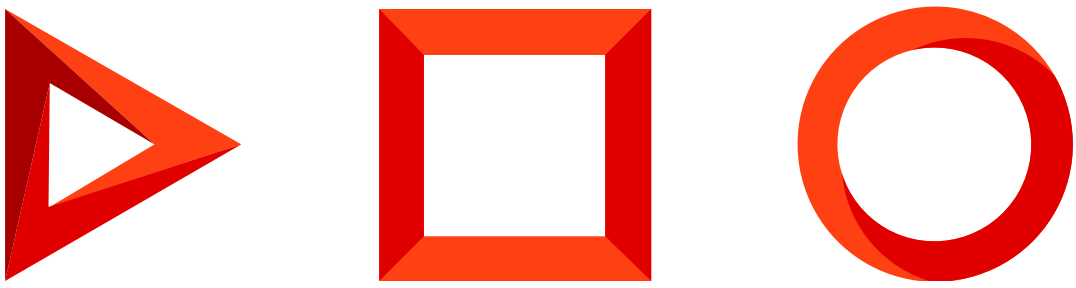
Creatio Academy

Архитектура

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Основное приложение Creatio

Основы

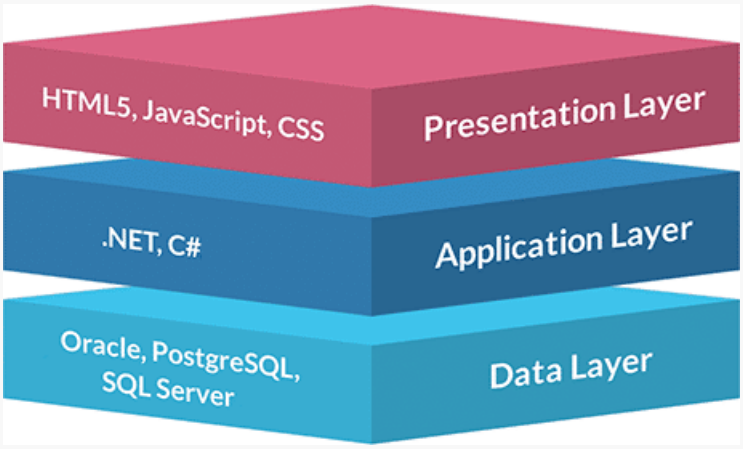
Классическая трехуровневая архитектура Creatio реализована с использованием современных технологий и инфраструктурных решений, которые придают приложению кросс-платформенность, гибкость и масштабируемость.

Возможные **схемы развертывания** приложения Creatio:

- **Без обеспечения отказоустойчивости**. Базовый вариант инфраструктуры без использования балансировщиков нагрузки.
- **С обеспечением отказоустойчивости**. Вариант инфраструктуры с горизонтальным масштабированием. Обеспечение отказоустойчивости реализуется за счет использования балансировщиков нагрузки серверов приложений, баз данных и кэширования.

Логические уровни приложения

Приложение Creatio имеет классическую **трехуровневую архитектуру** со следующими слоями: данные, приложение, представление.



Уровень представления

Уровень представления предоставляет пользователю доступ к интерфейсу приложения через веб-браузер либо мобильное приложение для Android или iOS. Содержит веб-страницы, код JavaScript и стили, определяющие **логику и внешний вид пользовательского интерфейса**.

Ключевые технологии, используемые на уровне представления: Angular, JavaScript, Ext.JS, HTML5, CSS.

Поддерживаемые браузеры: Chrome, Firefox, Edge и Safari.

В качестве альтернативного клиента реализовано мобильное приложение Creatio с адаптивным интерфейсом для Android или iOS.

Уровень приложения

Уровень приложения реализован для платформ .NET Framework и .NET Core. Может быть развернут на веб-серверах Windows, Linux, Mac. Уровень приложения определяет **основную бизнес-логику**, такую как динамическое управление кейсами, механизм бизнес-процессов, интеграция с телефонией и т. д.

Обрабатывает аутентификацию и авторизацию пользователей, выполняет проверку лицензий, создает экземпляры и запускает индивидуальную бизнес-логику, реализованную с помощью Creatio и low-code инструментов.
На уровне инфраструктуры представлен сервером приложений.

Уровень данных

Уровень данных хранит и управляет **данными** клиентов, настройками приложений, метаданными и данными аутентификации пользователей.

Используется для хранения в памяти данных сессии, часто используемых кэшей и быстрого взаимодействия между узлами в веб-ферме.

Поддерживаемые СУБД: Microsoft SQL Server, Oracle и PostgreSQL.

Сервер кэширования: база данных Redis.

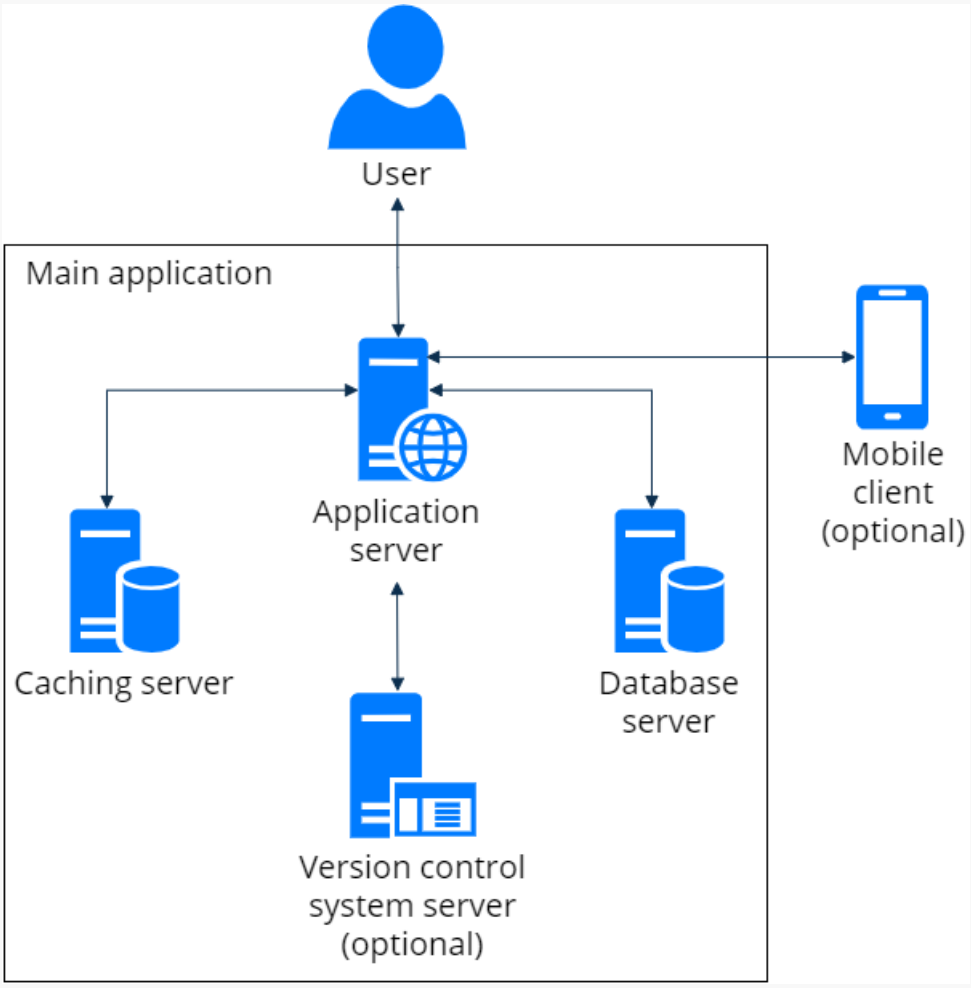
На уровне инфраструктуры представлен сервером кэширования и сервером баз данных.

Инфраструктура основного приложения

Компоненты инфраструктуры основного приложения:

- Сервер приложений.
- Сервер баз данных.
- Сервер кэширования.
- Сервер системы контроля версий (опциональный компонент). Используется при многопользовательской разработке.
- Мобильное приложение (опциональный компонент). Используется для доступа к основному приложению с мобильного устройства.

Общая схема архитектуры основного приложения Creatio без отказоустойчивости представлена на рисунке ниже.



На заметку. Архитектура основного приложения Creatio с отказоустойчивостью предполагает использование балансировщиков нагрузки, дополнительных серверов приложений, кэширования Redis и баз данных. Подробнее читайте в пункте Горизонтальное масштабирование.

Сервер приложений

Сервер приложений соответствует логическому уровню приложения и выполняет основную вычислительную работу системы.

Приложение доступно на платформах .NET Framework и .NET Core.

Варианты установки продуктов Creatio

Creatio products	.NET Framework	.NET Core
Marketing	+	+
Sales Enterprise	+	+
Sales Commerce	+	
Sales Team	+	
Service Enterprise	+	+
Customer Center	+	
Studio	+	+
Lending	+	
Bank Customer Journey	+	
Bank Sales	+	
Sales Enterprise & Marketing & Service Enterprise	+	+
Sales Enterprise & Marketing & Customer Center	+	
Sales Commerce & Marketing & Customer Center	+	
Sales Team & Marketing	+	
Sales Team & Marketing & Customer Center	+	+
Bank Sales & Bank Customer Journey & Lending & Marketing	+	+

Сервер приложений на платформе .NET Framework

Серверы приложений Creatio на платформе **.NET Framework** могут быть развернуты в Microsoft Internet Information Services (**IIS**) в **Windows**.

Требования к серверам приложений продуктов Creatio на платформе .NET Framework вы можете рассчитать в Калькуляторе системных требований.

Сервер приложений продуктов Creatio на платформе .NET Framework состоит из двух компонентов:

1. **Загрузчик** (`WebAppLoader`) — приложение, реализующее выполнение служебных функций системы, и

дальнейшее перенаправление пользователей в конфигурационную часть основного приложения Creatio.

Загрузчик отвечает за выполнение следующих действий:

- Авторизацию пользователей.
- Проверку лицензий и аутентификацию пользователей.
- Запуск планировщика, который отвечает за выполнение фоновых задач по расписанию.

На уровне файловой системы загрузчик размещен в корневой папке приложения.

После обработки в загрузчике запроса на авторизацию пользователи могут работать в конфигурационной части.

2. **Конфигурационная часть** (`WebApp`) — приложение, которое реализует конкретную конфигурацию в системе, а также отвечает за работу бизнес-логики системы.

   На уровне файловой системы конфигурационная часть размещена в папке `Terrasoft.WebApp`.

Сервер приложений на платформе .NET Core

.NET Core версии приложений доступны для развертывания в **ОС Linux** под управлением **Kestrel**.

Требования к серверам приложений продуктов Creatio на платформе .NET Core вы можете рассчитать в Калькуляторе системных требований.

Приложение Creatio на платформе .NET Core является неделимым и выполняет задачи **загрузчика** и **конфигурационной части**.

Описание продуктов Creatio на платформе .NET Core содержится в статье Продукты Creatio на платформе .NET Core.

Сервер баз данных

Сервер баз данных является частью логического уровня данных приложения.

В базе данных хранятся следующие элементы:

- Пользовательские данные.
- Данные, необходимые для работы самой системы.
- Конфигурационные настройки, которые определяют функциональность продукта.

В качестве системы управления базами данных (СУБД) могут использоваться следующие СУБД:

- MS SQL Server.
- Oracle (при развертывании on-site).
- PostgreSQL.

Актуальные версии поддерживаемых СУБД вы найдете в Калькуляторе системных требований после выполнения расчета.

Варианты установки продуктов Creatio

Creatio products	MS SQL	Oracle	PostgreSQL	PostgreSQL (.NET Core)
Marketing	+		+	+
Sales Enterprise	+		+	+
Sales Commerce	+		+	
Sales Team	+		+	
Service Enterprise	+		+	+
Customer Center	+		+	
Studio	+		+	+
Lending	+		+	
Bank Customer Journey	+		+	
Bank Sales	+		+	
Sales Enterprise & Marketing & Service Enterprise	+	+	+	+
Sales Enterprise & Marketing & Customer Center	+		+	
Sales Commerce & Marketing & Customer Center	+		+	
Sales Team & Marketing	+		+	
Sales Team & Marketing & Customer Center	+		+	+
Bank Sales & Bank Customer Journey & Lending & Marketing	+	+	+	+

Сервер кэширования Redis

Является частью логического уровня данных приложения и отвечает за решение следующих задач:

- Хранение данных пользователя и приложения (профиль пользователя, сессионные данные и т. п.).

- Хранение кэшированных данных.
- Обмен данными между узлами веб-фермы.

Для решения таких задач в архитектуре Creatio реализована **технология хранилищ данных**. В основе технологии — объектная модель классов, которая представляет собой унифицированный API для доступа из приложения к данным, расположенным во внешнем хранилище. В качестве внешнего хранилища Creatio использует сервер кэширования Redis.

Redis поддерживает следующие стратегии хранения данных:

- **Хранение данных только в памяти**. Персистентная база данных преобразуется в некий кэширующий сервер.
- **Периодическое сохранение данных на диск** (по умолчанию). Периодическое создание копий (snapshot) один раз в 1-15 минут в зависимости от времени создания предыдущей копии и количества измененных ключей.
- **Лог транзакций**. Синхронная запись каждого изменения в специальный append-only лог-файл.
- **Репликация**. Каждому серверу можно указать мастер-сервер, после чего все изменения на мастере будут воспроизводиться и на подчиненном сервере.

Способ хранения данных определяется конфигурированием сервера Redis.

Сервер системы контроля версий (опционально)

Является опциональным компонентом приложения. Используется, когда одновременно с эксплуатацией системы необходимо организовать разработку пользовательской конфигурации. Для **многопользовательской разработки** применение системы контроля версий является обязательным. Описание настройки сервера системы контроля версий содержится в статье Настроить систему управления версиями для среды разработки.

Сервер системы контроля версий отвечает за выполнение следующих функций:

- **Перенос изменений между приложениями при разработке**. Изменения переносятся с помощью пакетов, которые хранятся в виде набора файлов и папок на уровне файловой системы.
- **Хранение состояний конфигурации в виде пакетов определенной версии**. Система контроля версий хранит все конфигурационные элементы, которые разрабатываются в пакетах.

IDE Creatio настроена на работу с Subversion, но при разработке с использованием сторонних IDE могут использоваться и другие системы контроля версий.

Горизонтальное масштабирование

В Creatio существует возможность повысить производительность крупных проектов с помощью горизонтального масштабирования. Горизонтальное масштабирование используется для основного приложения Creatio с отказоустойчивостью.

