

Описание функциональных характеристик программного обеспечения

Data Quality Framework

Data Quality Framework – Описание функциональных характеристик программного обеспечения

Аннотация

Настоящий документ описывает назначение и функциональные характеристики программного обеспечения (далее – ПО) Data Quality Framework.

Исключительные права на ПО Data Quality Framework (далее – ПО DQF) принадлежат ООО «Клин Дейта».

Data Quality Framework – Описание функциональных характеристик программного обеспечения

Оглавление

1.	Описание функциональных характеристик	5
1.1.	Компонент контроля качества (ПО DQF: Engine).....	5
1.1.1.	Алгоритмы проверок.....	6
1.1.1.1.	Встроенные алгоритмы проверок	7
1.1.1.2.	Базовые алгоритмы проверок	7
1.1.1.3.	Комплексные алгоритмы проверок	9
1.1.1.4.	Конфигурации алгоритмов проверок	9
1.1.2.	Взаимодействие с брокером сообщений (API).....	9
1.1.3.	Мониторинг и контроль работоспособности.....	12
1.2.	Компонент сопоставления данных (ПО DQF: Federation).....	14
1.2.1.	Алгоритм сопоставления	14
1.2.1.1.	Данные для сопоставления	15
1.2.1.2.	Правила сопоставления.....	15
1.2.2.	Взаимодействие с брокером сообщений	16
1.2.3.	Мониторинг и контроль работоспособности.....	17
2.	Требования к программно-аппаратной платформе	18
2.1.	Требования к программно-аппаратной платформе компонента контроля качества (ПО DQF: Engine).....	18
2.1.1.	Требования к аппаратной инфраструктуре.....	18
2.1.2.	Требования к программному обеспечению	18
2.1.3.	Требования к сетевой инфраструктуре	19
2.2.	Требования к программно-аппаратной платформе компонента сопоставления данных (ПО DQF: Federation).....	19
2.2.1.	Требования к аппаратной инфраструктуре.....	19
2.2.1.1.	Конфигурация сервиса Узла.....	19
2.2.1.2.	Конфигурация БД Узла.....	20
2.2.1.3.	Конфигурация сервиса Центра.....	20
2.2.1.4.	Конфигурация БД Центра.....	20
2.2.1.5.	Конфигурация БД кэша Центра	20
2.2.2.	Требования к программному обеспечению	21
2.2.3.	Требования к сетевой инфраструктуре	21

Введение

Программное обеспечение Data Quality Framework (далее – ПО DQF) реализует функции контроля качества и сопоставления данных.

Компонент контроля качества (далее – ПО DQF: Engine) обеспечивает контроль данных, которые рассматриваются как целостное образование, состоящее из взаимосвязанных частей. ПО DQF: Engine нацелено на мониторинг качества данных на различных уровнях и обнаружение ошибок по сформированным правилам и критериям проверок.

Функциональные возможности ПО DQF: Engine позволяют с помощью специальных правил и алгоритмов производить верификацию проверяемого объекта и его атрибутов и выявлять ошибки, а также настраивать проверки данных в соответствии с предметной областью и заданной конфигурацией (бизнес-логики) исполняемой проверки без доработки программного обеспечения.

Компонент сопоставления данных (далее – ПО DQF: Federation) обеспечивает сопоставление данных объектов (физических, юридических и иных объектов) из разных источников.

Компоненты ПО DQF могут работать как комплексно, так и независимо друг от друга.

1. Описание функциональных характеристик

1.1. Компонент контроля качества (ПО DQF: Engine)

ПО DQF: Engine обеспечивает:

- выполнение проверок данных по заданным алгоритмам и настраиваемым параметрам;
- выполнение проверок по массивам данных и единичных записей;
- выявление ошибок (противоречий) в имеющихся и вносимых данных;
- верификацию невалидных данных с возможностью их замены, в том числе в автоматизированном режиме;
- мониторинг и оценку качества данных на предмет полноты, достоверности и непротиворечивости.

Особенностью, реализованной ПО DQF: Engine, является оперирование высокоабстрагированными объектами, с определенным уровнем обобщения, при наличии метамодели данных. Такая отвязка от предметной области позволяет применять ПО DQF: Engine к различным областям и моделям описания данных. При этом ПО DQF: Engine обладает определенной автономностью в аспекте настройки любых логических проверок без привлечения разработчиков за счет возможности настройки проверок данных в соответствии с предметной областью с использованием конфигураций алгоритмов проверок.

