Публичное акционерное общество «Сургутнефтегаз»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | УТВЕРЖДАЮ |
| Первый заместитель  генерального директора  ПАО «Сургутнефтегаз»  по финансовым вопросам и налогам |  | Главный инженер –  первый заместитель  генерального директора  ПАО «Сургутнефтегаз» |
|  |  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_ В.Г.Баранков |  | \_\_\_\_\_\_\_\_\_\_ А.Н.Буланов |
| «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. |  | «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

на ЗАКУПКУ ТЕХНОЛОГИЧЕСКОЙ ПЛАТФОРМЫ

ДЛЯ ПРОРАБОТКИ КОММЕРЧЕСКИХ ПРЕДЛОЖЕНИЙ

С ПОЭТАПНОЙ ЗАКУПКОЙ НЕОБХОДИМЫХ ЛИЦЕНЗИЙ И УСЛУГ

2022 г.

Содержание

[1. Общие сведения 3](#_Toc109285033)

[1.1. Основание 3](#_Toc109285034)

[1.2. Термины и сокращения 3](#_Toc109285035)

[1.3. Цели 10](#_Toc109285036)

[2. Общие требования 10](#_Toc109285037)

[3. Требования к компонентам платформы 12](#_Toc109285038)

[3.1. Общие требования к компонентам и составу платформы 12](#_Toc109285039)

[3.2. Требования к документации на компоненты платформы 13](#_Toc109285040)

[3.3. Компоненты управления данными 13](#_Toc109285041)

[3.3.1. Компонент РСУБД промышленного уровня (на основе PostgreSQL) 13](#_Toc109285042)

[3.3.2. Компонент распределенной СУБД в оперативной памяти 14](#_Toc109285043)

[3.3.3. Компонент распределенное объектное хранилище на основе S3 15](#_Toc109285044)

[3.3.4. Компонент хранения/обработки больших объёмов на основе Hadoop 15](#_Toc109285045)

[3.3.5. Компонент массово параллельной СУБД на основе Greenplum 15](#_Toc109285046)

[3.4. Компоненты аналитики данных 15](#_Toc109285047)

[3.4.1. Компонент для исследования, визуализации данных и отчетности 15](#_Toc109285048)

[3.4.2. Компонент средств ETL включая поточную загрузку и оркестрацию 15](#_Toc109285049)

[3.4.3. Компонент анализа качества данных 15](#_Toc109285050)

[3.5. Компоненты интеграционного взаимодействия 16](#_Toc109285051)

[3.5.1. Компонент интеграции микросервисов Service-Mesh 16](#_Toc109285052)

[3.5.2. Компонент распределенного брокера сообщений для передачи данных 16](#_Toc109285053)

[3.5.3. Компонент событийной передачи и обработки данных 17](#_Toc109285054)

[3.5.4. Компонент передачи и обмен файлами 18](#_Toc109285055)

[3.5.5. Компонент управления API 18](#_Toc109285056)

[3.5.6. Компонент шардирование приложений 20](#_Toc109285057)

[3.5.7. Компонент анализа нагрузки и управление размещением микросервисов 20](#_Toc109285058)

[3.6. Компоненты эксплуатации и безопасности 20](#_Toc109285059)

[3.6.1. Компонент централизованного логирования 20](#_Toc109285060)

[3.6.2. Компонент кросс платформенный мониторинг 21](#_Toc109285061)

[3.6.3. Компонент управление настройками приложений 22](#_Toc109285062)

[3.6.4. Компонент управление аутентификацией и авторизацией 23](#_Toc109285063)

[3.6.5. Компонент регистрация и долговременное хранение событий Аудита 24](#_Toc109285064)

[3.7. Компоненты управления разработкой и контейнеризации 25](#_Toc109285065)

[3.7.1. Управление контейнерами 25](#_Toc109285066)

[3.7.2. Компонент управления разработкой включая инструменты CI/CD 25](#_Toc109285067)

[3.7.3. Компонент моделирования и выполнения Бизнес процессов 25](#_Toc109285068)

[3.7.4. Компонент формирования печатных форм 26](#_Toc109285069)

[3.7.5. Компонент исполнения фоновых заданий 26](#_Toc109285070)

[3.8. Компоненты управления инфраструктурой 26](#_Toc109285071)

[3.8.1. Требования к подсистеме виртуализации вычислительных ресурсов 26](#_Toc109285072)

[3.8.2. Требования к подсистеме управления виртуализированными ресурсами 26](#_Toc109285073)

[3.8.3. Требования к подсистеме управления конфигурациями и развертыванием 26](#_Toc109285074)

[3.8.4. Требования к подсистеме вычислительной сети 26](#_Toc109285075)

[4. Требования к оборудованию и системному программному обеспечению 26](#_Toc109285076)

[5. Требования по информационной безопасности 26](#_Toc109285077)

[6. Установка Платформы, обучение персонала 27](#_Toc109285078)

[7. Требования к гарантийному обслуживанию и технической поддержке 28](#_Toc109285079)

[8. Расходы на эксплуатацию Платформы 29](#_Toc109285080)

[9. Проект реализации предложенного решения 29](#_Toc109285081)

# Общие сведения

## Основание

Технические требования на закупку технологической платформы (далее – Платформа) подготовлены на основании пункта 3.1 постановляющей части протокола рабочего совещания от 24.05.2022 №01-14-27-33, утверждённого Первым заместителем генерального директора ПАО «Сургутнефтегаз» по финансовым вопросам и налогам В.Г.Баранковым 10.06.2022.