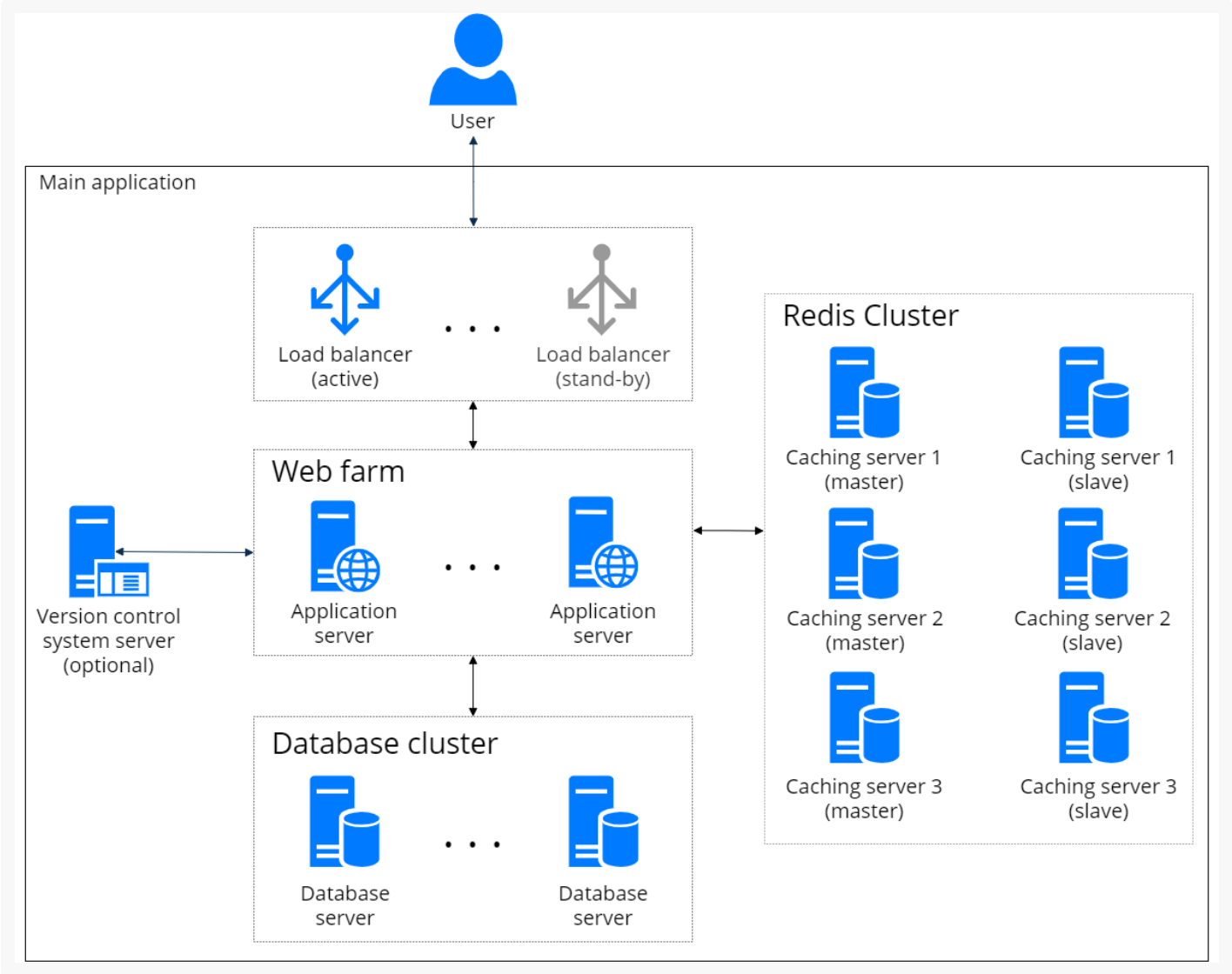
Компоненты инфраструктуры основного приложения с использованием горизонтального масштабирования:

- Балансировщики нагрузки. Балансировщик нагрузки может быть аппаратным или программным. Для работы в отказоустойчивом режиме используется балансировщик HTTP/HTTPS-трафика с поддержкой

протокола WebSocket. Описание установки и настройки балансировщика HAProxy содержится в статье Настроить горизонтальное масштабирование.

- Веб-ферма (несколько серверов приложений).
- Сервер кэширования с использованием механизма Redis Cluster (несколько серверов кэширования Redis).
- Сервер баз данных или кластер баз данных (несколько серверов баз данных).
- Сервер системы контроля версий (опциональный компонент).

Общая схема архитектуры основного приложения Creatio с использованием горизонтального масштабирования представлена на рисунке ниже.



Варианты развертывания приложения

Существуют следующие варианты развертывания приложения Creatio:

- On-site (развертывание приложения Creatio на локальных серверах клиента).
- Cloud (развертывание приложения Creatio в облаке).

Развертывание on-site

При развертывании приложения Creatio on-site все затраты, связанные с организацией серверной части (установка, настройка, сопровождение, администрирование), возлагаются на клиента.

Преимущества развертывания on-site:

- Высокая скорость и удобство разработки.
- Независимость сред разработки. Поскольку разработка ведется в отдельном приложении, исключено негативное влияние на других пользователей.
- Использование системы контроля версий для сохранения и переноса изменений.
- Возможность использовать IDE и настраивать процессы непрерывной интеграции.

Ограничения развертывания on-site:

- Необходимо обеспечить физическое наличие серверов для развертывания компонентов приложения.
- Требуется постоянное обновление, отладка, администрирование инфраструктуры.

При развертывании приложения on-site необходимо, чтобы серверы, на которых разворачиваются компоненты системы, а также клиентские компьютеры, на которых запускается система, отвечали определенным техническим требованиям. Для расчета параметров серверов, необходимых для развертывания приложения и контейнерных компонентов, воспользуйтесь калькулятором системных требований.

Описание этапов по развертыванию и настройке Creatio on-site на операционных системах Windows или Linux содержится в документации Развертывание on-site.

Развертывание cloud

При развертывании cloud приложение устанавливается на мощностях облачных дата-центров (Amazon, Azure), которые находятся под управлением компании Terrasoft. То есть физически вся серверная часть архитектуры и данные находятся в дата-центрах. Все вопросы, связанные с администрированием, быстродействием, масштабированием решаются силами сотрудников компании Terrasoft, а клиент использует только клиентскую часть приложения.

Преимущества развертывания cloud:

- Своевременное обновление.
- Максимально возможное быстродействие.
- Соответствие промышленным стандартам по доступности и защищенности данных.

Ограничения развертывания cloud:

- Наличие набора требований для развернутого cloud приложения Creatio.
- Невозможность использования сторонних IDE и СУБД.

Чтобы развернуть приложение Creatio cloud, попробуйте Тест-драйв на нашем официальном сайте. На протяжении 14-дневного пробного периода можно ознакомиться с основными возможностями приложения. По завершению пробного периода используемая демоверсия приложения может быть

перенесена на основную площадку Terrasoft.

Мобильное приложение

Основы

Мобильное приложение является инструментом для работы с данными приложения Creatio на мобильных устройствах. **Мобильное приложение Creatio** — это удаленное рабочее место, которое предоставляет быстрый доступ к данным клиентов, рабочему календарю, мобильной рассылке и т. д.

Мобильное приложение Creatio доступно для загрузки в App Store и Google Play на мобильные устройства, которые соответствуют требованиям.

Общие принципы работы

Преимущества использования мобильного приложения Creatio:

- Оперативный доступ и обмен информацией между сотрудником и офисом.
- Улучшение взаимодействия сотрудников и департаментов компании.
- Своевременное поступление информации.
- Быстрая реакция на входящую информацию.
- Повышение лояльности клиентов благодаря быстрой реакции.
- Повышение производительности сотрудников, которые работают «в полях».
- Доступ к информации даже при отсутствии интернет-соединения.

Мобильное приложение Creatio использует гибридный подход технической реализации. **Гибридное приложение** — это мобильное приложение, "упакованное" в native-оболочку. В отличие от мобильного native-приложения, является легко переносимым между различными платформами.

Перед началом работы с мобильным приложением необходимо в основном приложении выполнить первичную настройку.

Одним из этапов первичной настройки мобильного приложения является выбор режима работы. Реализован . Мобильное приложение Creatio поддерживает следующие режимы работы:

- **Гибридный режим**. Гибридный режим автоматически включается при отсутствии стабильного соединения с сервером Creatio. Этот режим позволяет создавать новые и редактировать существующие записи, работать с расписанием и с недавними записями раздела (10 записей), с которыми работал пользователь.
- **Online**. Для online-режима необходимо наличие интернет-соединения. При использовании этого режима пользователь работает напрямую с сервером Creatio, в качестве которого выступает основное приложение. Синхронизация конфигурационных изменений выполняется автоматически в режиме реального времени.
- **Offline**. Для offline-режима наличие интернет-соединения требуется только для первичного импорта и синхронизации. При использовании этого режима данные сохраняются локально на мобильном устройстве. Для получения конфигурационных изменений и актуализации данных необходимо вручную выполнять синхронизацию с сервером Creatio.

Синхронизация мобильного приложения с сервером Creatio выполняется с помощью сервиса работы с данными DataService. Если при синхронизации возникли конфликты, то информация о них отобразится в журнале синхронизации, который доступен в гибридном и offline режимах.

Для проверки корректности работы пользовательской функциональности, необходимо выполнить отладку мобильного приложения.

Схема работы

Мобильное приложение Creatio представляет собой набор модулей, необходимых для синхронизации с сервером Creatio. Схема работы мобильного приложения представлена ниже.

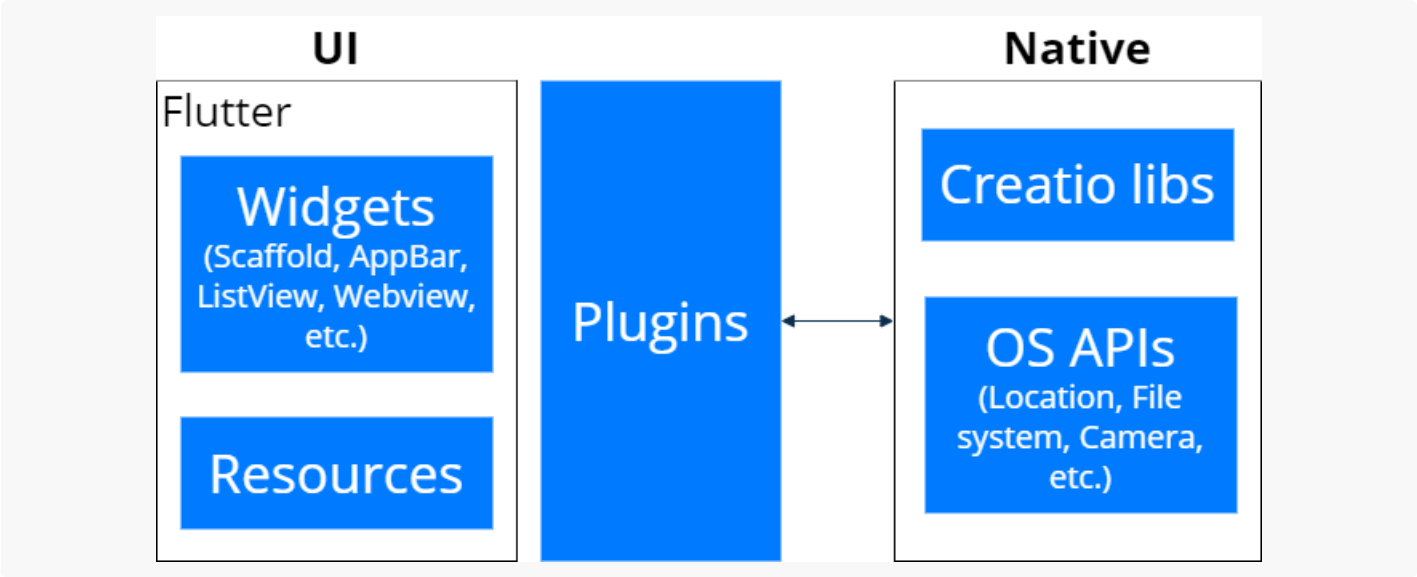
Каждый продукт и отдельно взятый сайт клиента может содержать следующие пользовательские **элементы**:

- Набор настроек мобильного приложения Creatio.
- Логику работы.
- Визуальный интерфейс.

Пользователю мобильного приложения Creatio необходимо установить его и выполнить синхронизацию с основным приложением.

Схема архитектуры

Схема архитектуры мобильного приложения Creatio представлена ниже.



Для создания гибридных приложений мобильное приложение Creatio использует фреймворк Apache Cordova, который предоставляет следующие **возможности**:

- Доступ к программному интерфейсу мобильного устройства (API) для взаимодействия с базой данных или оборудованием (например, камерой или картой памяти).
- Native-плагины для работы с API разных мобильных платформ (iOS, Android, Windows Phone и др.). Кроме того, разработка пользовательских плагинов позволяет добавлять функциональность и расширять API. Перечень доступных платформ и функциональность базовых native-плагинов Cordova содержится в документации Cordova.

Ядро мобильного приложения Creatio предоставляет унифицированный интерфейс для взаимодействия клиентских частей приложения. Используемые ядром JavaScript-файлы условно можно разделить на базовые и конфигурационные.

Базовые скрипты содержатся в сборке приложения, публикуемой в магазине приложений.
Компоненты базовых скриптов:

- MVC-компоненты (представления страниц, контроллеры, модели).
- Модули синхронизации (импорт и экспорт данных, импорт метаданных, импорт файлов и т. д.).
- Клиентские классы веб-сервисов.
- Классы, предоставляющие доступ к native-плагинам.

Конфигурационные файлы, полученные при синхронизации с сервером Creatio, приложение получает и локально сохраняет в файловой системе устройства. **Компоненты** конфигурационных файлов:

- Манифест мобильного приложения.
- Схемы разделов.
- Настройки разделов.

Манифест — это конфигурационный объект, свойства которого описывают структуру (объекты и связи между ними) мобильного приложения. **Группы свойств** манифеста мобильного приложения Creatio:

- Свойства интерфейса приложения (формирование разделов приложения, главного меню, настройка

пользовательских изображений).

- Свойства данных и бизнес-логики (описание импортируемых данных и пользовательская бизнес-логика обработки этих данных в мобильном приложении).
- Свойства синхронизации приложений (настройка синхронизации данных с основным приложением).

В разделе [*Визы*] ([*Approvals*]) мобильного приложения используется Flutter Framework.

Совместимость с продуктами Creatio

Мобильное приложение является частью платформы и доступно для пользователей основного приложения Creatio версий 7.15 и выше.

После установки приложения на мобильное устройство пользователь, указав параметры соединения с сервером Creatio, получает метаданные (структура приложения, системные данные) и данные от сервера. Эта схема работы позволяет мобильному приложению быть совместимым со **всеми продуктами Creatio**.

На заметку. Мобильное приложение не может использоваться пользователями портала.

Разработка приложений на платформе Creatio

Основы

Creatio — это low-code платформа, которая предоставляет широкие возможности для ускорения разработки, внедрения и масштабирования приложений. Платформа построена на открытых принципах кастомизации. Это позволяет создавать приложения разработчикам с разным уровнем квалификации — от бизнес-аналитика до разработчика полного цикла. В зависимости от сложности или типа бизнес-задачи, разработка приложений на платформе Creatio предполагает разные уровни кастомизации.

Уровни кастомизации Creatio

С точки зрения логических уровней взаимодействия архитектура платформы Creatio может быть представлена следующим образом:

При разработке приложений в Creatio необходимо учитывать, что уровень ядра — неизменяемый компонент платформы, а разработка реализуется на уровне конфигурации и базы данных.

База данных

База данных — это уровень физического хранения данных. В базе хранятся не только пользовательские данные, но и настройки приложения, а также настройки прав доступа к приложению.

Платформа Creatio предоставляет инструменты для работы с данными непосредственно из интерфейса приложения. Поэтому не возникает необходимости работы с объектами базы данных напрямую.

