базе Платформы обеспечивают своевременное оповещение обслуживающего персонала программных продуктов о наступивших нарушениях работоспособности.

Платформа содержит встроенный набор инструментов мониторинга компонентов, а также поддерживает ведение журнала системных событий и действий пользователей:

- Лог выполнения задач по загрузке и трансформации данных (время выполнения, ошибки загрузки, некорректные или неполные данные);
- Лог авторизаций и сессий пользователей;
- Лог выполненных пользователями запросов к данным.

Собранные Платформой логи могут быть проанализированы внешними инструментами, принятыми у Заказчика для анализа логов, таких как Zabbix, ELK Logstash, Prometheus и другие системы.