## Термины и сокращения

| **Сокращение/Термин** | **Наименование/Определение** |
| --- | --- |
| Авторизация | Процедура предоставления субъекту определенных прав доступа |
| Администратор | В настоящем документе: лицо, наделенное правами для осуществления деятельности в административной части АС |
| АРМ | Автоматизированное рабочее место |
| АС | Автоматизированная система |
| Аутентификация | Процедура проверки подлинности субъекта |
| БД | База данных |
| Бизнес-область | Прикладная область, объединяющая данные, микросервисы, компоненты и приложения определенного направления деятельности Общества |
| Бэклог | Список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации |
| ВМ | Виртуальная машина |
| Вектор изменений | Последовательность изменений прикладных объектов, передаваемых с целью репликации на дублирующее хранилище и в хранилище данных |
| Дефект  (программного обеспечения) | Отдельное несоответствие установленным требованиям, т.е. поведение функционала, являющегося отличным от заявленного в технической документации, переданной в эксплуатацию |
| Динамическая инфраструктура | средство управления облачной инфраструктурой с интерфейсом заказа и управления виртуальными серверами |
| Доступность  (ИТ-услуги) | Отношение фактического времени доступности ИТ-услуги (согласованного времени предоставления ИТ-услуги за вычетом простоев ИТ-услуги по причине инцидентов или технического обслуживания, не связанных непосредственно с предоставлением ИТ-услуги, но предоставление которой зависит от них) к согласованному времени предоставления ИТ-услуги |
| Заказчик,  Общество | ПАО «Сургутнефтегаз» |
| Интерфейс | Совокупность возможностей, средств, способов, методов и правил взаимодействия двух объектов, в частности человека с Системой, устройством или программой для обмена информацией между ними. |
| Инцидент  (технологический инцидент) | Незапланированное прерывание или снижение качества (включая доступность и производительность) осуществления бизнес-операций, предоставления ИТ-услуг, а также сбой элемента ИТ-инфраструктуры, который еще не оказал влияние на объект инцидента, в том числе вызванное отклонением фактического уровня качества данных от целевого |
| Импортозависимые технологии | Технологии чувствительные и зависимые от ограничений на их импорт |
| ИС | Информационная система |
| Катастрофоустойчивость | Обеспечение высокого уровня отказоустойчивости прикладной системы при отказе/катастрофе ЦОД целиком с учётом географического разнесения вычислительных ресурсов между двумя зданиями ЦОД |
| Композитный сервис | Набор работающих совместно (параллельно с существующими приложениями) микросервисов, обеспечивающих определенное решение |
| Компонент Платформы | Структурный элемент Платформы, обеспечивающий реализацию части функционала для разработки Приложений на Платформе |
| Контейнеризация | Метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных пространств пользователя вместо одного. |
| Кластер Контейнеризации,  Kubernetes,  K8S | ПО для оркестрации контейнеризированных приложений - автоматизации их развертывания, масштабирования и координации в условиях вычислительного кластера. |
| КТС | Комплекс технических средств |
| Микросервис  (сервис в системе с сервисно ориентированной архитектурой) | Отдельные программные компоненты в микросервисной архитектуре, характеризующиеся ограниченной функциональностью, идентифицируется строкой URL, с определенными и описанными интерфейсами взаимодействия.  Сервисы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определенных протоколах (HTTP, HTTP/2), соглашениях (REST, gPRC) и форматах обмена (JSON, XML, Protobuf). |
| Нагрузочный-стенд Платформы | Стенд нагрузочного тестирования - комплекс программных и технических средств, размещенный в облачной инфраструктуре, являющейся частью Платформы, предназначенный для сбора определения показателей производительности, оценки времени отклика в ответ на сформированные запросы разработанного Приложения и(или) его компонент Заказчиком с целью установления соответствия требованиям по нагрузке, предъявляемым для разработанного программного обеспечения |
| Омниканальность | Взаимная интеграция разрозненных фронтальных программных решений (каналов) коммуникации в единую систему, с целью обеспечения бесшовной и непрерывной коммуникации с клиентом |
| Отказоустойчивость | Свойство информационной системы или приложения сохранять свою работоспособность при отказе одного или нескольких компонент (сервисов, приложений, серверов, каналов связи и т.д.) |
| Платформа/Платформа разработки  (технологическая платформа) | Программно-аппаратная среда, предоставляющая функционально полный набор информационно-технологических сервисов, позволяющих эффективно создавать, развивать и эксплуатировать прикладное программное обеспечение |
| ПМИ | Программа и методика испытаний |
| ПО | Программное обеспечение |
| Пользователь | Сотрудник Общества |
| Пользователь Платформы | Участник команды разработки (инженер, разработчик, тестировщик, аналитик и др.), использующий Платформу в целях информационных систем и их компонентов. |
| Пользователь Платформы конечный,  Конечный Пользователь | Лицо использующее функции информационных систем и их компонентов, созданных и эксплуатирующихся на Платформе (далее - Конечный Пользователь). |
| ППО | Прикладное программное обеспечение, предназначенное для выполнения определенных задач и рассчитанное на непосредственное взаимодействие с пользователем |
| Платформа,  Технологическая платформа | Набор программных продуктов, предоставляющих совокупность функциональных возможностей и позволяющий обеспечить быстрое конструирование информационных систем из множества готовых компонентов. Решает задачи масштабирования и безопасности. Максимально снимает задачи технического уровня с разработчиков прикладной логики. |
| Приемочные испытания  Платформы | Комплекс мероприятий, определяемый в соответствии с программой и методикой испытаний, направленный на проверку передаваемой Платформы на предмет соответствия требований ТЗ. |
| Проксирование | Перехват объекта (web-трафик, сообщение и др.) с целью дальнейшей маршрутизации. |
| Проприетарное ПО | ПО являющееся собственностью авторов или правообладателей, а его использование Заказчиком определено и ограничено лицензионным соглашением |
| ПСИ | Приемо-сдаточные испытания |
| LOGON-стенд Платформы | Стенд приемо-сдаточных испытаний - комплекс программных и технических средств, размещенный в облачной инфраструктуре, являющейся частью Платформы, предназначенный для проведения испытаний разработанного на Платформе Приложения и(или) его компонент Заказчиком непосредственно перед его установкой на PROD-стенд. |
| Разработчик | АО «Национальные информационные технологии» |
| Ролевая модель | Набор прав доступа пользователя, определяющая границы полномочий в Системе. |
| РФ | Российская Федерация |
| Сервер | В контексте сетевой архитектуры «клиент-сервер»: программное обеспечение, являющееся поставщиком услуг заказчикам (клиентам). |
| Сервисы Платформы | IaaS, PaaS, Общеплатформенные сервисы |
| СУБД | Система управления базами данных. |
| СПО | Свободно-распространяемое ПО |
| ТЗ | Техническое задание |
| ТЗ, Техническое задание | Настоящее техническое задание на оказание услуг по передаче неисключительных прав и обеспечению функционирования, администрирования и бесперебойной работы программного обеспечения Платформы. |
| Узел | Совокупность функционально связанных элементов, предназначенных для выполнения определенных операций, например, преобразование, передача, хранение и управление информацией. |
| Фреймворк | Программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. |
| ЦОД | Центр хранения и обработки данных (англ. data center - Дата-центр) – специализированное здание для размещения (хостинга) серверного и сетевого оборудования. |
| Шардирование | Стратегия горизонтального масштабирования данных, при которой части данных размещаются на разных хостах кластера. Используется для работы с большими объемами данных и увеличения пропускной способности СУБД. |
| Штатный режим | Основной режим функционирования, при котором:  – исправно функционируют серверное программное обеспечение и технические средства;  – исправно функционирует системное, базовое и прикладное программное обеспечение.  Для обеспечения данного режима выполняются требования и выдерживаются условия эксплуатации программного обеспечения и комплекса технических средств, указанные в соответствующей технической документации.  Режим, при котором Платформа и(или) Приложения должны эксплуатироваться 24 часа в сутки, 7 дней в неделю и 365(6) дней в году. |
| ЭЦП | Электронная цифровая подпись |
| API | Application Programming Interface (интерфейс программирования приложений) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, микросервисом) или операционной Системой для использования во внешних программных продуктах |
| API First | Архитектура, для которой API пользователь является основным пользователем приложения, это означает, что этот API должен иметь наивысший приоритет, быть полным, адаптивным и хорошо документированным. |
| BPMN | (англ. Business Process Model and Notation) Язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и исполнением бизнес-процесса.  BPMN 2.0 – версия нотации, содержащая набор условных обозначений и их описания в формате XML для моделирования бизнес-процессов. |
| Cloud-native | Подход к созданию и выполнению приложений, использующий преимущества облачной модели. Обычно такие приложения строятся как набор слабо-зависимых друг от друга микросервисов с использованием технологии контейнеризации, слабо связанных между собой и упакованных в контейнеры. |
| CI,  Continuous Integration | Непрерывная интеграция - практика разработки программного обеспечения, которая заключается в постоянном слиянии рабочих копий программного кода в общую основную ветвь разработки (до нескольких раз в день) и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем. Переход к практике непрерывной интеграции позволяет снизить трудоемкость за счет обнаружения и устранения ошибок и противоречий на ранних этапах. |