Существуют задачи, которые логичнее и быстрее реализуются на уровне базы данных. Пользовательскую бизнес-логику можно реализовать на уровне базы данных с помощью представлений и хранимых процедур. Затем реализованную бизнес-логику можно вызывать в пользовательских конфигурационных элементах.

Ядро

Ядро — неизменяемая часть платформы, которая представляет собой набор библиотек с реализацией базовой функциональности приложения.

Back-end библиотеки реализованы на языке C# с использованием классов платформы .NET Framework. У разработчика есть возможность создавать экземпляры back-end классов и использовать функциональность back-end библиотек, но при этом нельзя вносить изменения в эти классы и библиотеки.

Основные **back-end компоненты ядра**:

- ORM-модель данных и методы работы с ней. В большинстве случаев для доступа к данным рекомендуется использовать именно объектную модель, хотя прямой доступ к базе данных также реализован в back-end компонентах ядра.
- Пакеты и механизм замещения.
- Библиотеки серверных элементов управления. К таким элементам управления относятся страницы, построенные на технологии ASP.NET, которые формируются на сервере, например, страницы раздела [*Конфигурация*] ([*Configuration*]).
- Системные веб-сервисы.

- Функциональность основных дизайнеров и системных разделов.
- Библиотеки для интеграции с внешними сервисами.
- Движок бизнес-процессов (`ProcessEngineService.svc`). Это элемент платформы, который позволяет выполнять алгоритмы, настроенные в виде диаграмм.

Front-end классы ядра реализованы на языке JavaScript с использованием различных фреймворков. Они предназначены для создания пользовательского интерфейса и реализации других бизнес-задач на стороне браузера. Основная **задача** front-end компонентов ядра — обеспечение работы клиентских модулей.

Основные **front-end компоненты ядра**:

- Внешние библиотеки клиентских фреймворков.
- Песочница (sandbox) — специальный компонент клиентского ядра, предназначенный для обеспечения взаимодействия между клиентскими модулями путем обмена сообщениями.
- Базовые модули — файлы на языке JavaScript, в которых реализована функциональность основных объектов платформы.

Конфигурация

Конфигурация — это набор функциональности, который доступен пользователям конкретного рабочего пространства, а именно:

- Серверная логика.
- Автогенерируемые классы, являющиеся продуктом работы настроек приложения.
- Клиентская логика (страницы, кнопки, действия, отчеты, бизнес-процессы и другие настраиваемые конфигурационные элементы).

Конфигурация является легко изменяемой частью приложения. Конкретную конфигурацию формируют следующие типы элементов:

- **Объекты** — сущности, предназначенные для хранения данных, которые объединяют таблицу в базе данных и класс на серверной стороне.
- **Бизнес-процессы** — настраиваемые элементы, которые представляют собой визуальный алгоритм пользовательских действий.
- **Клиентские модули**.

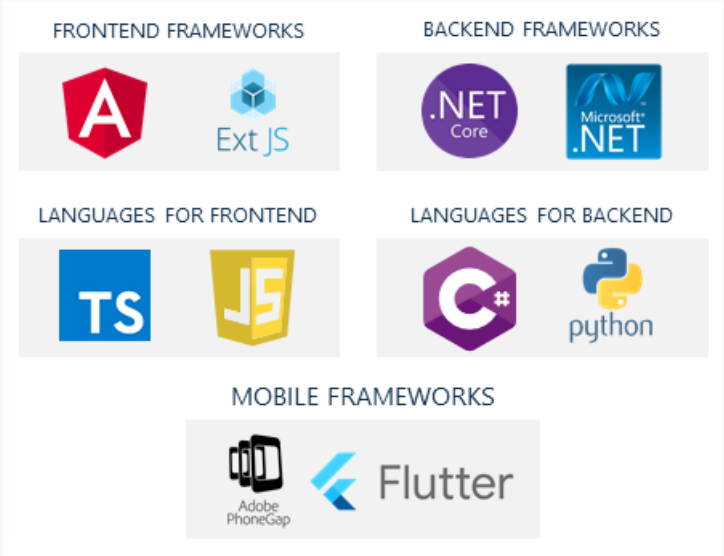
Инструменты разработки приложений

Creatio предоставляет широкий спектр инструментов для создания новых приложений и расширения существующих.

Стек open-source технологий

Поддержка стандартных языков программирования и фреймворков ускоряет разработку, обеспечивает эффективную поддержку и разработку компонентов приложения, а также гарантирует наличие квалифицированных специалистов с необходимым стеком технологий.

Платформа Creatio поддерживает следующие **технологии**:



Для реализации сложной бизнес-логики, интеграции и настройки, Creatio предоставляет **встроенную IDE**, инструменты которой позволяют ускорить решение типовых задач по конфигурированию платформы. Встроенная IDE позволяет использовать C# или JavaScript для реализации следующих **задач**:

- Расширение и изменение функций Creatio.
- Организация взаимодействия с системами контроля версий.
- Передача изменений между средой разработки, предпромышленной и промышленной средами.

Для кастомизации Creatio разработчики могут использовать **стороннюю IDE** (например, Microsoft Visual Studio), которая позволяет работать с проектами в локальной файловой системе. Это ускоряет и упрощает процесс разработки за счет использования знакомой IDE, наличия множества плагинов, расширений и интеграций с различными инструментами разработки, системами контроля версий и т. д. Также позволяет разработчикам сэкономить время на изучении новых инструментов и сосредоточиться на разработке кода C# и JavaScript в знакомой IDE.

Low-code/no-code разработка

Инструменты low-code/no-code разработки представляют собой набор визуальных drag-and-drop редакторов пользовательского интерфейса. Они позволяют выполнять следующие **задачи**:

- Автоматизировать бизнес-процессы и динамические кейсы.
- Моделировать структуру данных.
- Моделировать пользовательский интерфейс для веб- и мобильных приложений.
- Создавать информационные панели и отчеты.
- Настраивать интеграции.
- Строить прогнозные модели и т. д.

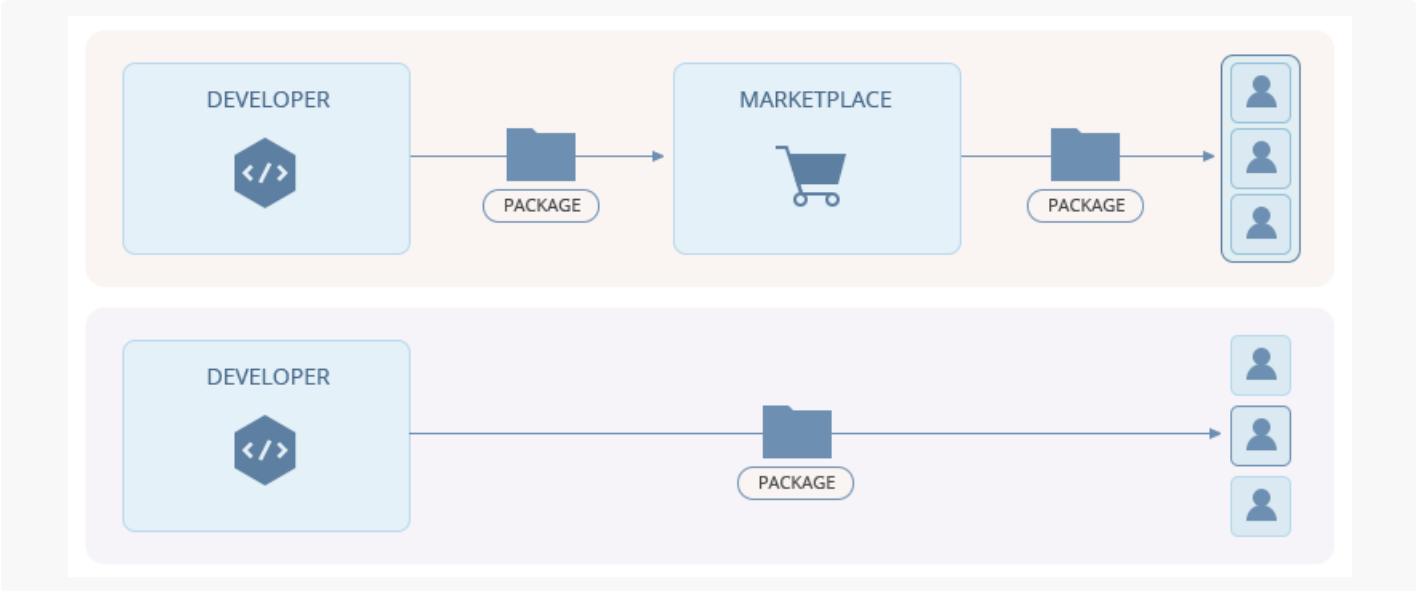
Большинство вариантов использования покрываются low-code инструментами. Минимальное ручное

кодирование, а также легкодоступные встроенные инструменты сделали разработку приложений доступной для бизнес-экспертов, опытных пользователей, аналитиков и citizen developers. Описание no-code инструментов содержится в статье Low-code/no-code.

Механизм пакетов

Независимо от инструмента, используемого для разработки приложения Creatio, все кастомизации упаковываются в пакеты, которые представляют собой ключевой компонент архитектуры Creatio. Пакетная архитектура позволяет выделять целостные функциональные блоки и оформлять их в виде отдельных модулей, управлять иерархией пакетов и версионностью. Это дает возможность быстро расширять конфигурацию, а также переносить изменения между средой разработки, предпромышленной и промышленной средами. Этот же механизм лежит в основе решений, публикуемых на площадке Marketplace.

Любой продукт Creatio — это набор пакетов, которые устанавливаются поверх ядра Creatio. Пакет содержит схемы объектов, исходный код, бизнес-процессы, отчеты и т. д. Он также может содержать сторонние сборки, SQL-сценарии, системные настройки и пользовательские данные.



Модель расширения Creatio основана на **принципе открытости-закрытости**, при котором основная логика приложения закрыта для прямых манипуляций, но открыта для расширения и модификации путем добавления пользовательских пакетов. Каждый пакет может быть расширен пакетом другого издателя (партнера-интегратора, разработчика Marketplace или заказчика). Это позволяет платформе эффективно совмещать out-of-the box продукты, рыночные решения и настройки клиентов практически в любой комбинации.

Архитектура пакетов — это основной механизм доставки и развертывания пользовательских приложений, расширений и шаблонов, полностью интегрированный в Creatio Marketplace. Creatio Marketplace представляет собой экосистему для разработки, распространения и получения кастомизаций — как пользовательских приложений и шаблонов, так и обновлений и изменений отраслевых приложений.

Пакеты не влияют на логику ядра платформы, обеспечивая беспроблемную доставку обновлений для ядра Creatio, а также параллельное развертывание и обновление настраиваемых функций.

Независимая от СУБД архитектура и собственная ORM

Creatio — это **СУБД-независимая платформа**, в основе которой лежит ORM, разработанная Creatio. Это позволяет разработчикам легко создавать и доставлять пользовательские приложения для различных конфигураций в базе данных Oracle, PostgreSQL или MS SQL Server без изменений в кодовой базе.

Бизнес-логика, основанная на процессах

Это среда для взаимодействия между разработчиками полного цикла и бизнес-аналитиками. Используя визуальные drag-and-drop редакторы бизнес-процессов, пользователи могут создавать собственную бизнес-логику, просто нарисовав необходимый процесс. Работа с бизнес-процессами описана в блоке статей Бизнес-процессы.

Интеграции

Платформа Creatio предоставляет необходимые инструменты для интеграции со сторонними системами и приложениями, включая поддержку REST API, протокола OData, SOAP-сервисов, возможности аутентификации по протоколам OAuth и LDAP. Интеграции можно разрабатывать как часть приложения Creatio или стороннего приложения. Сложные инструменты обеспечивают безопасность данных во время интеграции для идентификации и контроля доступа, а также управления структурой пользователей.

Low-code/no-code

Основы

Платформа Creatio предоставляет широкие возможности low-code/no-code кастомизации: от конфигурирования существующих приложений до разработки пользовательских бизнес-решений.

Low-code/no-code платформы ориентированы на непрофессиональных разработчиков (citizen developers) и используют визуальные интерфейсы с простой логикой и функциями drag-and-drop. **Встроенные инструменты** платформы Creatio дают возможность пользователям, не обладающим знаниями языков программирования или процессов разработки программного обеспечения, создавать свои приложения.

Преимущества low-code/no-code платформ:

- Быстрая разработка приложений.
- Быстрое развертывание.
- Исполнение и управление приложениями.
- Декларативная, высокоуровневая разработка.
- Возможность моделирования данных, разработки интерфейса и бизнес-логики.
- Пользовательское конфигурирование приложений.

Low-code/no-code инструменты

В Creatio реализованы следующие **low-code/no-code инструменты**:

- Дизайнер процессов.

- Мастер разделов.
- Дизайнер кейсов.
- Встроенные инструменты интеграции.
- Технологии машинного обучения.

Дизайнер процессов

Дизайнер процессов — это визуальный редактор для разработки исполняемых бизнес-процессов на основе нотации BPMN 2.0. Исполняемые бизнес-процессы позволяют реализовать пользовательскую бизнес-логику: от автоматизации рутинных задач до разработки сложных интеграций. Дизайнер процессов поддерживает **многопользовательскую разработку**. Использование встроенных элементов бизнес-процессов позволяет выполнять следующие **задачи**:

- Планировать пользовательские действия.
- Работать с интерфейсными страницами.
- Обрабатывать данные.
- Вызывать внешние веб-сервисы и т. д.

Назначение интерфейса настройки и встроенных инструментов валидации:

- Проектирование схемы бизнес-процесса.
- Изменение схемы бизнес-процесса.
- Отладка схемы бизнес-процесса с учетом нюансов выполнения.

Описание работы с бизнес-процессами в Creatio содержится в блоке статей Бизнес-процессы.

Мастер разделов

Мастер разделов позволяет выполнять следующие **задачи**:

- Настраивать пользовательский интерфейс.
- Добавлять и настраивать бизнес-логику.
- Создавать и настраивать разделы, страницы, детали, вкладки и мини-карточки.
- Добавлять поля, изменять их положение или скрывать.

Описание работы с мастером разделов содержится в блоке статей Настройка интерфейса и бизнес-логики.

Дизайнер кейсов

Дизайнер кейсов построен на основе ProcessEngineService и позволяет автоматизировать неструктурированные процессы, ход которых определяется динамически, в соответствии с принятой бизнес-логикой. Бизнес-кейсы состоят из стадий, каждая из которых может включать в себя ряд последовательных или параллельных действий — "шагов", как автоматических, так и пользовательских действий. No-code дизайнер кейсов позволяет выполнять следующие **задачи**:

- Изменять последовательность шагов на стадии процесса.

- Перемещать шаги на другие стадии процесса.
- Изменять последовательность стадий процесса с помощью drag-and-drop.

Встроенные инструменты интеграции

Creatio может интегрироваться с настраиваемыми веб-службами REST только с помощью low-code инструментов. Расширенные возможности интеграции (с использованием инструментов .Net, REST, SOAP, OData, открытого API), а также система администрирования и контроля доступа позволяют быстро и безопасно встраивать Creatio в информационную среду предприятия. На основе настраиваемой бизнес-логики Creatio выполняет следующую **последовательность действий**:

1. Генерирует запрос.
2. Отправляет запрос веб-сервису.
3. Получает ответ.
4. Извлекает необходимые данные.