ПО DQF: Engine может быть интегрировано в любую среду оркестрации, обеспечивает проверку объектов в памяти, без сохранения на диск.

При выполнении проверок ПО DQF: Engine осуществляет контроль за выполнением алгоритмов проверок, после чего предоставляет результаты проверок. При необходимости ПО DQF: Engine может осуществлять запросы дополнительных данных, необходимых для проведения проверок, в том числе из внешних информационных систем.

При росте нагрузки и количества запросов ПО DQF: Engine обладает качеством гибкого масштабирования. Также предусмотрена схема развертывания с горячим резервом с двумя активными экземплярами ПО и балансировщиком для обеспечения отказоустойчивости решения.

ПО DQF: Engine осуществляет автоматизированные проверки по трем направлениям:

- форматно-логический контроль:
 - наличие обязательных атрибутов;
 - соответствие атрибутов заданной длине и маске;
 - соответствие атрибутов требуемым форматам – буквенно-числовые последовательности, непечатаемые символы, специальные символы;
- проверки внутри модели данных:
 - проверки по внутренним справочникам;
 - проверки на соответствие нормативно-правовым актам;
 - проверки на связность и непротиворечивость объектов.
- интеграционные проверки с использованием внешних источников:
 - проверки по внешним справочникам;
 - проверки по данным из Единой системы межведомственного электронного взаимодействия;
 - проверки по данным из Национальной системы управления данными;
 - проверки по данным из внешних информационных систем.

1.1.1. Алгоритмы проверок

Алгоритм проверки – это конечный набор шагов для обеспечения проверки данных посредством конечного количества операций.

В ПО DQF: Engine реализованы следующие типы алгоритмов:

- встроенные;
- базовые;
- комплексные.

Любой алгоритм имеет основную ветвь исполнения и может иметь несколько альтернативных ветвей исполнения (отмены, боковые ответвления и др.). Чтобы обеспечить возможность организации циклов и альтернативных веток исполнения, каждый алгоритм возвращает статус, логика определения которого задается отдельно для каждого алгоритма. В случае необходимости передачи данных между алгоритмами в процессе выполнения проверки, помимо статуса алгоритм может передать дополнительные данные через контекст исполнения.

Контекст исполнения состоит из следующих данных:

- входящий запрос:

- исходный проверяемый объект;
- переменные исполнения, заполненные на предыдущих шагах комплексного алгоритма.

- исходящий результат:
 - статус выполнения базового алгоритма;
 - переменные исполнения, при необходимости дополненные результирующими данными.

1.1.1.1. Встроенные алгоритмы проверок

Встроенные алгоритмы – это связующее звено, объединяющее базовые алгоритмы в единую последовательность исполнения – комплексный алгоритм.

ПО DQF: Engine содержит следующие встроенные алгоритмы:

- последовательное исполнение шагов – all-of (последовательности действий);
- альтернативная ветвь исполнения- otherwise (ветвления);
- итерирование по коллекции – for-each (циклы);
- цикл, пока выполняется условие – while (циклы с условием).

1.1.1.2. Базовые алгоритмы проверок

Базовый алгоритм проверки – это атомарная операция, производимая в определенном контексте над объектом/атрибутом объекта определенного типа. Из базовых алгоритмов составляются логически связанные цепочки, которые формируют сущность проверки. Базовые алгоритмы проверок реализованы в java-коде ПО DQF: Engine.

Перечень базовых алгоритмов проверок, реализованных в ПО DQF: Engine, представлен в таблице (Таблица 1).

Таблица 1 - Перечень базовых алгоритмов проверок, реализованных в ПО DQF

№	Наименование (идентификатор в ПО DQF)	Содержание базового алгоритма проверки
1	GraphQLQuery	Запрос объектов из GraphQL-сервиса
2	CheckValueFromDictionary	Проверка значения на вхождение в справочник
3	CheckValue	Проверка и сравнение значений атрибутов: <ul style="list-style-type: none">– сравнение атрибута с константой;– проверка атрибута на нулевое значение;– проверка заполненности атрибута;