| CD,  Continuous Delivery | Непрерывная доставка - Подход к разработке программного обеспечения, при котором программное обеспечение производится короткими итерациями, гарантируя, что ПО является стабильным и может быть передано в эксплуатацию в любое время, а передача его не происходит вручную[[1]](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%B0%D1%8F_%D0%B4%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0#cite_note-1). Целью является сборка, тестирование и релиз программного обеспечения с большей скоростью и частотой. Подход позволяет уменьшить стоимость, время и риски внесения изменений путём более частных мелких обновлений в продакшн-приложение. |
| CIFS | Common Internet File System, также называют SMB (Server Message Block) — сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия |
| CPU | (от англ. Central Processing Unit) — центральный процессор |
| DEV-стенд Платформы | (от англ. Development, разработка, создание) - комплекс программных и технических средств, размещенный в облачной инфраструктуре, являющейся частью Платформы, предназначенный для разработки Приложений Заказчиком с использованием установленных на стенде компонентов Платформы (включая разработку, отладку, сборку и предварительное тестирование). |
| DevOps | Методология активного взаимодействия специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимная интеграция их рабочих процессов друг в друга для обеспечения качества продукта. Предназначена для эффективной организации создания и обновления программных продуктов и услуг. Основана на идее тесной взаимозависимости создания продукта и эксплуатации программного обеспечения, которая прививается команде как культура создания продукта (акроним от англ. development и operations; по-русски обычно произносится как «дево́пс»). |
| gRPC | (от англ. general-purpose Remote Procedure Call) - высокопроизводительный фреймворк разработанный компанией Google для вызова удаленных процедур, в качестве транспорта использует HTTP/2, в качестве инструмента описания типов данных и сериализации используется Protobuf. |
| GUI | (от англ. Graphic User Interface) – графический интерфейс пользователя |
| HTTP | (англ. HyperText Transfer Protocol) — протокол прикладного уровня передачи данных (как HTML так и произвольных данных) по технологии «клиент-сервер». |
| HTTP/2 | Вторая крупная версия сетевого протокола HTTP, используемая для доступа к World Wide Web. |
| HDD | Hard Disk Drive — жесткий диск |
| High-IOPS SSD | Solid-State Drive с высоким показателем IOPS,т.е. с высоким показателем количества операций ввода/вывода в секунду |
| IaaS  (Инфраструктура как услуга) | (от англ. Infrastructure-as-a-Service) — одна из моделей обслуживания, применяемая в области в облачных вычислениях, по которой потребителям предоставляются фундаментальные информационно-технологические ресурсы — виртуальные серверы с заданной вычислительной мощностью, операционной системой и доступом к сети. |
| IaC  (Инфраструктура как код) | (от англ. Infrastructure-as-Code) — подход управления и описания инфраструктуры через конфигурационные файлы и скрипты развертывания вместо ручного управления. |
| IP | (от англ. Internet Protocol) Протокол маршрутизации сетевого уровня стека TCP/IP. Протокол, который объединяет отдельные компьютерные сети в единую сеть. Неотъемлемой частью протокола является адресация сети (Уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP. В единой компьютерной сети требуется глобальная уникальность адреса; в случае работы в локальной сети требуется уникальность адреса в пределах сети). |
| ITIL | (от англ. IT Infrastructure Library в переводе с англ. библиотека инфраструктуры информационных технологий) – это библиотека, в которой описаны лучшие практические способы организации работы структурных единиц или всех компаний, задействованных в предоставлении услуг в сфере информационных технологий. |
| JPA  (Java API) | Java Persistence API - спецификация архитектуры, которая предоставляет возможность сохранять в удобном виде Java-объекты в базе данных. |
| JSON | (от англ. JavaScript Object Notation) - текстовый формат обмена данными, основанный на JavaScript, легко читается людьми, независим от языка программирования (для многих языков существует готовый код для создания и обработки данных в формате JSON). |
| Low Latency Disk | (LL NVME) диски — SSD диски NVME (Non-Volatile Memory Express) с низким, по сравнению с обычным SSD диском или даже High-IOPS SSD диском, временем отклика |
| NFS | Network File System — протокол сетевого доступа к файловым системам |
| Multitenancy | (от англ. с «множественная аренда», мультиарендность) — элемент архитектуры программного обеспечения, где единый экземпляр приложения, запущенного на сервере, обслуживает множество организаций-клиентов. Мультиарендность противопоставляется архитектуре из множественных экземпляров (англ. multiinstance), где для каждой организации-клиента создаются отдельные программные экземпляры. В мультиарендной архитектуре программные приложения работают одновременно с несколькими конфигурациями и наборами данных нескольких организаций, а каждая организация-клиент работает со своим экземпляром виртуального приложения, видя только свою конфигурацию и свой набор данных. |
| MVCC | (от англ. multiversion concurrency control) — управление параллельным доступом посредством многоверсионности — механизм СУБД для обеспечения параллельного доступа к базам данных, при котором пишущие транзакции не блокируют читающих, и читающие транзакции не блокируют пишущих. |
| OpenId  (OpenId Connect) | Открытый стандарт децентрализованной системы аутентификации, представляющей пользователю возможность создать единую учетную запись для аутентификации на множестве не связанных друг с другом web-ресурсов, используя отдельного провайдера аутентификации (сервера авторизации).  OpenId Connect - третье поколение OpenId-технологии. |
| ORM | (от англ. Object-Relational Mapping, рус. объектно-реляционное отображение, или преобразование) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». |
| OSI | Сетевая модель OSI (The Open Systems Interconnection model) |
| OTT | (One-time token) Одноразовый токен, действующий только для одного сеанса аутентификации и(или) промежутка времени. |
| PaaS | Platform as a Service, модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию информационно-технологических платформ: операционных систем, систем управления базами данных, связующему программному обеспечению, средствам разработки и тестирования, размещённым у провайдера. В этой модели вся информационно-технологическая инфраструктура, включая вычислительные сети, серверы, системы хранения, целиком управляется провайдером, провайдер же определяется набор доступных для потребителей видов платформ и набор управляемых параметров платформ, а потребителю предоставляется возможность использовать платформы, создавать их виртуальные экземпляры, устанавливать, разрабатывать, тестировать, эксплуатировать на них прикладное программное обеспечение, при этом динамически изменяя количество потребляемых вычислительных ресурсов |
| PROD-стенд Платформы | (от англ. Production, производство) - комплекс программных и технических средств, размещенный в облачной инфраструктуре, являющейся частью Платформы, предназначенный для промышленной эксплуатации разработанного Приложения и(или) его компонент Заказчиком. |
| Protobuf | (от англ. Protocol Buffers) протокол сериализации для передачи структурированных данных между клиентом и сервером. Используя строгую типизацию полей и бинарный формат для передачи структурированных данных, protobuf потребляет меньше ресурсов. Альтернатива JSON/XML. |
| RAM | Random Access Memory — оперативная память |
| REST | (от англ. ReprEsentational State Transfer) архитектурный стиль взаимодействия компонентов распределенного приложения в сети. REST представляет собой согласованный набор условий, учитываемых при проектировании распределенной гипермедиа-системы.  Традиционно реализации используют такие стандарты, как HTTP, URL, JSON и XML. |
| Round-robin | Алгоритм распределения нагрузки распределенной вычислительной системы методом перебора и упорядочения ее элементов по круговому циклу. |
| RPC-вызовы | (от англ. Remote Procedure Call) Удаленный вызов процедур— класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (на удаленных компьютерах, либо в независимой сторонней системе на том же устройстве). |
| S3 хранилище | Облачное хранилище объектов, любого объема, с высоким уровнем надежности и доступности (например, файлов любого типа). |
| SLA | Термин методологии ITIL (от англ. Service Level Agreement, SLA), обозначающий формальный договор между заказчиком (в рекомендациях ITIL заказчик и потребитель — разные понятия) услуги и ее поставщиком, содержащий описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги. |
| SMTP | (от англ. Simple Mail Transfer Protocol) Простой протокол передачи почты, широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP. |
| SOA | (от англ. Service-Oriented Architecture) Сервис-ориентированная архитектура – подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимодействия. |
| SSD | Solid-State Drive — твердотельный накопитель, энергонезависимое немеханическое запоминающее устройство на основе микросхем памяти, альтернатива HDD |
| TCP | (от англ. Transmission Control Protocol) Протокол передачи данных в коммуникационной сети/сети интернет. |
| TCP/IP | Сетевая модель передачи данных (набор протоколов/соглашений), представленных в цифровом виде, например, HTTP/HTTPS, FTP, POP3, SMTP, TELNET. |
| TESТ-стенд Платформы,  QAS-стенд Платформы | (от англ. Testing, Тестирование) - комплекс программных и технических средств, размещенный в облачной инфраструктуре, являющейся частью Платформы, предназначенный для интеграционно-функционального тестирования разработанного Приложения и(или) его компонент Заказчиком. Характеризуется наличием настроенных адаптеров со смежными системами интеграционного ландшафта (или их эмуляторов). |
| UI | (от англ. User Interface) Пользовательский интерфейс. |
| UML | UML (англ. Unified Modeling Language) Унифицированный язык моделирования) — язык графического описания для объектного моделирования при разработке программного обеспечения, используется для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. |
| URL | (от англ. Uniform Resource Locator) - система унифицированных адресов электронных ресурсов, единообразный определитель местонахождения ресурса (в том числе файла). |
| WorkFlow | Совокупность этапов процесса, последовательный план выполнения отдельных задач/работ. |
| XML | (англ. eXtensible Markup Language) –текстовый формат, предназначенный для хранения структурированных данных для обмена информацией между информационными системами. |