Данные, полученные от веб-сервиса, можно использовать для создания или обновления записей в базе данных Creatio, а также для реализации специальной бизнес-логики или автоматизации.

Описание способов интеграции содержится в статье Интеграции.

Инструменты машинного обучения

Технологии машинного обучения позволяют автоматизировать процесс принятия решений путем анализа исторических данных и выявления зависимостей между большими объемами информации. В Creatio реализованы следующие **модели** машинного обучения:

- Прогнозирование справочных полей — позволяет автоматически заполнять поле на основании данных текущей записи и решений, принятых пользователями ранее в аналогичных ситуациях.
- Прогнозирование числовых полей — позволяет рассчитать прогнозное значение числового поля.
- Прогнозирование рейтинга записей — позволяет выполнять скоринг (прогнозирование рейтинга) записей в любом разделе системы.
- Рекомендательное прогнозирование — позволяет создавать подборки записей из объектов системы и предлагать их клиентам или партнерам.

Приложение Studio free

Основы

Studio free — это приложение для совместной работы над моделированием бизнес-процессов, которое напрямую взаимодействует с веб-клиентом. Бизнес-процессы в Studio free основываются на нотации BPMN 2.0, которая широко используется среди специалистов по автоматизации бизнеса, от аналитиков до разработчиков.

Общие принципы работы

Свойства приложения Studio free:

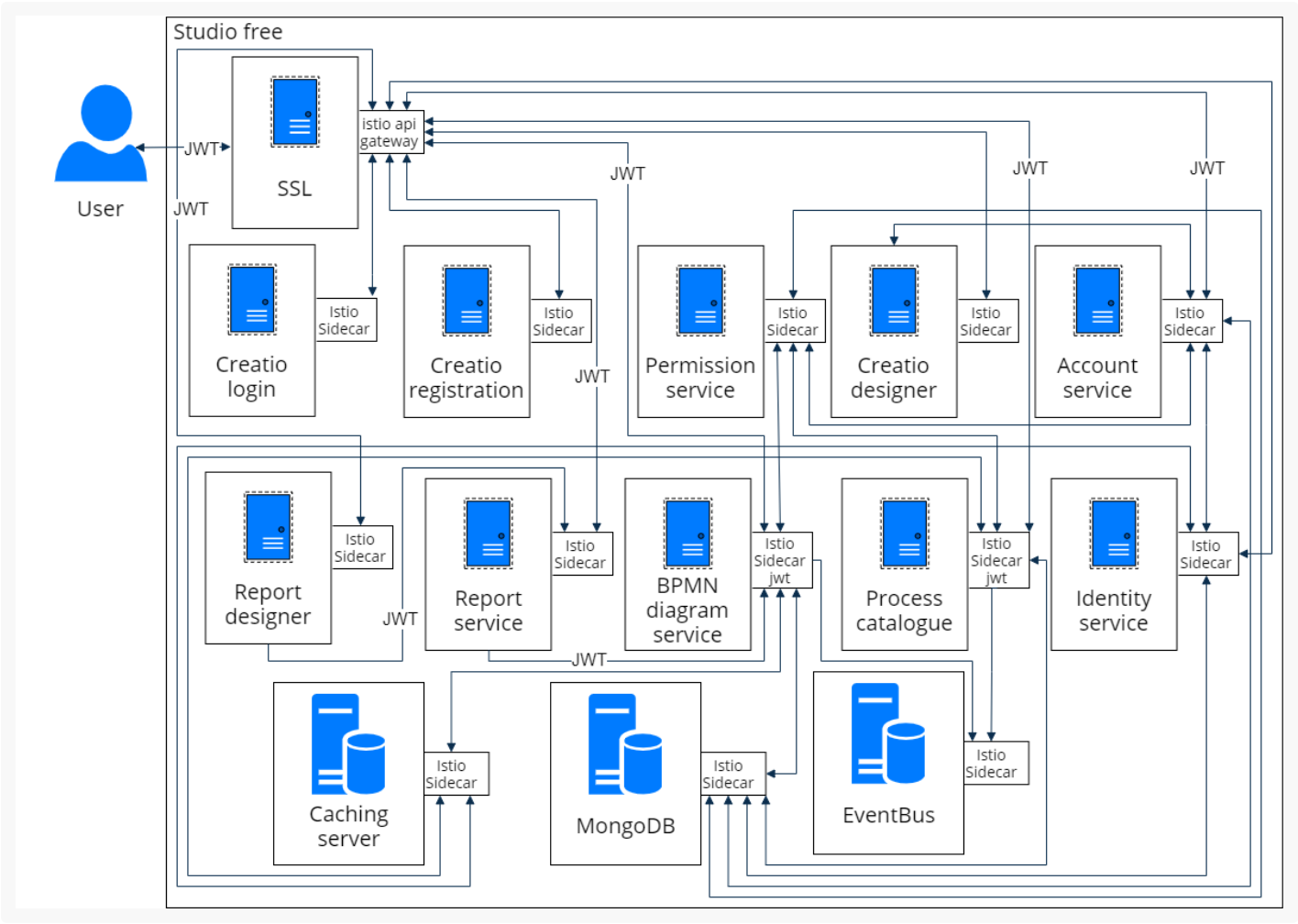
- **Сокращение времени на построение процессов**. Создавайте диаграммы процессов с помощью визуальных инструментов и сохраняйте их в библиотеке процессов. Настраиваемая структура, навигация и единая среда управления помогут упорядочить бизнес-процессы компании быстро и с учетом необходимых требований.
- **Стандартизация управления бизнес-процессами**. Мультифункциональный дизайнер процессов обеспечит соответствие требованиям нотации BPMN 2.0. Это позволяет моделировать процессы, которые будут понятны заказчикам, бизнес-аналитикам, разработчикам и конечным пользователям. А благодаря поддержке формата *.bpmn доступен импорт процессов и их экспорт в дизайнер процессов Creatio и другие системы.
- **Организация совместной работы команды**. Привлекайте к работе коллег или сторонних экспертов, совместно редактируйте процессы в режиме реального времени, делитесь ссылками на процессы для удаленного просмотра и комментирования.
- **Упрощение документирования процессов**. Фиксируйте информацию и добавляйте необходимые пояснения параллельно с построением диаграммы. Система позволит выгрузить описание процесса в формате *.pdf для получения целостной документации по процессу и ее использования вне системы.

Инструменты приложения Studio free:

- Дизайнер процессов. Дизайнер процессов с поддержкой нотации BPMN 2.0 предоставляет набор инструментов для управления сложными структурированными бизнес-процессами в визуальном редакторе.
- Библиотека процессов. Все бизнес-процессы компании хранятся в библиотеке процессов, которая позволяет выполнять следующие **действия**:
  - Добавлять новые процессы в библиотеку из дизайнера или при помощи импорта.
  - Организовывать бизнес-процессы в иерархическую структуру.
  - Искать процессы и папки.

Схема работы

Схема работы приложения Studio free представлена ниже.



Совместимость с продуктами Creatio

Функциональность приложения Studio free доступна как отдельный продукт, а также совместима со **всеми продуктами Creatio** путем экспорта (формат *.bpmn, *.svg или *.png) и импорта (формат *.bpmn) бизнес-процессов.

Варианты развертывания

Приложение Studio free доступно только в облаке на площадке Террасофт, не имеет открытого API и возможностей доработки клиентами.

Back-end (C#)

Основы

Разработка решений в Creatio предполагает разные уровни кастомизации в зависимости от сложности или типа бизнес-задачи. При этом следует учитывать, что уровень ядра является неизменяемым компонентом системы, а разработка в Creatio реализуется на уровне конфигурации.

Кастомизация приложений на уровне back-end реализуется в направлениях работы с данными, веб-

сервисами, настройкой бизнес-логики объектов и т. д.

Для этих целей Creatio предоставляет набор различных инструментов — от разработки схемы [*Исходный код*] ([*Source code*]) или [*Действие процесса*] ([*User Task*]) во встроенной IDE до создания завершенных пользовательских проектов, подключаемых к приложению в качестве внешних библиотек.

Направления back-end разработки

На уровне back-end реализуются следующие направления разработки:

- ORM-модель данных и методы работы с ней.
- Реализация прямого доступа к базе данных.
- Создание и использование системных веб-сервисов.
- Настройка интеграции с внешними сервисами.
- Работа с компонентами системы и микросервисами.
- Расширенная настройка бизнес-процессов и встроенных процессов объектов приложения.
- Разработка бизнес-логики объектов.

ORM-модель данных и прямой доступ к базе данных

ORM-модель данных реализована в системе классами пространства имен `Terrasoft.Core.Entities` :

- `Terrasoft.Core.Entities.EntitySchemaQuery` — построение запросов выборки записей из таблиц базы данных с учетом прав доступа текущего пользователя.
- `Terrasoft.Core.Entities.Entity` — доступ к объекту, который представляет собой запись в таблице базы данных.

В большинстве случаев для доступа к данным рекомендуется использовать именно объектную модель, хотя прямой доступ к базе данных также реализован в back-end компонентах ядра.

Прямой доступ к базе данных предоставляет группа классов back-end ядра приложения из пространства имен `Terrasoft.Core.DB`:

- `Terrasoft.Core.DB.Select` — построение запросов выборки записей из таблиц базы данных.
- `Terrasoft.Core.DB.Insert` — построение запросов на добавление записей в таблицы базы данных.
- `Terrasoft.Core.DB.InsertSelect` — построение запросов на добавление записей в таблицы базы данных.
- `Terrasoft.Core.DB.Update` — построение запросов на изменение записей в таблице базы данных.
- `Terrasoft.Core.DB.Delete` — построение запросов на удаление записей в таблице базы данных.
- `Terrasoft.Core.DB.DBExecutor` — возможность создания и выполнения сложных запросов (например, с несколькими вложенными фильтрациями, различными комбинациями join-ов и т. д.) к базе данных.

Веб-сервисы

Сервисная модель Creatio предоставляет базовый набор веб-сервисов, с помощью которых может быть организована интеграция Creatio с внешними приложениями и системами.

Примеры **системных сервисов**:

- odata — обмен данными с Creatio по протоколу OData 4.
- ProcessEngineService.svc — запуск бизнес-процессов Creatio из внешних приложений.

Кроме системных, в Creatio реализованы **конфигурационные веб-сервисы**, предназначенные для вызова из клиентской части приложения.

Дополнительно к базовым сервисам разработчик имеет возможность создать **пользовательский веб-сервис**, предназначенный для решения узких бизнес-задач конкретного проекта.

Внешние сервисы

В ядре приложения реализованы классы, обеспечивающие интеграцию с внешними сервисами. Например, пространство имен `Terrasoft.Social` предоставляет возможность интеграции с различными социальными сетями.

Компоненты системы и микросервисы

В ядре приложения реализованы классы, предоставляющие возможности для работы с компонентами системы и микросервисами. Например, пространство имен `Terrasoft.Sync` ядра приложения предоставляет классы встроенного механизма синхронизации с внешними хранилищами данных (Sync Engine), который позволяет создавать, изменять и удалять `Entity` в системе на основании данных из внешних систем и экспортировать данные во внешние системы. Процесс синхронизации осуществляется с помощью класса `SyncAgent`.

Бизнес-процессы и встроенные процессы объектов

Back-end разработка может потребоваться для **настройки сложных бизнес-процессов** (Элемент процесса [*Задание-сценарий*] ([*Script-task*])) либо для создания пользовательских **повторяющихся операций бизнес-процесса** (Конфигурационная схема [*Действие процесса*] ([*User Task*])).

Встроенные процессы объекта могут быть настроены как при помощи no-code инструментов, так и с использованием back-end разработки. Использование программного кода позволяет более гибко настроить поведение объекта в случае наступления определенных событий.

Бизнес-логика объектов

В Creatio есть возможность разрабатывать бизнес-логику объектов без использования событийных подпроцессов. Для этого достаточно создать класс-наследник базового ядрового класса `Terrasoft.Core.Entities.Events.BaseEntityEventListener` и декорировать его атрибутом `EntityEventListener` с указанием имени сущности, для которой необходимо выполнить подписку событий. После чего можно переопределять методы-обработчики нужных событий.

Инструменты и утилитные возможности back-end разработки

Схема исходного кода

Основной возможностью back-end разработки является создание в пользовательском пакете схемы типа [*Исходный код*] ([*Source code*]).

Все изменения схемы, выполняемые в дизайнере, осуществляются в оперативной памяти. Чтобы изменения были сохранены на уровне метаданных схемы, схему следует сохранить. Для того чтобы изменения произошли на уровне базы данных, схему необходимо опубликовать.

Разработку схемы типа [*Исходный код*] ([*Source code*]) можно вести в файловой системе с использованием удобной IDE.

Конфигурирование бизнес-процессов

Включение специфической back-end логики в бизнес-процесс возможно с помощью программного кода, выполняемого элементом [*Задание-сценарий*] ([*Script-task*]).

Во время работы с бизнес-процессами в Creatio часто возникает необходимость выполнять однотипные операции. Для этих целей лучше всего подходит элемент [*Выполнить действие процесса*] ([*User task*]), для которого существует возможность выбрать наиболее подходящий в конкретной ситуации тип действия — [*Пользовательское действие*] ([*User task*]).

По умолчанию в системе доступно множество пользовательских действий, однако могут возникнуть ситуации, когда для выполнения определенного бизнес-процесса необходимо создать новое пользовательское действие.

Создать новое пользовательское действие можно с помощью конфигурационной схемы [*Действие процесса*] ([*User Task*]). В простой реализации действие процесса частично повторяет логику элемента процесса [*Задание-сценарий*] ([*Script-task*]), однако созданное однажды действие можно использовать многократно в разных процессах. А при внесении изменений в действие процесса, такие изменения сразу же будут применены ко всем процессам, в которых это действие использовалось.

Внешние библиотеки

В структуру пользовательского пакета могут быть включены внешние библиотеки, созданные пользователем. Это позволяет реализовать сложную логику, механизмы наследования, инкапсуляции при разработке специфического проектного решения.

Пакет-проект

Одним из инструментов Creatio для ускорения back-end разработки приложения является пакет-проект. Это пакет, который позволяет разрабатывать функциональность как обычный C#-проект. Новая функциональность в виде скомпилированной библиотеки и *.cs-файлов включается в файловый контент пакета. При старте или перезапуске приложения Creatio собирает информацию о том, что в пакетах есть подготовленные библиотеки и сразу же подключает их в приложение.

Сервис глобального поиска

Основы

Сервис глобального поиска (Global Search Service) создан для интеграции поисковой системы

ElasticSearch с приложением Creatio. **Назначение** сервиса — поиск данных в приложении Creatio из командной строки. Поиск данных выполняется по всем разделам приложения, включая пользовательские, независимо от места запуска поиска — из главного меню либо из раздела.

Общие принципы работы

Задачи сервиса глобального поиска:

- Подписка клиента — создание индекса в ElasticSearch и сохранение связи индекс-приложение.
- Отключение клиента — удаление по требованию индекса в ElasticSearch.
- Участие в процессе индексации — получение информации из базы данных.