№	Наименование (идентификатор в ПО DQF)	Содержание базового алгоритма проверки
		<ul style="list-style-type: none"> – проверка атрибута на соответствие определенному типу; – проверка атрибута на соответствие регулярному выражению; – проверка атрибута путем математического преобразования по формуле; – проверка атрибута типа «дата» на соответствие временному интервалу (периоду); – проверка атрибута на уникальность значения в пределах проверяемого объекта; – проверка атрибута на соответствие справочному значению внутренних справочников; – проверка связности данных в системе Заказчика (проверка ссылочной целостности); – сравнение атрибутов связанных объектов; – Проверка ФИО на наличие опечатки в написании; – проверка количества проверяемых объектов.
4	SetValue	Установка значения атрибута, в том числе с использованием математических операций.
5	Collect	Сбор и фильтрация данных
6	AppendValue	Добавление атрибута в список
7	CheckSnils	Проверка СНИЛС подсчетом контрольного числа в соответствии с алгоритмом, присвоения кода качества
8	CheckPersonalInn	Проверка идентификационного номера налогоплательщика (далее – ИНН) подсчетом контрольного числа в соответствии с алгоритмом для физических лиц, присвоения кода качества
9	CheckLegalEntityInn	Проверка ИНН подсчетом контрольного числа в соответствии с алгоритмом для юридических лиц, присвоения кода качества
10	CheckOgrn	Проверка ОГРН подсчетом контрольного числа в соответствии с алгоритмом, присвоения кода качества

№	Наименование (идентификатор в ПО DQF)	Содержание базового алгоритма проверки
11	CheckAddress	Проверка адреса на вхождение в Государственный адресный реестр (далее – ГАР)
12	Continue	Продолжить исполнение

1.1.1.3. Комплексные алгоритмы проверок

Комплексные алгоритмы проверок определяются в виде последовательностей шагов – вызовов базовых и встроенных алгоритмов.

1.1.1.4. Конфигурации алгоритмов проверок

Конфигурация алгоритма проверки содержит машиночитаемое описание последовательности применения параметризованных базовых алгоритмов с использованием встроенных алгоритмов. Конфигурация алгоритма проверки содержит следующие реквизиты:

- код порождаемой ошибки;
- список шагов алгоритма проверки, содержащих параметризованные конкретными значениями вызовы базовых алгоритмов, а также дополнительные параметры для формирования результата выполнения проверки.

Синтаксис конфигурации построен на основе языка YAML 1.2.2.

ПО DQF: Engine не осуществляет контроль версий конфигураций алгоритмов проверок.

1.1.2. Взаимодействие с брокером сообщений (API)

Программный интерфейс ПО DQF: Engine обеспечивает прием атомарных заданий на проверку и передачу результатов их выполнения в асинхронном режиме с использованием брокера сообщений. Атомарное задание на проверку – запрос на одно выполнение проверки данных по единичному алгоритму проверки.

В ПО DQF: Engine реализуется контракт взаимодействия с помощью сообщений в соответствии с AMQP (Advanced Message Queuing Protocol). Получение атомарных заданий на проверку реализуется из входящей очереди брокера сообщений. Передача результатов выполнения атомарных заданий на проверку реализуется в исходящую очередь брокера сообщений. Все сообщения имеют определенную структуру как способ запроса в брокер сообщений, состоящую из заголовочной части и тела сообщения.

Data Quality Framework – Описание функциональных характеристик программного обеспечения

Заголовочная часть сообщения-запроса и сообщения-ответа общая и содержит атрибуты, приведенные в таблице (Таблица 2).

Таблица 2 - Заголовочная часть сообщения-запроса и сообщения-ответа

№	Состав заголовка
1	Уникальный корреляционный идентификатор задания на проверку
2	Дата и время отправки сообщения

Тело сообщения-запроса на выполнение проверки представляет собой JSON-объект, который содержит атрибуты, приведенные в таблице (Таблица 3).

Таблица 3 - Тело сообщения-запроса на выполнение проверки

№	Атрибуты JSON-объекта	Примечание
1	Идентификатор настройки алгоритма проверки	
2	JSON-объект с данными проверяемого объекта	При передаче задания на выявление ошибки в рамках проверки входящих данных проверяемого объекта
3	Список JSON-объектов, содержащих данные проверяемых объектов одного типа	При передаче задания на выявление ошибки в рамках проверки входящих данных массива объектов
4	Один идентификатор объекта внешней системы	При передаче задания на выявление ошибки в рамках проверки объекта, данные по которому необходимо запросить DQF: Engine
5	Список идентификаторов объектов одного типа	При передаче задания на выявление ошибки в рамках проверки массива объектов, данные по которым необходимо запросить DQF: Engine

Тело сообщения-ответа с результатами обработки сообщения-запроса на выполнение атомарной проверки по факту завершения выполнения алгоритма проверки представляет собой JSON-объект, который содержит атрибуты, приведенные в таблице (Таблица 4).