**Катастрофой** для прикладной системы, т.е. прекращением ее функционирования, считается:

* выход из строя сервера, на котором работает прикладная система;
* выход из строя дисковой подсистемы, на которой расположена актуальная информация прикладной системы;
* выход из строя системного программного обеспечения или базы данных;
* выход из строя коммуникационного оборудования, которое используется в работе прикладной системы;
* выход из строя системы электропитания одного из ЦОД;
* выход из строя системы охлаждения одного из ЦОД;
* выход из строя одного из ЦОД (по любой причине, в т.ч. пожар, разрушение, затопление, и т.п.).

## Цели

Платформа предназначена для создания, развития, развертывания и эксплуатации прикладного программного обеспечения ПАО «Сургутнефтегаз» в отказоустойчивой и катастрофоустойчивой среде.

# Общие требования

* 1. Предложить варианты конфигурации технологической платформы (далее - Платформа) согласно данным приложений X. (Если компоненты платформы будут вынесены в отдельные приложения. Если нет, удалить)
  2. При проектировании Платформы должен быть реализован принцип централизованного управления, функционал которого в том числе включает формирование различных отчетов и статистических данных.
  3. Платформа должна обеспечивать требования к катастрофоустойчивости и доступности информационных систем:
     1. Время восстановления работоспособности прикладных систем – RTO (Recovery Time Objective):
* продуктивные системы – до 4 часов;
* системы разработки, тестирования, предпродуктивные – до 5 календарных дней;
  + 1. Уровень восстановления данных – RPO (Recovery Point Objective). Максимально допустимый уровень потери данных при катастрофе системы:
* для продуктивных систем – до 1 мин.;
* системы разработки, тестирования – до 24 часов;
  1. Платформа должна обеспечивать резервное копирование и восстановление информационных систем и компонентов Платформы встроенными средствами или внешними системами резервного копирования для обеспечения требований п.2.3.
  2. Платформа должна обеспечивать возможность добавления новых вычислительных узлов без остановки и сбоев в функционировании информационных систем.
  3. Платформа должна обеспечивать возможность вывода вычислительных узлов из эксплуатации и освобождение их от нагрузки без остановки и сбоев в функционировании информационных систем.
  4. Приоритетным системным программным обеспечением считать операционную систему «Альт Сервер», РСУБД – PostgreSQL совместимые их разновидности.
  5. Платформа должна обеспечивать наличие компонента, отвечающего за доступ и безопасность, который обеспечивает идентификацию, аутентификацию, авторизацию пользователей, регистрацию событий аудита.
  6. Платформа должна обеспечивать наличие среды виртуализации вычислительных ресурсов.
  7. Приоритетной средой исполнения приложений Платформы должна являться среда Контейнеризации.
  8. Платформа должна обеспечивать автоматизацию управления жизненным циклом контейнеров и сервисов: возможность создания, развертывания, запуска, управления приложениями в кластерной контейнерной среде, планирование ресурсов, управление масштабируемостью, балансировку нагрузки.
  9. Платформа должна обеспечивать наличие компонента мониторинга, предназначенного для сбора, хранения, предоставления информации о текущем состоянии платформы (компоненты платформы, аппаратные ресурсы платформы).
  10. Платформа должна обеспечить наличие компонента логирования для централизованного сбора и отображения журналов компонентов платформы (журнал доступа, журнал аудита, системный журнал, журнал приложений).
  11. Платформа должна иметь решения по интеграции в ИТ-инфраструктуру Заказчика. Данные решения должны быть иметь документацию с описанием протоколов и общих подходов для их использования.
  12. Платформа должна обеспечивать исполнение решений и приложений:
      1. разработанных на основе и технологиях платформы;
      2. сторонних контейнеризированных приложений;
      3. сторонних виртуализированных приложений;
      4. гибридных приложений (сочетание 2.15.1-2.15.3).
  13. Платформа должна позволять подключать сторонние приложения к общеплатформенным компонентам аутентификции, авторизации, мониторинга, аудита, логирования и т.д.
  14. Платформа должна поддерживать подход «инфраструктура как код» в том числе включать в себя инструменты и «скрипты» по автоматизированному развертыванию компонент и приложений на основе Terraform, Ansible, Helm или подобных технологий с возможностью расширения Заказчиком.
  15. Платформа должна обеспечивать наличие компонента антивирусной защиты (Касперский).
  16. Все предлагаемые программные решения должны быть последних версий, стабильно функционировать и быть совместимыми между собой.
  17. Платформа должна поддерживать много-ландшафтные инсталляции с возможностью переноса изменений и настроек приложений по ландшафту.
  18. Платформа должна функционировать в полном объеме включая все процессы без прямого доступа к сети Интернет.