Особенности сервиса глобального поиска:

- Поиск выполняется по всем полям записи — как текстовым так и справочным, а также по деталям [*Адреса*] ([*Addresses*]), [*Средства связи*] ([*Communication options*]) и [*Платежные реквизиты*] ([*Banking details*]).
- Файлы и ссылки, прикрепленные на вкладке [*Файлы и примечания*] ([*Attachments and notes*]) страницы записи, могут быть найдены по названию либо по описанию.
- Поисковые запросы обрабатываются с учетом распространенных опечаток и морфологии — учитываются разные формы слов, введенных в строку поиска. Поисковый запрос не чувствителен к регистру.
- Результаты поиска отображаются по релевантности, как для всего списка результатов, так и при фильтрации найденных записей. Например, если поиск выполнялся из раздела, то в начале списка будут отображены записи этого раздела.
- Если у пользователя нет прав доступа на определенную колонку объекта, то такая колонка не отображается на странице результатов глобального поиска.

Параметры поиска задаются при помощи системных настроек:

- [*Вес объекта по умолчанию для глобального поиска*] (код [*GlobalSearchDefaultEntityWeight*]) и [*Вес первичной колонки по умолчанию для глобального поиска*] (код [*GlobalSearchDefaultPrimaryColumnWeight*]) — настройка правил отображения результатов поиска.
- [*Отображать результаты поиска по частичному совпадению*] (код [*UseInexactGlobalSearch*]) — отображение в результатах поиска данных, которые найдены с учетом опечаток и морфологии.
- [*Доля совпадения для отображения в результатах поиска, %*] (код [*GlobalSearchShouldMatchPercent*]) — регулирование выдачи результатов поиска по частичному совпадению и повышение вероятности найти данные при неточном поисковом запросе.

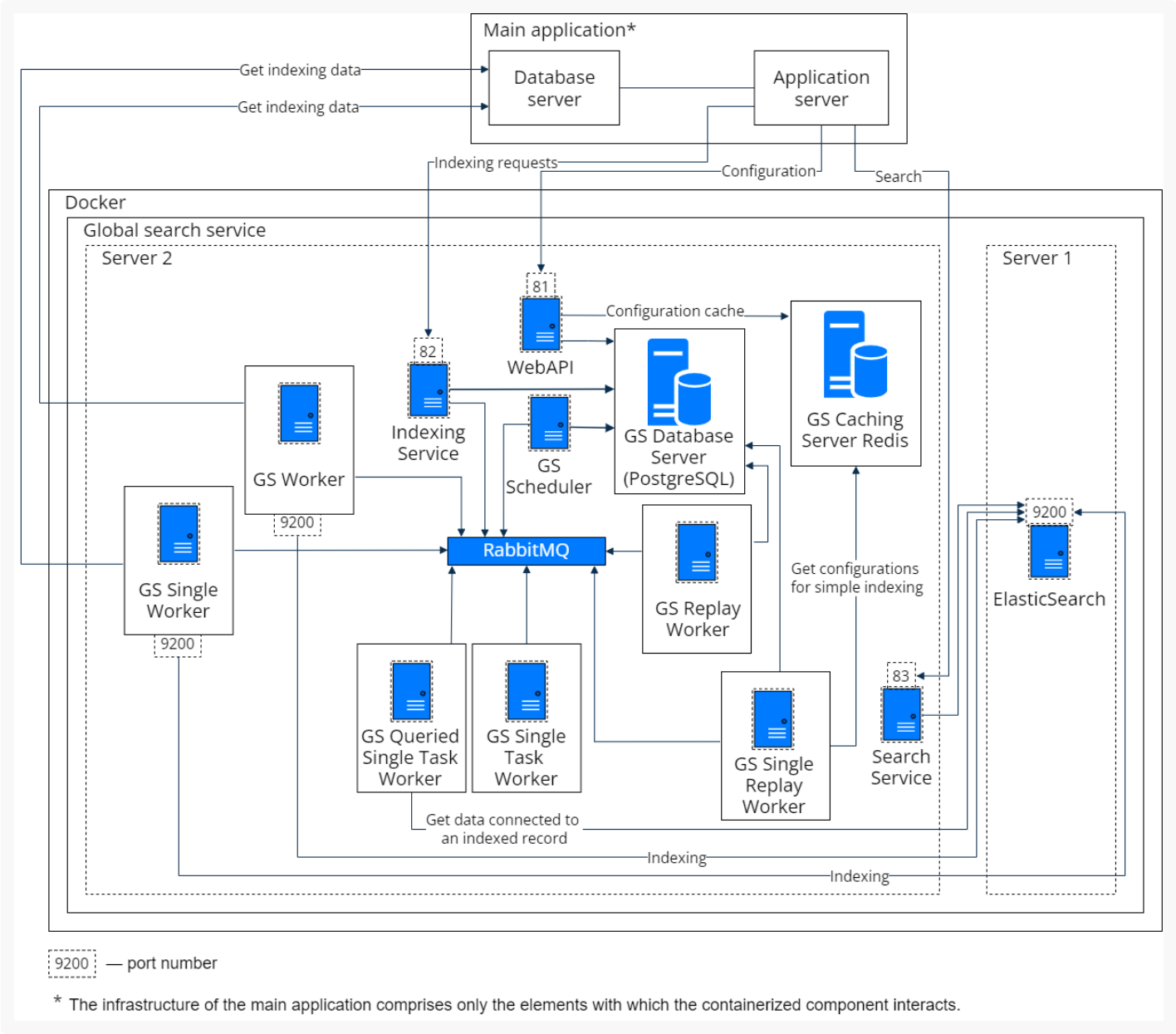
Схема работы

Компоненты сервиса глобального поиска:

- RabbitMQ — брокер сообщений.
- ElasticSearch — поисковая система.
- GS Database Server — база данных для конфигурирования компонентов глобального поиска.

- GS Caching Server Redis — хранилище данных, которое используется для кэширования и быстродействия.
- WebAPI — веб-сервис для конфигурирования компонентов глобального поиска.
- Indexing Service — веб-сервис для обработки запросов точечного индексирования данных системы.
- GS Scheduler — планировщик задач индексации данных из Creatio в ElasticSearch.
- GS Worker — компонент для индексирования данных из Creatio в ElasticSearch по задачам компонента GS Scheduler.
- GS Replay Worker — компонент для обработки результатов индексации (результаты работы компонента GS Worker).
- GS Single Worker — компонент для точечной индексации данных бизнес-процессов в ElasticSearch по запросу бизнес-процесса.
- GS Single Replay Worker — компонент для обработки исключений в процессе точечной индексации (результаты работы компонента GS Single Worker).
- GS Single Task Worker — компонент для постановки задач компоненту GS Single Worker.
- GS Queried Single Task Worker — компонент для формирования задач компоненту GS Single Worker.

Схема работы сервиса глобального поиска представлена ниже.



Масштабируемость

Использование кластера баз данных позволяет выполнить масштабирование сервиса глобального поиска в крупных проектах. Описание кластеризации содержится в документации ElasticSearch.

Совместимость с продуктами Creatio

Сервис глобального поиска имеет версии 1.4, 1.5, 1.6, 1.7, 2.0, которые совместимы со **всеми продуктами Creatio** версий 7.10 и выше.

Варианты развертывания

Сервис глобального поиска разворачивается on-site и cloud.

Для приложений, развернутых on-site, необходимо выполнить предварительную настройку сервиса. Для

настройки необходимы 2 сервера (физические или виртуальные машины), которые должны отвечать техническим требованиям. На серверах должна быть установлена операционная система Linux, которая официально поддерживает программное обеспечение Docker. Перечень операционных систем, которые поддерживает Docker, содержится в документации Docker.

Рекомендуем устанавливать актуальную версию сервиса глобального поиска.

Front-end (JS)

Основы

Front-end платформы — это набор модулей, которые определяются и загружаются асинхронно по мере необходимости.

Основой front-end Creatio является ядро.

Уровень ядра предоставляет:

- возможность использования объектно-ориентированного подхода и механизма наследования;
- инструменты для определения и асинхронной загрузки модулей и их зависимостей;
- функциональность базовых элементов системы;
- механизм взаимодействия модулей.

Front-end разработка в Creatio осуществляется на уровне конфигурации и представляет собой создание новых и расширение базовых визуальных и невизуальных модулей, схем моделей представления.

Компоненты front-end ядра приложения

Внешние JS-библиотеки

ExtJS — JavaScript-фреймворк для разработки веб-приложений и пользовательских интерфейсов. В Creatio ExtJS используется как механизм создания структуры классов клиентской части ядра. Позволяет реализовать объектно-ориентированный подход, который в чистом виде не реализован в JavaScript. Предоставляет возможность создавать классы, реализовывать иерархию наследования, группировать классы в пространства имен.

RequireJS — библиотека, которая реализует подход Asynchronous Module Definition (AMD). Подход AMD декларирует механизм определения и асинхронной загрузки модулей и их зависимостей.

Angular — JavaScript-фреймворк для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки. В Creatio реализована возможность встраивания кастомных Angular-компонентов с использованием единого ядра Angular.

Базовые JS-классы

Неизменяемая часть системы. Эти классы:

- Обеспечивают работу клиентских модулей конфигурации.
- Определяют функциональность основных объектов системы, элементов управления, перечислений и констант.
- Хранятся в виде исполняемых файлов на диске в составе папок приложения.

Механизм сообщений

Sandbox — компонент ядра, который служит диспетчером при взаимодействии модулей системы. Sandbox предоставляет механизм обмена сообщениями между модулями (методы `sandbox.publish()` и `sandbox.subscribe()`) и загрузки модулей по требованию в интерфейс приложения (метод `sandbox.load()`).

Асинхронное определение модулей

Front-end платформы имеет **модульную структуру**.

Модуль — инкапсулированный в обособленный блок набор функциональности, который в свою очередь может использовать другие модули в качестве зависимостей.

Создание модулей в JavaScript декларируется паттерном программирования "Модуль". Классическим приемом реализации этого паттерна является использование анонимных функций, возвращающих определенное значение (объект, функцию и т. д.), которое ассоциируется с модулем. При этом значение модуля экспортируется в глобальный объект.

Для управления большим количеством модулей в Creatio загрузка модулей и их зависимостей выполняется в соответствии с подходом **Asynchronous Module Definition** (AMD).

Подход AMD декларирует механизм определения и асинхронной загрузки модулей и их зависимостей, который в процессе работы с системой позволяет подгружать только те данные, которые необходимы для работы в текущий момент. В Creatio для работы с модулями используется загрузчик **RequireJS**.

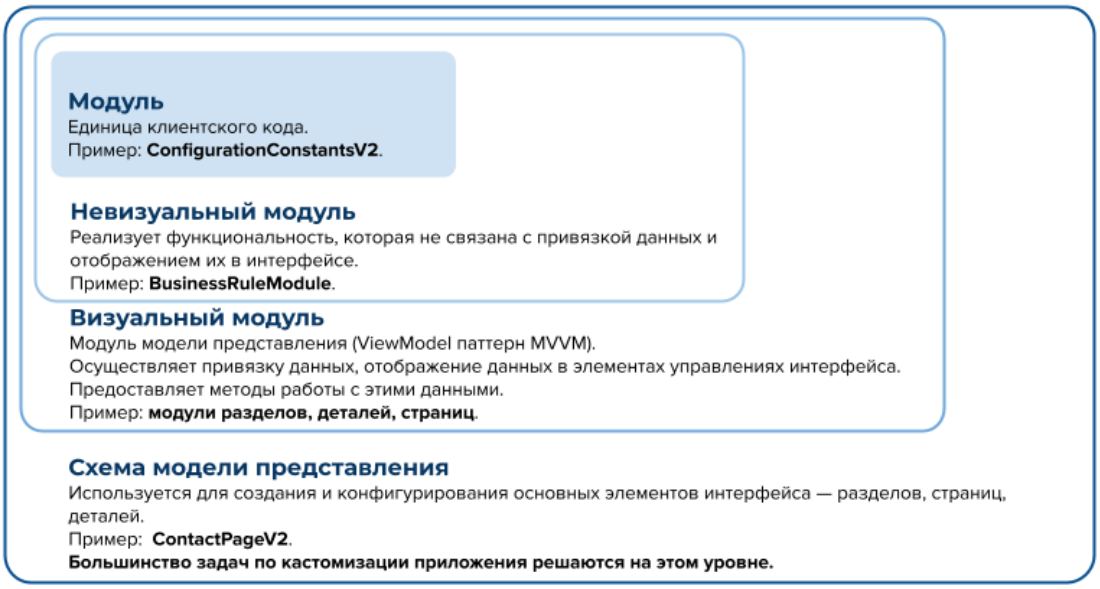
Принципы определения модулей в Creatio:

- Объявление модуля выполняется в специальной функции `define()`, которая регистрирует функцию-фабрику для инстанцирования модуля, но при этом не загружает его немедленно в момент вызова.
- Зависимости модуля передаются как массив строковых значений, а не через свойства глобального объекта.
- Загрузчик выполняет загрузку всех модулей-зависимостей, переданных в качестве аргументов в `define()`. Модули загружаются асинхронно, при этом фактически порядок их загрузки определяется загрузчиком произвольно.
- После того как загрузчиком будут загружены все указанные зависимости модуля, будет вызвана функция-фабрика, которая вернет значение модуля. При этом в функцию-фабрику в качестве аргументов будут переданы загруженные модули-зависимости.

Модульная разработка в Creatio

Реализация всей пользовательской функциональности выполняется в **клиентских модулях**.

Иерархия модулей в Creatio



Несмотря на некоторые функциональные различия, все клиентские модули Creatio имеют одинаковую структуру описания, которая соответствует формату описания модулей AMD.

Общая структура описания клиентского модуля

```
define(
    "ModuleName",
```

```
    "dependencies",
    function(dependencies) {
        // someMethods…
        return { moduleObject };
});
```

- `ModuleName` — имя модуля;
- `dependencies` — модули-зависимости, функциональность которых можно использовать в текущем модуле;
- `moduleObject` — конфигурационный объект созданного модуля.

Виды клиентских модулей в Creatio:

- Невизуальный модуль.
- Визуальный модуль.
- Схема модели представления.

Невизуальный модуль

Содержит реализацию функциональности системы, которая не сопряжена с привязкой данных и отображением их в интерфейсе.

Структура описания невизуального модуля

```
define(
    "ModuleName",
    "dependencies",
    function(dependencies) {
        // Методы, реализующие необходимую бизнес-логику.
        return;
});
```

Примерами невизуальных модулей являются утилитные модули, которые реализуют служебные функции.

Визуальный модуль

Используется для создания пользовательских визуальных элементов системы.

Реализует модель представления (ViewModel) согласно шаблону MVVM. Инкапсулирует в себе данные, которые отображаются в элементах управлениях графического интерфейса, а также методы работы с этими данными.

Визуальный модуль содержит **методы**:

- `init()` — метод инициализации модуля.

Отвечает за инициализацию свойств объекта класса, а также за подписку на сообщения.

- `render(renderTo)` — метод отрисовки представления модуля в DOM.
  Должен возвращать представление.

  Принимает единственный аргумент `renderTo` — элемент, в который будет вставлено представление объекта модуля.
- `destroy()` — метод, отвечающий за удаление представления модуля, удаление модели представления, отписку от ранее подписанных сообщений и уничтожение объекта класса модуля.

Для создания визуального модуля можно использовать базовые классы ядра. Например, создать в модуле класс-наследник `Terrasoft.configuration.BaseModule` или `Terrasoft.configuration.BaseSchemaModule`. Это базовые классы, в которых в необходимой степени уже реализованы необходимые методы визуального модуля — `init()`, `render(renderTo)` и `destroy()`.