Таблица 4 - Тело сообщения-ответа с результатами обработки сообщения-запроса

№	Атрибуты JSON-объекта	Примечание
1	Код результата обработки атомарного задания на проверку	
2	Опциональный список, содержащий пути в формате JSONPath к реквизитам объекта, в которых были обнаружены ошибки	В случае обнаружения ошибок

№	Атрибуты JSON-объекта	Примечание
3	Опциональный список, содержащий значения дополнительных атрибутов проверяемого объекта	В случае обнаружения ошибок и необходимости формирования сообщений пользователям систем Заказчика, взаимодействующих с ПО DQF
4	Содержание системной ошибки	В случае возникновения системной ошибки

ПО DQF: Engine не обеспечивает аутентификацию и авторизацию атомарных заданий на проверку из брокера сообщений.

С целью временного хранения полученных декларативных описаний (конфигураций) алгоритмов проверок применяется механизм кэширования.

1.1.3. Мониторинг и контроль работоспособности

ПО DQF: Engine осуществляет ведение журналов диагностических событий, в которых автоматически фиксируются возникающие нештатные ситуации и ошибки, и сохранение в этих журналах информации, необходимой для идентификации проблемы при возникновении аварийных ситуаций, либо ошибок в программном обеспечении. В таблице (Таблица 5) приведены примеры диагностических событий и характеристик.

Таблица 5 - Примеры диагностических событий и характеристик ПО DQF: Engine

№	Диагностическое событие/характеристика
1	Получение сообщения-запроса на атомарную проверку
2	Возврат сообщения-ответа, содержащего результат выполнения атомарной проверки
3	Запрос данных из внешней систем-источников
4	Получение данных из внешних систем-источников
5	Выполнение шагов алгоритма проверки
6	Время выполнения алгоритма проверки
7	Системные ошибки и предупреждения

Для контроля показателей работоспособности/сбоев ПО DQF: Engine при его функционировании с применением средств мониторинга программных компонентов, а также контроля корректности функционирования ПО DQF: Engine с применением сигнализаторов (извещателей), предусмотрены метрики, приведенные в таблице (Таблица 6).

Таблица 6 - Метрики работоспособности ПО DQF: Engine

№	Метрика работоспособности
1	Текущее состояние работоспособности сервиса
2	Количество одновременно исполняемых проверок
3	Количество заэкшированных в данный момент конфигураций алгоритмов проверок
4	Количество заэкшированных в данный момент объектов внешних систем Заказчика

1.2. Компонент сопоставления данных (ПО DQF: Federation)

ПО DQF: Federation обеспечивает:

- сопоставление данных из разных источников по настраиваемым правилам;
- формирование инцидентов качества данных при выявлении конфликтов сопоставлений.

Особенностью, реализованной ПО DQF: Federation, является оперирование только данными, необходимыми для сопоставления, при этом для повышения уровня практической безопасности данные могут быть изменены функцией необратимого преобразования, в результате которой невозможно восстановить первоначальные персональные или иные бизнес-значимые данные. ПО DQF: Federation не хранит персональные или иные бизнес-значимые данные в открытом виде, исходные данные остаются строго внутри своих систем-источников.

ПО DQF: Federation реализует сопоставление измененных данных в потоковом режиме, предоставляя результат сопоставлений с минимальными возможными задержками.

Для повышения сопоставимости данных они должны предварительно пройти стандартизацию по единым для всех источников правилам, а также проверку качества.

Для обеспечения сопоставления задействованы следующие сервисы:

- узлы, обеспечивающие хранение свойств объекта, необходимых для сопоставления;
- центр, обеспечивающий управление поиском сопоставлений, формирование инцидентов качества данных, хранение результатов сопоставления.

При росте нагрузки и количества запросов ПО DQF: Federation обладает качеством гибкого масштабирования. При росте нагрузки количество экземпляров сервисов центра и узлов может быть увеличено, система автоматически сбалансирована.

С целью временного хранения данных и блокировок обрабатываемых объектов применяется механизм распределенного кэширования.

1.2.1. Алгоритм сопоставления

ПО DQF: Federation находит сопоставления объектов по заранее настроенным правилам сопоставления, описанным в п. 1.2.1.2. Результат сопоставления ПО DQF: Federation сохраняет в БД Центра.