# Требования к компонентам платформы

## Общие требования к компонентам и составу платформы

* + 1. Все компоненты платформы должны предусматривать неограниченное по количеству инсталляций, пользователей, вычислительным или другим ресурсам использование в Обществе или его дочерних Обществах (см. список дочерних обществ в Приложение 1).
    2. На все компоненты должна предоставляться стандартная и инцидентная техническая поддержка (см. [Требования к гарантийному обслуживанию и технической поддержке](#_Требования_к_гарантийному)).
    3. Все компоненты платформы должны быть разработаны на основе Открытого и/или Свободно-распространяемого ПО и библиотек и не содержать импортозависимых компонент на момент поставки платформы.
    4. Сторонние проприетарные компоненты, в случае наличия, должны поставляться на условиях сублицензирования, принцип единого Поставщика.
    5. Компоненты платформы должны быть слабосвязанные, но при этом интегрированы.
    6. Поставщик должен предоставить описание связанности (зависимость) компонент платформы и отразить в документации на платформу в целом.
    7. Платформа должна включать следующие основные компоненты:
       1. Компоненты управления данными
* Компонент РСУБД промышленного уровня (на основе PostgreSQL), включая инструменты архивирования и репликации;
* Компонент распределенной СУБД в оперативной памяти по технологии DataGrid;
* Компонент распределенное объектное хранилище на основе S3;
* Компонент хранения и обработки больших объёмов данных на основе Hadoop;
* Компонент массово параллельной СУБД на основе Greenplum;
  + - 1. Компоненты аналитики данных:
* Компонент для исследования и визуализации данных в том числе формирования отчетности;
* Компонент средств ETL включая поточную загрузку данных и оркестрацию загрузок;
* Компонент анализа качества данных;
  + - 1. Компоненты интеграционного взаимодействия
* Компонент распределенного брокера сообщений для передачи и репликации потоков данных;
* Компонент событийной передачи и обработки данных;
* Компонент передачи и обмен файлами;
* Компонент интеграции микросервисов Service-Mesh;
* Компонент управления API;
* Компонент шардирование приложений;
* Компонент анализа нагрузки и управление размещением микросервисов;
  + - 1. Компоненты эксплуатации и безопасности
* Компонент централизованного логирования;
* Компонент кросс платформенный мониторинг;
* Компонент управление настройками приложений;
* Компонент управление аутентификацией и авторизацией;
* Компонент регистрация и долговременное хранение событий Аудита;
  + - 1. Компоненты управления разработкой и контейнеризации
* Компонент управления контейнерами;
* Компонент управления разработкой включая инструменты CI/CD;
* Компонент моделирования и выполнения Бизнес процессов;
* Компонент формирования печатных форм;
* Компонент исполнения фоновых заданий;
  + - 1. Компоненты управления инфраструктурой
* Компонент управления виртуализацией;

## Требования к документации на компоненты платформы

* + 1. Каждый компонент при поставке должен сопровождаться пакетом документации как минимум включающим:
* Требование к функционированию компонента и совместимости в том числе с аппаратным обеспечением, требования и условия эксплуатации;
* Руководство по развертыванию и обновлению;
* Руководство по администрированию и настройке;
* Руководство по эксплуатации и обслуживанию;
* Руководство по сайзингу прикладных решений на основе компонента;
* Расширенное общее и функциональное описание, руководство по использованию, руководство пользователя;
  + 1. Отдельные виды документов могут быть совмещены в один документ.
    2. Для нескольких компонент платформы может поставляться один документ определенного вида вплоть до документа по платформе в целом.
    3. Документация обновляется поставщиком одновременно с обновлением ПО компонента.

## Компоненты управления данными

### Компонент РСУБД промышленного уровня (на основе PostgreSQL)

Компонент Платформы должен обеспечивать управление высокодоступными, отказоустойчивыми объектно-реляционными базами данных в приложениях любого класса критичности, включая:

* поддержку стандарта SQL (включая работу с аналитическими функциями, общими табличными выражениями и рекурсивными запросами);
* обеспечение реализации атомарности, согласованности, изолированности (уровня READ COMMITED), надежности при обработке и хранении информации;
* поддержку транзакций с использованием версионности данных (MVCC) для обеспечения конкурентной работы с данными;
* поддержку ссылочной целостности данных;
* поддержку горячего (без остановки работы базы данных) резервного копирования, в том числе с ведомых узлов, в случае использования кластеров данных (функциональность снятия резервной копии не должна различаться между ведущим и ведомым узлами);
* поддержку журналирования работы для возможности восстановления после сбоев на требуемый момент времени;
* наличие встроенного процедурного языка для реализации хранимых подпрограмм;
* поддержку реализации пользовательских типов данных (включая табличные) и работы с ними в запросах и хранимых подпрограммах;
* поддержку последовательностей;
* поддержку материализованных представлений;
* поддержку составных, функциональных и частичных индексов;
* поддержку секционированных таблиц;
* поддержку работы с часовыми поясами и возможность хранения времени с точностью до миллисекунд;
* поддержку создания триггеров на операции DDL для таблиц и представлений;
* поддержку создания триггеров на операции DML для баз данных;
* поддержку либо возможность реализации аудита действий пользователей, включая подключения к базе данных и отключения от неё;
* функционал по управлению пользовательскими блокировками;
* функционал по управлению пользовательскими сессиями.

Встроенные либо реализуемые дополнительным функционалом возможности:

* работы со внешними, по отношения в БД данными, в частности: файлы (чтение и запись), сторонние базы данных (выполнение DML);
* работы с данными ГИС;
* организации кластеров данных (в том числе отказоустойчивых с автоматическим либо ручным переключением) с синхронной/асинхронной репликацией данных;
* выполнения внутренних заданий по расписанию.

Желательные возможности:

* выполнение кластеризации (упорядочивания) данных в таблице;
* сжатие данных на уровне страниц/таблиц или табличных пространств;
* проведение инкрементального резервного копирования;
* сжатие резервных копий;
* проверка целостности резервных копий;
* поддержка шифрования данных (как внутри БД, так и пересылаемых между СУБД и клиентом).

Необходимо наличие инструментов администрирования (работа с пользователями и привилегиями, анализ работы базы данных, просмотр свойств объектов, оптимизация и устранение проблем).