Структура описания визуального модуля (наследника базового класса Terrasoft.BaseMo...

```
define("ModuleName", "dependencies", function(dependencies) {
    // Определение класса модуля.
    Ext.define("Terrasoft..configuration.className") {
        alternateClassName: "Terrasoft.className"
        extend: "Terrasoft.BaseModule",
        …
        // Свойства и методы.
        …
    };
    // Создание объекта модуля.
    return Ext.create(Terrasoft.className)
});
```

Примерами визуальных модулей являются модули, реализующие функциональность элементов управления на странице приложения.

Схема модели представления

Наиболее частыми задачами кастомизации приложения являются создание и доработка основных элементов интерфейса — разделов, страниц, деталей.

Модули этих элементов имеют **шаблонизированную структуру** и называются схемами моделей представления.

Схема модели представления является конфигурационным объектом для генерации представления и модели представления генераторами Creatio `ViewGenerator` и `ViewModelGenerator`.

Для большинства задач кастомизации в Creatio используется механизм замещения базовых схем. К таким схемам относятся все схемы из предустановленных пакетов конфигурации.

Основные базовые схемы моделей представления:

- `BasePageV2`

- `BaseSectionV2`
- `BaseDetailV2`

Структура схемы модели представления

```
define("SchemaName", "dependencies",
    function(dependencies) {
        return {
            entitySchemaName: "ExampleEntity",
            mixins: {},
            attributes: {},
            messages: {},
            methods: {},
            rules: {},
            businessRules: {},
            modules: {},
            diff:
        };
});
```

Примерами схем моделей представления являются схемы страниц, разделов и деталей.

Сервис поиска и объединения дублей

Основы

Сервис поиска и объединения дублей (Bulk Duplicate Search) используется для дедупликации (поиска и объединения) дублирующихся записей (дублей), которые могут появиться при добавлении данных в разделах приложения Creatio.

Общие принципы работы

В Creatio реализованы следующие **виды поиска дублей**:

- Массовый поиск дублей — поиск выполняется по всей базе. Запускается вручную либо автоматически.
- Локальный поиск дублей — поиск предусматривает проверку существования дублей для конкретной записи. Запускается при создании новой записи в момент ее сохранения в разделе.

Не запуская поиск, можно в ручном режиме выбрать дублирующие записи и выполнить их слияние. Такая опция доступна для всех разделов приложения.

Возможности сервиса поиска дублей:

- Использование преднастроенных правил поиска дублей в разделах [*Контрагенты*] ([*Accounts*]), [*Контакты*] ([*Contacts*]) и [*Лиды*] ([*Leads*]).
- Настройка пользовательских правил поиска дублей контактов, контрагентов и лидов в

соответствующих разделах.

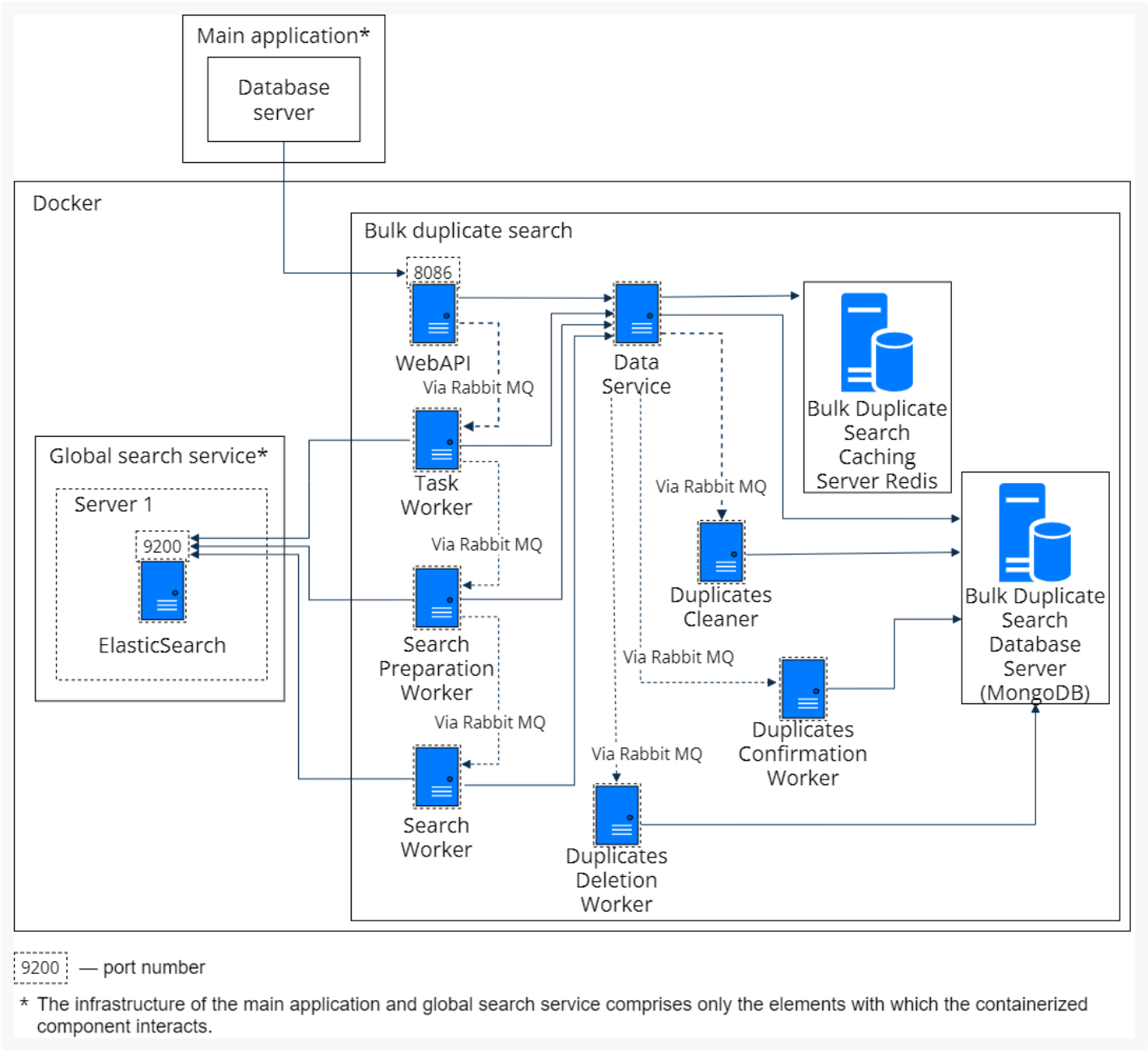
- Создание правил поиска дублей для любого раздела системы, в том числе пользовательского.

Схема работы

Компоненты сервиса поиска и объединения дублей:

- RabbitMQ — брокер сообщений. Компонент сервиса глобального поиска.
- ElasticSearch — поисковая система. Компонент сервиса глобального поиска.
- Redis — хранилище данных, которое используется для кэширования и быстродействия.
- MongoDB — документоориентированная система управления базами данных.
- WebAPI — веб-сервис для общения в приложении Creatio.
- Data Service — внутренний сервис коммуникаций с компонентом MongoDB.
- Duplicates Search Worker — компонент поиска дублей.
- Duplicates Deletion Worker — компонент для точечного удаления дублей.
- Duplicates Confirmation Worker — компонент для группировки и фильтрации найденных дублей с учетом уникальности.
- Duplicates Cleaner — компонент очистки дублей.
- Deduplication Task Worker — компонент для постановки задачи дедупликации.
- Deduplication Preparation Worker — компонент для подготовки процесса дедупликации, который формирует запросы для поиска дублей согласно правилам.

Схема работы сервиса поиска и объединения дублей представлена ниже.



Масштабируемость

Использование кластера баз данных в крупных проектах позволяет выполнить масштабирование сервиса поиска и объединения дублей. Описание кластеризации содержится в документации ElasticSearch.

Совместимость с продуктами Creatio

Сервис поиска и объединения дублей имеет версии 1.0-1.5 и 2.0, которые совместимы со **всеми продуктами Creatio** версий 7.14 и выше.

Варианты развертывания

Сервис поиска и объединения дублей можно развернуть on-site и cloud.

Для пользователей приложения Creatio, которое развернуто cloud, функциональность сервиса поиска и объединения дублей включена по умолчанию.

Для пользователей приложения Creatio, которое развернуто on-site, необходимо выполнить предварительную настройку сервиса глобального поиска в поисковой системе ElasticSearch. Для настройки сервиса поиска и объединения дублей необходим сервер (физическая или виртуальная машина), который должен отвечать техническим требованиям. На сервере должна быть установлена операционная система Linux, которая официально поддерживает программное обеспечение Docker. Перечень операционных систем, которые поддерживает Docker, содержится в документации Docker. Подключение функциональности сервиса поиска и объединения дублей описано в статье Настроить массовый поиск дублей.

Рекомендуем устанавливать актуальную версию сервиса поиска и объединения дублей.

Интеграции

Основы

Интеграция — это обмен данными между системами с возможной последующей обработкой. **Цель** интеграции — автоматический перенос данных, которые внес пользователь, из одной системы в другую.

Платформа Creatio предлагает широкие возможности для интеграции внешних программных продуктов. Открытый API Creatio позволяет создавать интеграционные решения любой сложности.

Creatio поддерживает следующие **способы интеграции**:

- Интеграция внешних приложений с Creatio.
- Интеграция Creatio с внешними приложениями.

Выбор способа интеграции зависит от следующих факторов:

- Потребностей клиента.
- Типа и архитектуры внешних приложений.
- Компетенции разработчика.

Интеграция внешних приложений с Creatio

Задачи интеграции внешних приложений с Creatio:

- Выполнение CRUD-операций с объектами Creatio.
- Запуск бизнес-процессов.
- Реализация пользовательских задач, которые можно решить в рамках открытого API Creatio.

Сервисы работы с данными

Сервисы работы с данными используются для выполнения CRUD-операций с объектами Creatio. В Creatio реализованы следующие сервисы работы с данными:

- Протокол OData.
- Сервис DataService.

Протокол OData

OData (Open Data Protocol) — это утвержденный ISO/IEC стандарт OASIS, который определяет набор лучших практик для построения и использования REST API. Протокол позволяет создавать службы на основе REST, которые с помощью HTTP-запросов предоставляют веб-клиентам возможность публиковать и редактировать ресурсы, идентифицированные с использованием URL и определенные в модели данных.

Приложение Creatio поддерживает протоколы OData 4 и OData 3. OData 4 предоставляет больше возможностей, чем OData 3. Основное **отличие** протоколов — ответ на запрос, возвращаемый сервером, имеет разный формат данных. Различия протоколов OData 3 и OData 4 описаны в официальной документации OData. При планировании интеграции с Creatio по протоколу OData необходимо использовать протокол версии 4.

Детальное описание протокола содержится в документации OData.

Сервис DataService

DataService (разработан Creatio) — сервис, который реализует связь между клиентской и серверной частями платформы. С помощью DataService выполняется передача данных, введенных в пользовательском интерфейсе, в серверную часть приложения для последующей обработки и сохранения в базу данных.

Сервис запуска бизнес-процессов

Системный веб-сервис ProcessEngineService.svc используется для запуска бизнес-процессов из внешнего приложения.

Пользовательский веб-сервис

В конфигурации Creatio существует возможность создавать пользовательские веб-сервисы, которые

используются для реализации специфических интеграционных задач. Конфигурационный веб-сервис представляет собой RESTful-сервис, реализованный на базе технологии WCF.

Интеграция Creatio с внешними приложениями

Используя low-code/no-code инструменты интеграции, можно объединить корпоративные приложения в единую цифровую экосистему. Интеграция Creatio с внешними приложениями предполагает разработку или использование готовых интеграционных решений

Разработка интеграционных решений

Используя no-code инструменты, Creatio позволяет настроить интеграцию с пользовательским RESTful API. После настройки интеграции с веб-сервисом его можно вызвать в бизнес-процессе. Инструменты REST API позволяют взаимодействовать со сторонними веб-сервисами без привлечения разработчиков.

Готовые интеграционные решения

Готовые интеграционные решения, которые предоставляет Creatio, представлены ниже.

Creatio предоставляет **готовые интеграционные решения** со следующими внешними приложениями:

- Порталом OneLogin, который используется в качестве единой точки входа для всех сервисов компании.
- Программным компонентом Active Directory Federation Services (ADFS), который используется для управления возможностью единого входа для всех пользователей системы.
- Функциональностью Just-In-Time User Provisioning (JIT UP), которая избавляет от необходимости создания учетных записей для каждого отдельного сервиса и поддержания актуальности базы пользователей вручную.

- Протоколом прикладного уровня Lightweight Directory Access Protocol (LDAP), который обеспечивает доступ к специализированной базе данных, где обычно хранятся учетные данные пользователей, компьютеров и т. д.
- Почтовым сервисом по протоколу IMAP/SMTP.
- Почтой, календарем и контактами Google.
- Сервисами телефонии Webitel, Oktell, Asterisk, Cisco Finesse, TAPI, CallWay, Infinity, Avaya.
- Сервисами обмена сообщениями и совместной работы MS Exchange и Microsoft 365.

Сервис трекинга событий сайта

Основы

Важно. В следующем релизе Creatio функциональность трекинга событий сайта будет отключена. В качестве альтернативы можно использовать сервис аналитики Matomo. Данные из Matomo импортируются в Creatio при помощи приложения Marketplace: Matomo connector for Creatio.

Сервис трекинга событий сайта предназначен для отслеживания событий, происходящих на сайте клиента. Сервис отправляет данные в облачный сервис трекинга для обработки и отображения в приложении Creatio.

Общие принципы работы

Сервис трекинга событий сайта интегрирован в приложение Creatio и предоставляет возможность построения аналитических отчетов и создания процессов, основанных на активности пользователей на сайте.

Возможности сервиса трекинга событий сайта:

- Сбор данных о просмотрах страниц, количестве посетителей, источниках трафика, свойствах устройств и т. д.
- Сбор данных о действиях посетителей, достижениях целей, конверсиях, маршрутах, кейсах использования сайтов и мобильных приложений.
- Деанонимизация посетителей сайтов.

Виды трекингового кода:

- Базовый — позволяет отслеживать посещение страниц сайта.
- Код отслеживания событий — позволяет отслеживать события сайта с заданными пользователем параметрами. Например, заполнение определенного поля на странице, нажатие кнопки, просмотр видео. Код отслеживания событий используется только вместе с базовым трекинговым кодом.

Для отслеживания на сайте действий клиентов необходимо выполнить ряд предварительных настроек в приложении Creatio, а затем — поместить JavaScript-код, который отправляет события в сервис, в исходный HTML-код каждой страницы веб-сайта. Настройка сервиса трекинга событий сайта описана в статье Настроить трекинг событий сайта. В результате в приложение Creatio будет поступать информация обо всех переходах потенциальных клиентов на веб-сайт и их действиях на сайте.

Трекинговый код запускается при выполнении клиентом тех действий на сайте, для которых предварительно было настроено отслеживание.