Сопоставление данных объектов обладает следующими особенностями:

- отсутствие сбора всех данных объекта в одном месте, отсутствие информации о сопоставлении объектов в сервисах хранения их свойств (Узлах);
- рекурсивное продолжение поиска сопоставлений, процесс сопоставления продолжается для результирующего объекта, содержащего расширенный набор значений свойств объектов, сопоставленных ранее (транзитивное сопоставление);
- неизменность результата сопоставлений от порядка поступления обновлений, в том числе невозможность обновления старыми данными более актуальных.

1.2.1.1. Данные для сопоставления

ПО DQF: Federation при сопоставлении объектов оперирует следующими данными:

- карточкой с записью о сопоставляемом объекте;
- свойствами карточки, которые могут быть двух типов:
 - реквизит – свойство, которое содержит не более одного значения;
 - атрибут – свойство, которое может содержать несколько значений;
- компонентами свойств карточки, которые представляют собой части свойств, которые выделяются в случае, если для компонентов свойств предусмотрена отдельная логика определения совпадений;
- фрагментами компонентов свойств карточки или свойств карточки, технически разделенных на части и преобразованных в целях повышения практической безопасности.

В целях повышения уровня практической безопасности данные по фрагментам должны храниться изолированно друг от друга. Администрирование изолированных узлов, обеспечивающих хранение свойств объекта, необходимых для сопоставления, должно осуществляться независимыми специалистами.

Перечень свойств и компонент свойств формируется в соответствии с заданными правилами, определяемыми индивидуально на основе бизнес-требований.

Количество фрагментов и алгоритм необратимого преобразования данных подбирается индивидуально в зависимости от правил сопоставления, параметров нагрузки и требований информационной безопасности.

1.2.1.2. Правила сопоставления

Правила сопоставления – это сценарии, описывающие при каких условиях две карточки считаются сопоставленными. Правила задаются в разрезе компонентов свойств

карточки и методов сравнения этих компонентов. Перечень правил сопоставления формируется в соответствии с заданными правилами, определяемыми индивидуально на основе бизнес-требований.

Возможные методы сравнения представлены в таблице (Таблица 7).

Таблица 7 - Методы сравнения компонентов

№	Имя метода сравнения	Описание
1	Равенство обязательных компонентов свойств	Значения заданы в обеих карточках и есть пара значений из первой и второй карточки, совпадающих между собой.
2	Равенство необязательных компонентов свойств	Одно из условий должно быть выполнено: – значения заданы в обеих карточках и есть пара значений из первой и второй карточки, совпадающих между собой; – значения не заданы в обеих карточках одновременно.
3	Не противоречие	Для свойства типа реквизит одно из условий должно быть выполнено: – значения заданы в обеих карточках и есть пара значений из первой и второй карточки, совпадающих между собой. – значение не задано хотя бы в одной из карточек. Свойства типа атрибут всегда не противоречат друг другу.

Две карточки считаются сопоставленными по данному правилу сопоставления, если сопоставились все компоненты реквизитов и атрибутов из данного правила по заданному в правиле методу сопоставления.

1.2.2. Взаимодействие с брокером сообщений

ПО DQF: Federation обеспечивает прием данных на сопоставление по конкретной карточке. Получение карточек на сопоставление реализуется из входящей очереди брокера сообщений.

ПО DQF: Federation не обеспечивает аутентификацию и авторизацию сообщения на сопоставление из брокера сообщений. Все сообщения имеют определенную структуру как способ запроса в брокер сообщений, состоящую из заголовочной части и тела сообщения.

Заголовочная часть сообщения на сопоставление соответствует спецификации <https://www.w3.org/TR/trace-context-1/>

Тело сообщения на сопоставление представляет собой JSON-объект, который содержит атрибуты, приведенные в таблице (Таблица 8).

Таблица 8 - Тело сообщения на сопоставление

№	Имя атрибута JSON-объекта	Тип атрибута	Описание
1	sourceId	String	Идентификатор источника.
2	cardId	String	Идентификатор карточки в источнике.
3	version	Number	Версия обновления – монотонно возрастающая последовательность для истории изменений одной и той же карточки.
4	fragments	Object	Ассоциативный массив: – имя фрагмента; – список его значений, упорядоченных в соответствии с индексами значений. Должен быть пуст, если карточка удалена.
5	deleted	Boolean	Флаг, показывающий, удалена ли карточка.