### Компонент распределенной СУБД в оперативной памяти

Компонент Платформы должен обеспечивать надежное распределенное хранение данных в оперативной памяти и распределенную обработку этих данных, в том числе:

* надежное хранение данных в оперативной памяти с журналированием на диск (нулевая потеря данных как при выходе из строя отдельных узлов, так и при полной аварийной остановке кластера);
* обеспечение доступа к данным при помощи API и при помощи языка SQL;
* обеспечение распределенной обработки данных (запуск кода на узле, хранящем необходимые данные, либо параллельный запуск кода на группе узлов кластера, обобщение результатов выполнения);
* консистентное резервное копирование данных кластера с возможностью восстановления данных на всех узлах кластера.

### Компонент распределенное объектное хранилище на основе S3

### Компонент хранения/обработки больших объёмов на основе Hadoop

Компонент должен обеспечивать хранение структурированных и неструктурированных данных в объемах, исчисляемых петабайтами, а также высокую скорость обработки данных с учетом распределенной модели исполнения запросов. В том числе компонент должен обеспечивать:

* работу с распределенной файловой системой для хранения больших объемов данных, поблочно распределенных между узлами вычислительного кластера;
* репликацию данных на уровне узлов кластера для повышения отказоустойчивости файловой системы;
* работу с распределенными параллельными вычислениями при обработке больших объемов данных;
* доступ к хранимым данным средствами SQL и NoSQL;
* поддержку пакетной и потоковой загрузки данных;
* настройку ограничения доступа к данным в распределенной файловой системе;
* инструменты для планирования заданий и управления вычислительными ресурсами кластеров;
* инструменты для разворачивания, управления и мониторинга вычислительных кластеров;
* хранение неструктурированных данных (изображения, видео файлы, аудио файлы, скан-копии документов, интернет-статьи, электронные письма, файлы всех типов).

### Компонент массово параллельной СУБД на основе Greenplum

## Компоненты аналитики данных

### Компонент для исследования, визуализации данных и отчетности

### Компонент средств ETL включая поточную загрузку и оркестрацию

* Компонент должен предоставлять набор сервисов для настройки, обработки и передачи данных между системами. Обеспечивает агрегацию данных из различных источников для наполнения хранилищ данных в соответствии с потребностью бизнес-модели, предназначен для:
* поддержки множества структурированных и неструктурированных форматов исходных данных для загрузки в том числе прямое чтение из СУБД и CSV;
* поддержки стандартных протоколов работы и доступа к данным, таких как SFTP, KAFKA, SSL, HTTPS, SSH, HDFS, SQL;
* Компонент должен поддерживать:
* кластеризацию с возможностью работы на нескольких узлах для повышения производительности обработки данных;
* преобразования потоков данных в том числе в формат PARQUET из CSV;
* управления множеством параллельных потоков данных;
* графический пользовательский интерфейс для настройки и управления потоками данных;

### Компонент анализа качества данных

Компонент анализа качества данных должен обеспечивать:

* инструменты для настройки проверок и бизнес-правил, которые позволяют контролировать качество данных;
* подключение функционала контроля качества данных к процессу пакетной загрузки данных;
* создание, редактирование проверок качества данных на базе собственных бизнес-правил или параметризация готовых шаблонов проверок;
* запуск пользовательских проверок качества данных;
* визуализацию текущего состояния проверок качества данных, а также хранение версионности проверок.

## Компоненты интеграционного взаимодействия

### Компонент интеграции микросервисов Service-Mesh

Компонент Платформы должен обеспечивать:

* возможность создания различных стратегий балансировки запросов между несколькими экземплярами или версиями приложения (равномерно(round-robin), с использованием весов), с помощью конфигурационных файлов kubernetes;
* возможность создания правил маршрутизации запросов на основе различных стратегий (поля заголовка, части uri), с помощью конфигурационных файлов kubernetes;
* возможность создания политик доступа к опубликованному сервису на основе данных principal, с помощью конфигурационных файлов kubernetes;
* квотирование максимального количества одновременных подключений к приложению с помощью конфигурационных файлов kubernetes;
* развертывание компонент платформы в виде отдельных элементом (в том числе режим sidecar) без влияния на прикладной код;
* динамическое изменение интеграционных политик (правил маршрутизации, политик доступа, стратегия балансировки) в режиме runtime (без остановки приложения);
* формирование метрик трассировки запросов и мониторинга компонент.

### Компонент распределенного брокера сообщений для передачи данных

Компонент должен обеспечивать:

* возможность работы с распределенной очередью сообщений;
* возможность работы с распределенным брокером сообщений Kafka;
* маппинг на основе xls и JSON Template.
* В части работы с очередями сообщений должно обеспечиваться:
* чтение сообщений из очередей и их отправку в соответствующие сервисы по протоколу gRPC;
* gRPC API отправки JMS сообщений в очереди;
* работу в режимах:
  + асинхронном - без ожидания ответа;
  + синхронном - с ожиданием ответа и отправкой его инициатору;
* подключение к нескольким брокерам очередей;
* логирование событий приема и отправки сообщений;
* формирование трассировочной информации;
* возможность подключения к брокеру очереди с использованием аутентификации по SSL-сертификатам;
* формирование метрик мониторинга.
* В части работы с брокером сообщений Kafka:
* чтение сообщений из топиков Kafka и их отправку в соответствующие сервисы по протоколу gRPC;
* запись сообщений в топики Kafka посредством gRPC API;
* подключение к нескольким брокерам Kafka, в том числе и с помощью механизма аутентификации по SSL-сертификатам;
* логирование событий приема и отправки сообщений;
* авторизацию вызовов;
* формирование метрик мониторинга.

В части маппинга:

* выполнение функции преобразование формата сообщения запроса xml на основе xsl, с последующим вызовом сервиса назначения, получения и преобразования ответа, возврат его потребителю;
* выполнение функции преобразование формата сообщения запроса json на основе jsonT, с последующим вызовом сервиса назначения, получения и преобразования ответа, возврат его потребителю.

### Компонент событийной передачи и обработки данных

Компонент событийной передачи Платформы должен обеспечить:

* возможность создания новых потоков событий со следующими параметрами:
  + срок хранения событий в транспортном слое и/или максимальный размер потока событий;
  + число копий события для обеспечения отказоустойчивости публикации событий;
  + наименование потока событий согласно правилам наименования;
  + максимальный размер события для данного потока;
  + допустимый параллелизм обработки события;
* возможность назначения прав на публикацию и подписку по каждому потоку событий на основании модели авторизации, базирующейся на полном наименовании сертификата, используемого при подключении;
* соединение с узлами передачи событий на основе протокола TLS;
* возможность публикации события (plaintext, json и xml-нотации) с сохранением его в транспортном слое;
* возможность потребления события из потока по следующим правилам:
  + потребители могут объединяться в группы для обеспечения масштабирования обработки;
  + все активные экземпляры потребителей, принадлежащих одной группе, получают суммарно весь актуальный поток событий;
  + каждая активная группа потребителей получает весь актуальный поток событий;
* удаление события по достижению заданного срока хранения/максимального размера потока.