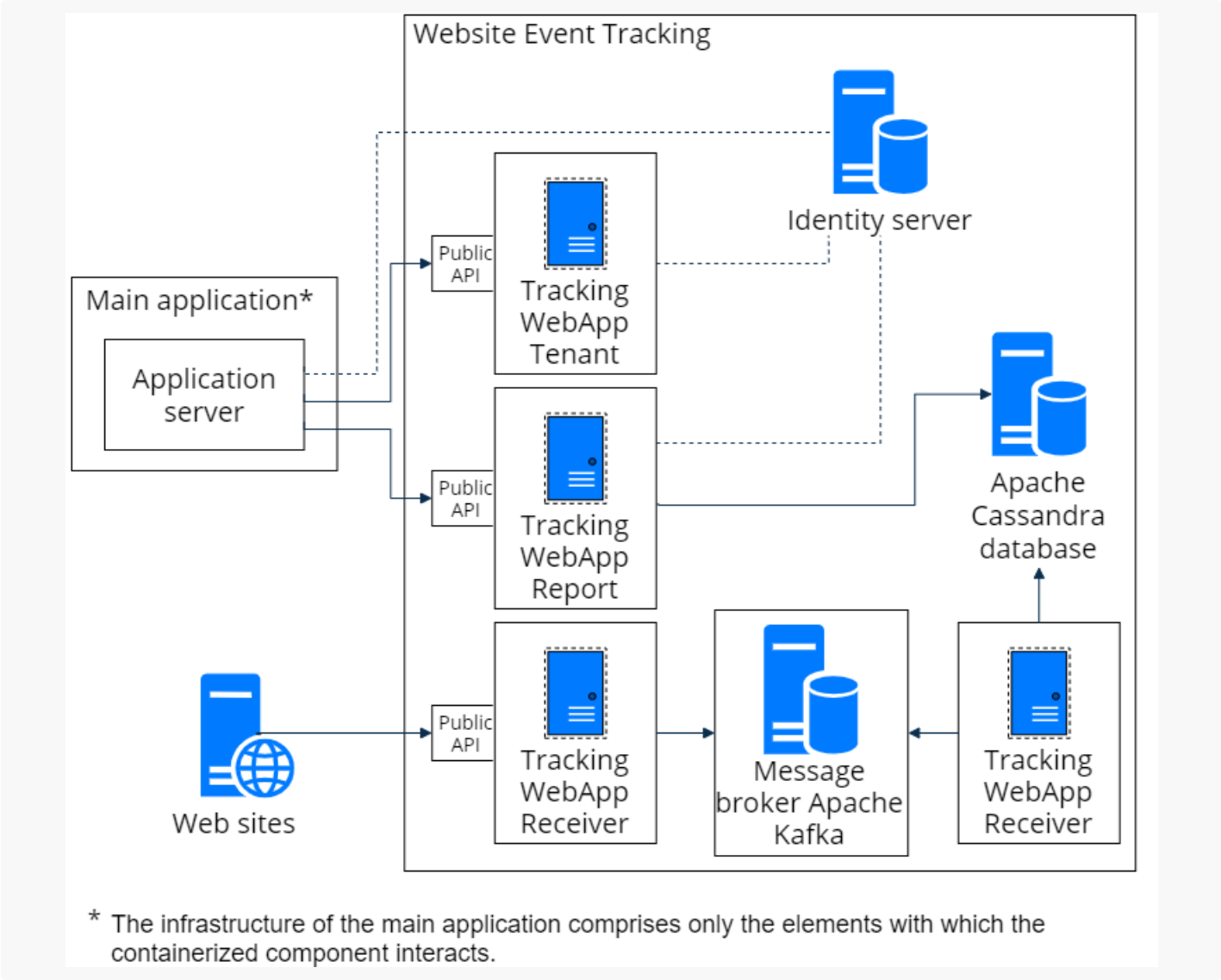
Трекинговый код формирует cookie-файл `BpmTrackingId`, который сохраняет уникальный идентификатор сессии клиента. Благодаря этому все действия, которые клиент совершил на сайте, отобразятся на вкладке [*События сайта*] ([*Website events*]) страницы лида, созданного после регистрации клиента. Приложение Creatio получает информацию обо всех событиях сайта клиента — как тех, которые были выполнены после заполнения формы лендинга и регистрации клиента, так и тех, которые клиент выполнял до регистрации.

Схема работы

Компоненты сервиса трекинга событий сайта:

- Identity server — компонент для идентификации приложения Creatio в сервисе трекинга событий сайта.
- Tracking WebApp Tenant — отвечает за хранение метаданных и настроек подключенного приложения Creatio.
- Tracking WebApp Report — компонент для выборки отчетных данных по запросу.
- Tracking WebApp Receiver — компонент для приема входящих событий от веб-сайтов.
- Message broker Apache Kafka — компонент для обмена данных между другими компонентами.
- Tracking Raw Receiver — компонент для получения необработанных данных по отслеживанию событий.
- Apache Cassandra database — распределенная система управления базами данных, относящаяся к классу NoSQL-систем и рассчитанная на создание высокомасштабируемых и надежных хранилищ больших массивов данных, представленных в виде хэша. Хранит данные отслеживания.

Схема работы сервиса трекинга событий сайта представлена ниже.



Масштабируемость

Благодаря микросервисной архитектуре сервиса трекинга событий сайта масштабируемость достигается стандартными методами, применимыми для каждого компонента, включая кластеризацию, горизонтальное и вертикальное масштабирование. При необходимости хранения больших объемов данных сервис трекинга событий сайта может масштабироваться и создавать новые кластеры (Nodes).

Совместимость с продуктами Creatio

Сервис трекинга событий сайта совместим с **продуктом Marketing Creatio** версий 7.16.4 и выше.

Варианты развертывания

Сервис трекинга событий сайта можно развернуть только cloud.

Сервис обогащения данных

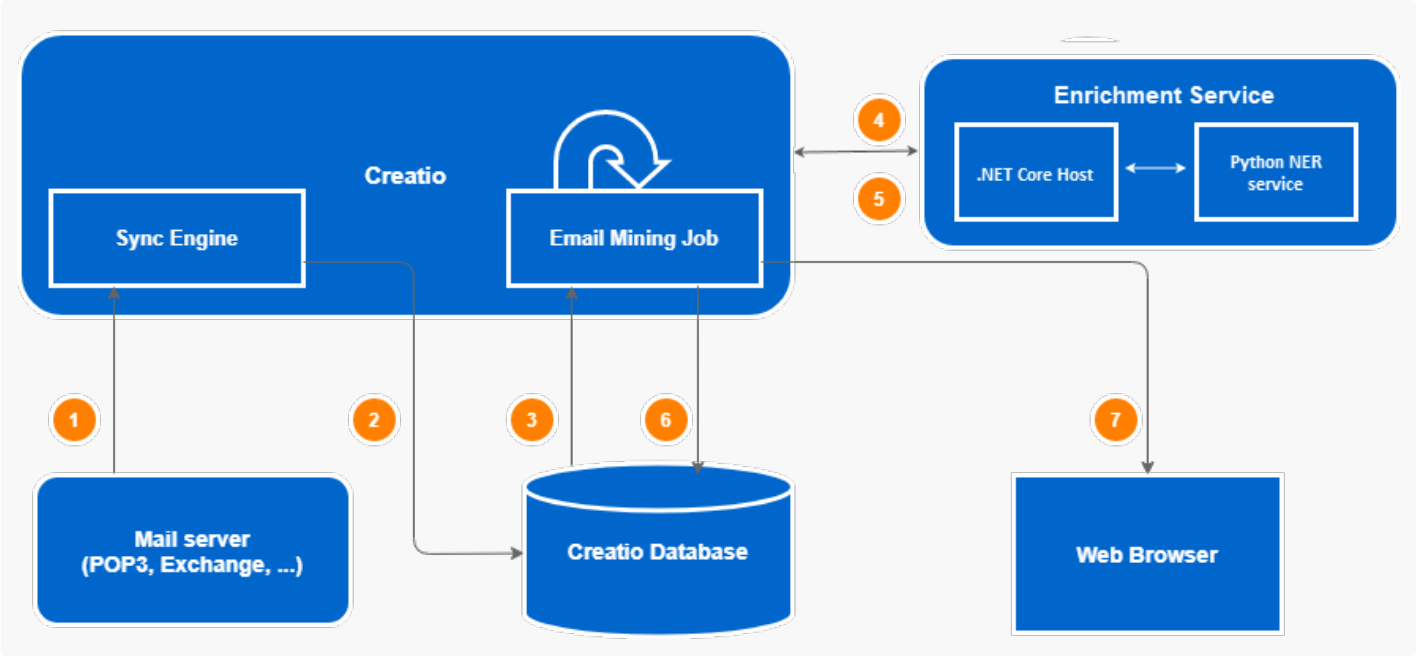
Основы

Сервис обогащения данных предназначен для поиска актуальной информации о компаниях и контактах, а также средствах связи с ними из email-сообщений и открытых источников в Интернете.

Схема работы

Схема работы сервиса обогащения контакта/контрагента информацией из email:

1. Существующий механизм синхронизации Sync Engine выполняет синхронизацию с почтовым сервером. Почтовый сервер передает Sync Engine новые письма (1).
2. **Sync Engine** сохраняет полученные письма в базе данных в виде активностей с **типом "Email"** (2).
3. Планировщик Creatio периодически выполняет задание, которое запускает **процесс "Email Mining Job"** (3). Этот процесс выбирает из базы данных порцию последних (по дате создания) активностей с типом "Email", которые ранее не были им обработаны. Из каждой записи активности выбирается тело письма и его формат (plain-текст или html).
4. Процесс "Email Mining Job" по каждому выбранному письму отправляет http-запрос в облачный сервис обогащения данных Enrichment Service (4).
5. **Enrichment Service** выполняет следующие операции (5):
   - Выделяет из письма цепочку отдельных сообщений (ответов).
   - Выделяет подпись (signature) для каждого сообщения.
   - Из подписи выделяет сущности (entity extraction) — контакт (ФИО), телефоны, email- и web-адреса, социальные сети, другие средства связи, адреса, название организации.
6. Процесс "Email Mining Job" разбирает полученную от сервиса структуру и сохраняет ее в необработанном виде в базе данных Creatio (6).
7. Процесс "Email Mining Job" уведомляет о завершении извлечения информации из письма. Сообщения отправляются по каналам WebSocket пользователям, которые в коммуникационной панели видят обрабатываемые письма (7).



Совместимость с продуктами Creatio

Сервис обогащения данных совместим со всеми продуктами Creatio версий 7.10 и выше.

Варианты установки

Сервис обогащения данных можно использовать для on-site и cloud приложений.

Для использования функциональности обогащения данных в Creatio должен быть указан **персональный ключ** облачных сервисов, а также **URL подключения** к облачным сервисам Creatio.

Для этого используются системные настройки:

- [*Адрес сервиса обогащения контрагентов*] ([*Account enrichment service URL*]) — по умолчанию эта настройка заполнена для всех приложений.
- [*Адрес сервиса семантического анализа текста*] ([*Text parsing cloud service*]) — настройка URL-адреса сервиса обогащения данных контактов.
- [*API-ключ облачных сервисов Creatio*] ([*Creatio cloud services API key*]) — для cloud приложений настройка заполнена по умолчанию. Для настройки сервиса on-site приложений необходимо в службе поддержки запросить персональный ключ и после получения ключа выполнить настройку сервиса. Процесс настройки сервиса для on-site приложений описан в статье Настроить сервис обогащения данных.

Сервис машинного обучения

Основы

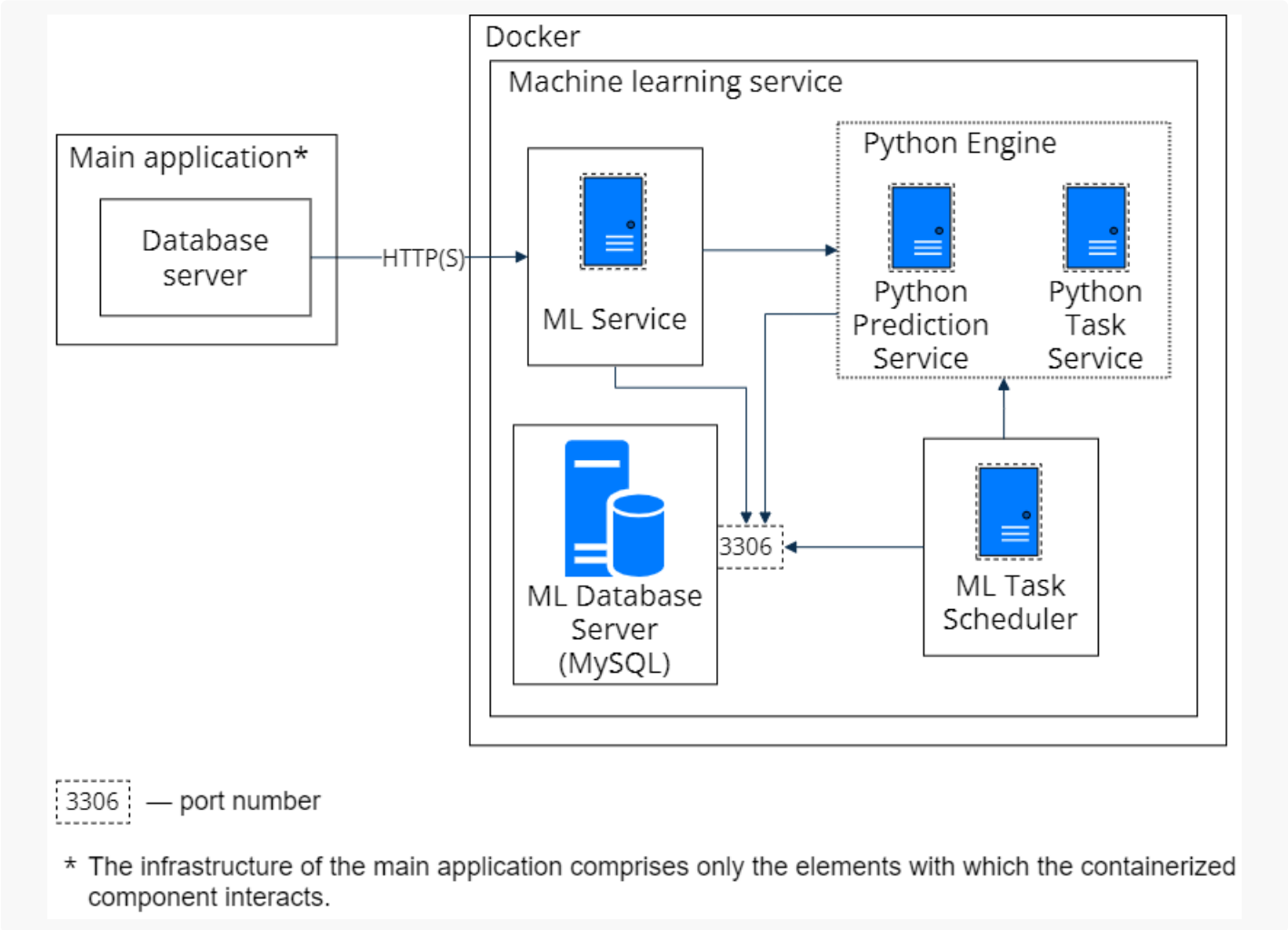
Сервис машинного обучения (сервис прогнозирования значений справочного поля) использует методы статистического анализа для создания модели прогнозирования на основании набора исторических данных.