1.2.3. Мониторинг и контроль работоспособности

ПО DQF: Federation осуществляет ведение журналов диагностических событий, в которых автоматически фиксируются возникающие нештатные ситуации и ошибки, и сохранение в этих журналах информации, необходимой для идентификации проблемы при возникновении аварийных ситуаций, либо ошибок в программном обеспечении. В таблице (Таблица 9) приведены примеры диагностических событий и характеристик.

Таблица 9 - Примеры диагностических событий и характеристик ПО DQF: Federation

№	Диагностическое событие/характеристика
1	Получение сообщения-запроса на проведение сопоставления
2	По карточке произошло сопоставление
3	По карточке завершено сопоставление: (один из двух вариантов): – по карточке было отменено сопоставление; – по карточке завершено сопоставление.
4	Системные ошибки и предупреждения

Для контроля показателей работоспособности/сбоев ПО DQF: Federation при его функционировании с применением средств мониторинга программных компонентов, а также контроля корректности функционирования ПО DQF: Federation с применением сигнализаторов (извещателей), предусмотрены метрики, приведенные в таблице (Таблица 10).

Таблица 10 - Метрики работоспособности ПО DQF: Federation

№	Метрика работоспособности
1	Текущее состояние работоспособности сервиса
2	Количество обновленных карточек, ожидающих сопоставление
3	Количество обработанных исходных карточек

2. Требования к программно-аппаратной платформе

2.1. Требования к программно-аппаратной платформе компонента контроля качества (ПО DQF: Engine)

2.1.1. Требования к аппаратной инфраструктуре

Процессор:

- Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);
- от 6 ядер.

ОЗУ:

- от 4 Гб.

Дисковое пространство:

- от 50 Гб.

2.1.2. Требования к программному обеспечению

Управление очередью сообщений:

- брокер сообщений (RabbitMQ).

Сервер приложений:

- ПО Docker 20.10+;
- Операционная система с возможностью установки ПО Docker 20.10+.

Веб сервер/балансировщик нагрузки:

- nginx версии 1.15.

ПО для организации отказоустойчивости сетевых сервисов и балансировки нагрузки:

- HAProxy версии 2.0.

Мониторинг:

- Prometheus версии 2.27.1;
- Grafana версии 7.5.7.

Логирование (ELK):

- Elasticsearch версии 7.11.2;
- Kibana версии 7.11;
- Fluent Bit версии 1.7.9.

2.1.3. Требования к сетевой инфраструктуре

Предъявляются следующие требования к сетевой инфраструктуре:

- пропускная способность канала между сервером приложений и внешними информационными системами, направляющими запросы в ПО DQF, составляет не менее 1 Гбит/с;
- пропускная способность каналов сервером приложений и брокером сообщений – не менее 1 Гбит/с.

2.2. Требования к программно-аппаратной платформе компонента сопоставления данных (ПО DQF: Federation)

2.2.1. Требования к аппаратной инфраструктуре

2.2.1.1. Конфигурация сервиса Узла

Процессор:

- Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);
- от 4 ядер.

ОЗУ:

- от 4 Гб.

Дисковое пространство:

- от 5 Гб.

2.2.1.2. Конфигурация БД Узла

Процессор:

- от 8 ядер.

ОЗУ:

- от 16 Гб.

Дисковое пространство:

- от 400 Гб.

2.2.1.3. Конфигурация сервиса Центра

Процессор:

– Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);

- от 8 ядер.

ОЗУ:

- от 8 Гб.

Дисковое пространство:

- от 5 Гб.

2.2.1.4. Конфигурация БД Центра

Процессор:

- от 16 ядер.

ОЗУ:

- от 32 Гб.

Дисковое пространство:

- от 200 Гб.

2.2.1.5. Конфигурация БД кэша Центра

Процессор:

- от 4 ядер.

ОЗУ:

- от 16 Гб.

Дисковое пространство:

- от 5 Гб.

2.2.2. Требования к программному обеспечению

Управление очередью сообщений:

- брокер сообщений (Kafka).

Сервер приложений:

- ПО Docker 20.10+;
- Операционная система с возможностью установки ПО Docker 20.10+.

СУБД:

- PostgreSQL 14+.

2.2.3. Требования к сетевой инфраструктуре

Предъявляются следующие требования к сетевой инфраструктуре:

- пропускная способность канала между сервером приложений и внешними информационными системами, направляющими запросы в ПО DQF, составляет не менее 1 Гбит/с;
- пропускная способность каналов сервером приложений и брокером сообщений – не менее 1 Гбит/с.