Компонент потоковой обработки событий Платформы должен обеспечивать:

* возможность создания новых заданий на потоковую обработку событий с учетом текущей емкости экземпляра потоковой обработки событий;
* возможность запуска/остановки заданий на потоковую обработку событий;
* возможность запускать несколько параллельно исполняющихся идентичных задач на потоковую обработку событий;
* возможность описания преобразований (трансформация, фильтрация, агрегация) одного или нескольких входящих потоков, имеющих объектную модель и публикуемых в формате XML или JSON, на псевдоязыке, с порождением одного или нескольких результирующих потоков событий. Псевдоязык должен поддерживать следующие операнды:
  + математические (сложение, вычитание, умножение, деление, целочисленное деление);
  + строковые (конкатенация, получение подстроки, разделение на несколько строк, проверка префикса/суффикса, поиск по регулярным выражениям);
  + условный оператор;
  + цикл;
  + работа с датой-временем (сравнение, получение текущей);
  + работа с последовательностями (создание, расширение).

### Компонент передачи и обмен файлами

Компонент Платформы должен обеспечить:

* доставку файлов с возможностью временного хранения с использованием внешнего хранилища по протоколу S3;
* взаимодействие с компонентом файловой передачи по протоколу gRPC;
* шифрованную передачу файлов.

### Компонент управления API

Компонент должен обеспечивать ведение метаинформации:

* функции сбора, агрегации и предоставления потребителям метаинформации о разработанных микросервисах приложений;
* импорт конфигураций микросервисов;
* предоставление шлюзу конфигураций микросервисов;
* регистрация экземпляров шлюзов;
* получение статуса экземпляра шлюза;
* получение параметров настройки экземпляра шлюза;
* получение перечня выведенных из балансировки узлов микросервисов на шлюзе;
* просмотр, изменение и удаление конфигураций микросервисов.

Компонент Платформы должен обеспечить единую входную точку для внешних систем к микросервисам приложений внутреннего контура в микросервисной архитектуре, в том числе:

* получение конфигураций из реестра микросервисов для прикладной бизнес-области;
* изменение конфигурации шлюза в среде исполнения;
* балансировка запросов между группами узлов микросервисов и узлами микросервисов внутри группы;
* определение доступности узла микросервисов (активная и пассивная диагностика работоспособности), при недоступности узла микросервисов узел выводится из балансировки, при этом запросы на него не направляются;
* стабильное подключение сессии приложения к узлу микросервисов (sticky session);
* квотирование максимального количества запросов/сек к группе балансировки микросервиса.

Компонент Платформы должен обеспечить доступ к микросервисам приложений во внутреннем контуре сети для внутренних приложений в микросервисной архитектуре:

* reverse proxy для http и https запросов (приложение выставляет свои rest end-points на шлюзе - шлюз при поступлении запросов на эти end-points проксирует на сервера с инсталляцией приложения);
* настройку системных параметров соединений:
  + таймауты соединения клиента;
  + таймауты соединения с upstream сервером;
  + максимальный размер тела запроса, ответа.
* пассивную проверка работоспособности;
* активную проверку работоспособности:
  + http и https протоколы;
  + поддержку платформенного агента проверки работоспособности и его статусной модели;
  + мониторинг в режиме реального времени статуса активной работоспособности по всем серверам;
  + возможность проверки работоспособности и доступности приложения в кластере;
* настраиваемое добавление http заголовков в запросы (entryPoint для статики и другие):
  + сквозное проксирование заголовков (для трекинга запросов);
  + работа заголовков X-Forwarded-For;
* управление выбором минорной и мажорной версии микросервиса. Настраиваемая маршрутизация запроса – выбор группы балансировки для запроса по заданной комбинации значений определенных http заголовков;
* mTLS:
  + кеширование сессий;
  + правила по проверке CN потребителя;
* функционал slow start – плавное нарастание нагрузки при включении сервера (circuit breaker);
* поддержку алгоритмов распределения запросов в группе балансировки:
  + round robin;
  + consistent hash по выбранному параметру запроса, например заголовок или cookie;
  + sticky по выбранному параметру запроса, например заголовок или cookie. Выбор алгоритма расчета значения sticky cookie; md5, sha, text;
* квотирование запросов (rate limits):
  + на весь сервер;
  + к отдельному микросервису в разрезе других микросервисов;
  + к отдельному микросервису в разрезе каналов, tenantCode:
    - квота на сумму запросов к микросервису;
    - квоты по произвольным http заголовкам.
* отправку метрик работы в систему мониторинга, включая координаты источника запроса и назначения запроса (трекинга запросов);
* отправку лога работы шлюза в систему журналирования, настраиваемый список атрибутов записи лога;
* функционал web сервера – раздача статики, включая настройку параметров кеширования статики на клиенте;
* пользовательский интерфейс для мониторинга текущего состояния;
* пользовательский интерфейс для просмотра и изменения параметров кластера шлюзов, online изменение параметров на шлюзе.

### Компонент шардирование приложений

Компонент Платформы должен обеспечить корректную маршрутизацию запросов к приложениям в геораспределенной топологии, а также с учетом возможности одновременной эксплуатации нескольких контуров/инсталляций Платформы, в том числе:

* маршрутизацию http и https запросов клиентов в рассчитываемый по правилам маршрутизации элемент топологии. Элементы топологии могут находиться в разных ЦОД (георезервирование), а также обслуживать запросы в рамках определенной инсталляции/контура Платформы;
* маршрутизацию http и https запросов с возможностью учета шардирования данных;
* получение маршрута назначения для определенных клиентов;
* при маршрутизации запросов должен учитываться статуса режима резервирования элементов топологии;
* кэширование маршрутов в оперативной памяти (кэш должен быть вытесняемым, его размер настраиваемым);
* установку правил выбора маршрута для входящего запроса;
* возможность сквозного проксирования http заголовков для трекинга запросов и возможность работы заголовков X-Forwarded-For;
* осуществлять отправку метрик работы в систему мониторинга;
* осуществлять отправку лога работы в систему журналирования, при этом должна быть возможность настройки уровня записи лога.