Схема работы

Сервис машинного обучения состоит из следующих **компонентов**:

- **ML Service** — веб-сервис машинного обучения. Единственный компонент, доступный извне.
- **Python Engine** — движок машинного обучения, представляющий собой сервисную оболочку библиотек машинного обучения с открытым исходным кодом.
- **ML Task Scheduler** — планировщик задач.
- **MySQL** — база данных MySQL.

Схема работы сервиса машинного обучения



В процессе работы сервиса создается **модель прогнозирования** — алгоритм, который строит прогнозы и позволяет автоматически принимать полезное решение на основе исторических данных.

Существует два основных **этапа работы** сервиса для каждой модели:

- обучение;
- прогнозирование.

Обучение

На этапе обучения выполняется "тренировка" ML модели на сервисе.

Основные шаги обучения:

1. Установление сессии передачи данных и обучения.
2. Последовательная выборка порции данных для модели и их загрузка в сервис.
3. Запрос на постановку в очередь для обучения.
4. ML Task Scheduler обрабатывает очередь.
5. Python Engine выполняет обучение модели и запись параметров в БД.
6. Creatio периодически опрашивает сервис для получения статуса модели.
7. Как только для статуса модели установлено значение [*Done*] — модель готова для прогнозирования.

Прогнозирование

Задача прогнозирования выполняется через вызов облачного сервиса с указанием `Id` экземпляра модели и данных для прогноза.

Результат работы сервиса — набор значений с вероятностями, который сохраняется в Creatio в таблице `[MLPrediction]`.

Если существует прогноз в таблице `[MLPrediction]` по определенной записи сущности, то на странице записи автоматически отображаются спрогнозированные значения для поля.

Масштабируемость

Масштабирование сервиса машинного обучения выполняется за счет использования Docker и Kubernetes.

Совместимость с продуктами Creatio

Сервис машинного обучения для **on-site** приложений Creatio совместим со всеми продуктами Creatio версий 7.10 и выше.

Сервис машинного обучения для **cloud** приложений совместим со всеми продуктами Creatio версий 7.13.3 и выше.

Для настройки сервиса в более ранних версиях Creatio необходимо использовать docker-образ соответствующей версии с Docker Hub.

Варианты развертывания

Для использования функциональности предиктивного анализа данных в Creatio **on-site** необходимо выполнить предварительную настройку.

Для настройки сервиса необходим сервер (физический или виртуальный компьютер) с установленной ОС Linux или Windows. Установка компонентов сервиса выполняется с помощью Docker.

Для промышленной среды следует использовать сервер с ОС Linux. Сервер на базе Windows можно использовать только для среды разработки.

Для получения Docker-контейнеров, предназначенных для Windows, обратитесь в службу поддержки.

Установка сервиса машинного обучения описана в статье Сервис машинного обучения.

Сервис синхронизации Exchange Listener

Основы

Назначение сервиса синхронизации Exchange Listener — синхронизация Creatio с почтовыми сервисами MS Exchange и IMAP/SMTP с использованием механизма подписки.

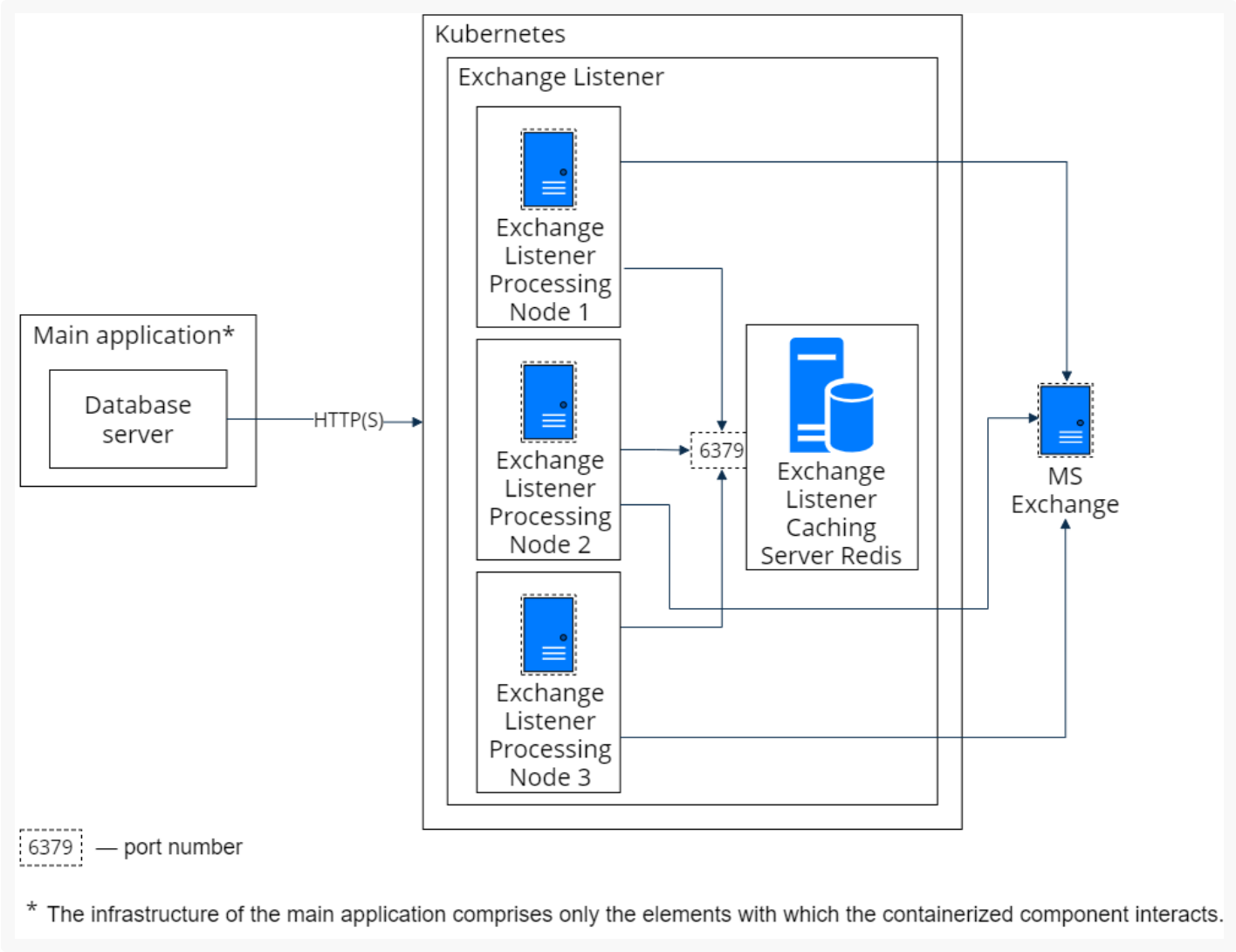
Важно. Начиная с версии 7.17.0, сервис синхронизации Exchange Listener — единственный вариант работы с почтовыми сервисами. Не поддерживается работа с почтовыми сервисами без использования микросервиса.

Схема работы

Обязательные **компоненты** сервиса синхронизации Exchange Listener:

- Основной модуль Exchange Listener.
- NoSQL СУБД Redis.

Схема работы сервиса синхронизации Exchange Listener представлена на рисунке ниже.



Модуль Exchange Listener

Назначение модуля Exchange Listener — используя учетные данные почтового ящика, создает подписку (subscription) для получения событий при поступлении новых писем. Открытая подписка остается в памяти компонента для обеспечения оперативной реакции на получение нового письма. При получении соответствующего события выполняется загрузка экземпляра письма.

NoSQL СУБД Redis

Назначение NoSQL СУБД Redis — создание масштабируемой и отказоустойчивой системы узлов-обработчиков. Хранилище Redis содержит информацию об обслуживаемых почтовых ящиках. Это позволяет любому контейнеру обработать запросы Creatio на создание новой подписки или проверить статус конкретной подписки, независимо от того, на каком узле открыта подписка.

Обязательные **требования** к Redis:

- Для версий приложения 7.18.2 и выше, авторизированный доступ сервиса Exchange Listener к Redis.
- Выделена отдельная база данных для работы сервиса Exchange Listener.

Масштабируемость

Обработка запросов по умолчанию выполняется отдельными узлами типа `StatefulSet` из расчета 1 реплика обработчика на 50 активных почтовых ящиков. За количество реплик отвечает параметр `replicaCount`. При необходимости можно увеличить количество обработчиков, указав необходимое значение при установке. Возможна настройка автоматического масштабирования по количеству активных подписок.

Совместимость с продуктами Creatio

Сервис синхронизации Exchange Listener версии 1.0 (поддержка MS Exchange) совместим со всеми продуктами Creatio версий 7.15.2 и выше.

Сервис синхронизации Exchange Listener версии 2.0 (поддержка IMAP/SMTP) совместим со всеми продуктами Creatio версий 7.16 и выше.

Варианты установки

Для развертывания сервиса и обеспечения работы приложения на **продуктовой среде** предпочтительным способом является использование оркестратора Kubernetes и пакетного менеджера Helm. Инструкция по развертыванию сервиса с использованием Kubernetes содержится в статье Развернуть сервис синхронизации с использованием Kubernetes.

Для быстрого развертывания в среде разработки можно использовать Docker. Инструкция по развертыванию сервиса с использованием Docker содержится в статье Развернуть сервис синхронизации в Docker.

Для развертывания сервиса достаточным условием является использование in-memory хранилища.

Сервис бандлирования статического контента

Основы

Для повышения производительности весь клиентский контент (исходный код клиентских схем, css-стили) генерируется в специальном каталоге приложения Creatio. При большом количестве файлов браузеру необходимо выполнять большое количество запросов к приложению при его первоначальной загрузке. Чтобы избежать этого, выполняется объединение всех однотипных файлов в один bundle-файл — **бандлирование**. Для объединения однотипных файлов статического контента Creatio разработан сервис бандлирования статического контента.

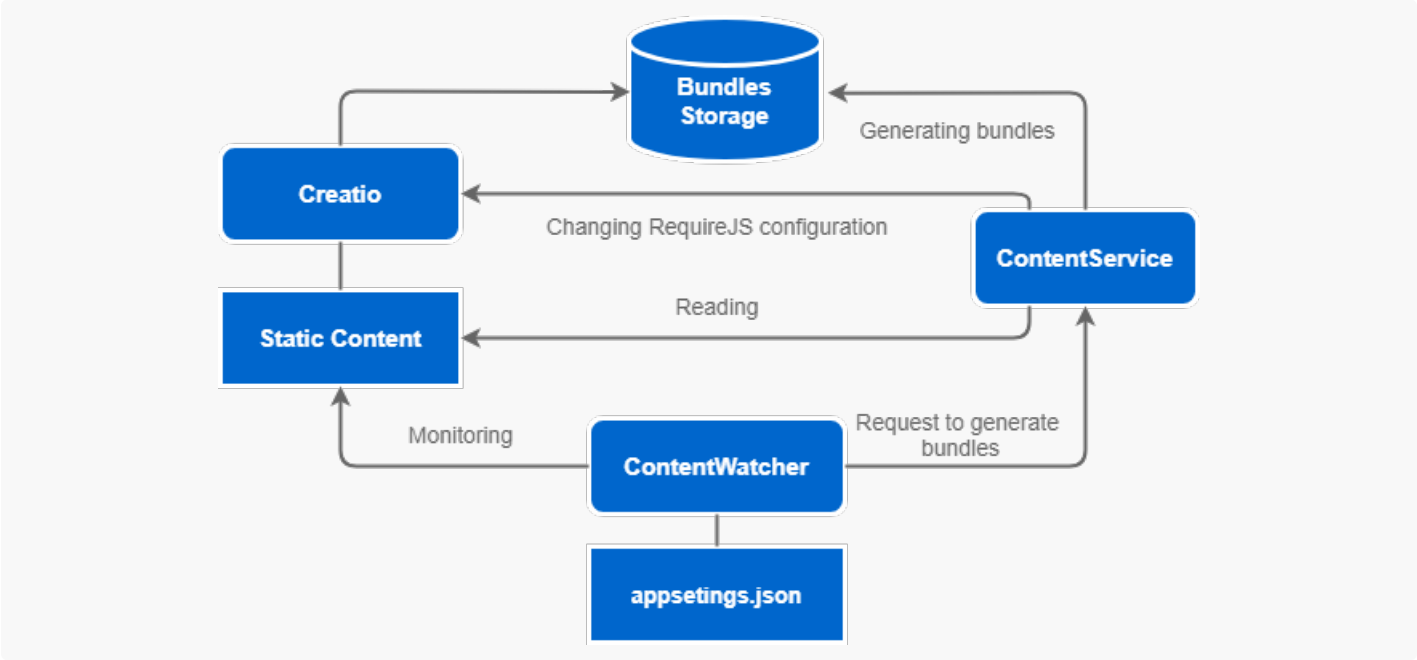
Схема работы

Приложение-наблюдатель (`ContentWatcher`) мониторит файлы в каталоге со статическим контентом и сообщает об изменениях в них веб-сервису.

Веб-сервис (`ContentService`) выполняет перегенерацию bundle-файлов приложения при поступлении определенных запросов (например, от `ContentWatcher` или вручную). После бандлирования веб-сервис

изменяет определенный конфигурационный файл Creatio таким образом, чтобы он содержал информацию о необходимости использовать bundle-файлы вместо оригинального статического контента.

Принципиальная схема сервиса бандлирования статического контента



ContentService

Является .NET Core 2.1 веб-сервисом и имеет следующие операции (точки доступа):

- `/` — проверка работоспособности сервиса (метод GET).
- `/process-content` — генерирует минифицированные bundle-файлы (метод POST).
- `/clear-bundles` — очищает bundle-файлы (метод POST).
- `/minify-content` — минифицирует контент (метод POST).

ContentWatcher

Является .NET Core 2.1 приложением. Запускается как служба (также может запускаться через .NET Core CLI Tools). Основная задача — наблюдать за изменением указанного в параметре `fileFilter` файла, который находится по пути, указанном в параметре `directory`. По умолчанию это файл `_MetaInfo.json`. Изменение файла сообщает об обновлении статического контента. При обнаружении изменений `ContentWatcher` оповещает `ContentService` о необходимости перегенерировать bundle-файлы.

Совместимость с продуктами Creatio

Сервис бандлирования статического контента совместим со всеми продуктами Creatio версий 7.11 и выше.

Варианты установки

Сервис бандлирования статического контента можно использовать для on-site и cloud приложений.

Для развертывания сервиса **on-site** используется Docker-контейнер.

Веб-сервис может быть установлен без приложения-наблюдателя (`ContentWatcher`). В таком случае запросы к `ContentService` на бандлирование или минификацию необходимо выполнять вручную.

Компоненты сервиса могут быть установлены на том же компьютере, что и Creatio, или на отдельном компьютере. Если они установлены отдельно, то у них должен быть сетевой доступ к файлам статического контента Creatio.

Чтобы развернуть сервис для **cloud** приложения обратитесь в службу поддержки.

Системные требования:

- Сервер под управлением Linux OS (рекомендуются стабильные версии Ubuntu или Debian), с установленной и настроенной стабильной версией docker. С этого сервера должны быть разрешены запросы к хранилищу образов Docker Hub.
- На сервере должны быть установлены Docker и Docker Compose.