### Компонент анализа нагрузки и управление размещением микросервисов

## Компоненты эксплуатации и безопасности

### Компонент централизованного логирования

Компонент должен обеспечивать:

* запись логов и их хранение в централизованном хранилище;
* централизованное хранилище – это хранилище поддерживающее репликацию, горизонтальное масштабирование и индексирование собираемых журналов;
* транспорт должен обеспечивать возможности буферизирования сообщений на случай сбоев в подсистеме хранения;
* сбои на стороне журналирования не должны останавливать функционирования приложения потребителя;
* в случае недоступности сервиса журналирования, допускается частичная потеря записей журналов;
* запись и хранение цепочек вызовов приложений (и(или) микросервисов) в централизованном хранилище;
* каждый новый запрос системы помечается идентификатором traceId. В рамках запроса выделяются шаги, которые помечаются своим spanId;
* сбои на стороне сервиса журналирование не должны останавливать функционирования приложения потребителя;
* в случае недоступности сервиса журналирование, допускается частичная потеря записей спанов;
* транспорт должен обеспечивать возможности буферизирования сообщений на случай сбоев в подсистеме хранения;
* инструментарий просмотра и поиска событий журналирования:
* административное рабочее место (далее – АРМ) должно поддерживать поиск по телу сообщения. В фильтре поиска обязательно должно быть заполнено значение времени, за которое осуществляется поиск записей;
* АРМ должен поддерживать поиск по дополнительному атрибутному составу;
* АРМ должен поддерживать возможность сортировки записей по дате события;
* АРМ должен поддерживать возможность просмотра событий журналирования как в текстовом, так и в табличном виде;
* пользовательские запросы на поиск событий журналирования должны аудироваться в подсистеме Аудит;
* АРМ должен поддерживать выгрузку журналов в текстовый файл с учетом указанных значений в фильтре поиска;
* факт выгрузки журналов должен аудироваться в подсистеме Аудит;
* выгрузка должна быть доступна только для аутентифицированных/авторизованных пользователей системы;
* АРМ должен поддерживать поиск по всему атрибутному составу трейсов. В фильтре поиска обязательно должно быть заполнено значение времени, за которое осуществляется поиск записей;
* трассировка должна быть доступна для отображения в АРМ в виде дерева вызовов и длительности их выполнения;
* пользовательские запросы на поиск трейсов должны аудироваться в подсистеме Аудит;
* инструментарий по просмотру трейсов связанных с событием журналирования:
* АРМ должен предоставлять возможность просматривать цепочки вызовов для связанных с ними событий журналирования по traceID;
* факт запроса на просмотр трейсов связанных с событием журналирования должны аудироваться в подсистеме Аудит;
* АРМ должен поддерживать возможность сохранения фильтров настроенных пользователем для дальнейшего их переиспользования;
* АРМ должен поддерживать возможность настройки TimeZone для пользователя;
* пользовательские запросы на изменение настроек должны аудироваться в подсистеме Аудит;
* осуществлять вход в АРМ журналирование могут только аутентифицированные и авторизованные пользователи системы.

### Компонент кросс платформенный мониторинг

Компонент мониторинга должен обеспечивать:

* возможность использовать любую реализацию API (в том числе OpenSource) для установки точек мониторинга, если метрики выставляются в формате Prometheus версии 2.0 по заданному URL в прикладном приложении;
* возможность собирать метрики в разрезе тенантов в формате Prometheus версии 2.0 по заданному URL в прикладном приложении, запущенному системе контейнеризации;
* возможность интеграции с Prometheus системы контейнеризации и получении от него инфраструктурных метрик контейнеров, принадлежащие определенному проекту;
* возможность визуализации полученных метрик в виде таблиц, графиков в интерфейсе администратора системы мониторинга;
* возможность сбора метрик с узлов и виртуальных машин и его отключение;
* возможность создавать подключение клиента, указывая необходимые параметры и объекты хранилища;
* возможность маршрутизации метрик в разные топики в зависимости от данных в метрике;
* возможность использовать API для получения метрик и просмотра данных, отключения от мониторинга.

Подсистема хранения метрик должна обеспечивать:

* хранение метрик с возможностью изменения сроков хранения данных;
* управление объектами внутри хранилищ;
* обеспечение единой ролевой модели для всех хранилищ;
* предобработка данных перед загрузкой в хранилища;
* выдача данных по запросу.

### Компонент управление настройками приложений

Компонент должен обеспечивать хранение, управление и предоставление по запросу параметров конфигурации библиотек, сервисов Платформы и прикладных приложений, разработанных на Платформе, а именно:

* хранение всех настроек в едином хранилище;
* возможность, используя АРМ администратора:
  + просмотреть список параметров и конфигураций (совокупный набор параметров, определенных областью сервиса или приложения потребителя);
  + найти параметр или конфигурацию по следующим критериям: идентификатор параметра или конфигурации, идентификатор потребителя, значение, тип параметра;
  + выполнить загрузку модели конфигурации из файла формата .xml;
  + выполнить загрузку настроек конфигурации из файла \*.properties;
  + поддерживать создание и хранение следующих типов параметров: Строка, Число, Дата, Логический;
  + создать новый параметр;
  + просмотреть значения параметра;
  + редактировать значения параметров;
  + удалить значение параметра;
  + просмотреть историю изменения значения параметра;
  + удалить параметр;
  + поддерживать несколько версий конфигурации;
  + экспортировать список параметров в файл в формате json;
  + экспортировать настройки конфигурации в файл \*.properties и \*.yml;
  + переопределить значения параметров и конфигураций;
  + сравнить параметры между разными хранилищами настроек;
  + реплицировать изменения параметров между хранилищами настроек;
  + создать запрос на подтверждение второй рукой изменений значения параметров конфигурации;
  + подтвердить второй рукой изменения значения параметров конфигурации;
  + отклонить второй рукой изменения значения параметров конфигурации;
  + удалить конфигурацию;
  + применить изменения значений параметров и конфигурации в runtime-режиме;
  + восстановить значение параметра;
* возможность загрузить параметры и конфигурации в хранилище настроек без использования UI:
  + через REST API;
  + через CLI;
  + для загрузки использовать формат файла json, xml;
  + возможность, получить значения параметров и конфигураций:
  + через JAVA API получения параметров;
  + через JAVA API получения конфигураций;
  + через REST API получения параметров;
  + через REST API получения конфигураций;
* журналирование и аудит действий по изменению настроек. События аудита содержат следующую информацию:
  + дату изменения;
  + время изменения;
  + автор изменений;
  + старое значение;
  + новое значение.

### Компонент управление аутентификацией и авторизацией

Компонент аутентификации должен обеспечивать идентификацию, аутентификацию пользователей при доступе к приложениям, в том числе:

* обработку запросов на аутентификацию от пользователей, при необходимости принудительный запрос аутентификационных данных:
  + реализация жизненного цикла (создание, изменение, блокировка) учётных записей пользователей;
  + аутентификация пользователей по паролю;
  + аутентификация пользователей по Open ID Connect;
  + аутентификация пользователей на прокси-сервере компонента;
  + поддержка HTTP(s);
  + передачу идентификатора пользователя для нужд идентификации в подсистемах:
  + передача информации о пользователе в формате JWT;
* формирование событий аудита в подсистемы аудита для их последующей обработки:
  + события аудита передаются в подсистему аудита Платформы;
  + передаются все действия по администрированию учётных записей пользователей;
  + передаются изменения настроек компонента;
  + передаются события аутентификации, как успешные, так и неуспешные;
  + создание аутентификационной сессии;
* ограничение доступа неаутентифицированного пользователя к подсистемам:
  + ограничение доступа на уровне IAM прокси, за исключением контекстов с разрешённым анонимным доступом;
  + каждая система разрешает доступ к своим ресурсам только при наличии валидного токена компонента.

Компонент Платформы должен обеспечивать авторизацию доступа пользователей к функционалу приложений на основе проверки ролей и прав доступа, в том числе:

* загрузку ролевой модели приложения, используя АРМ администратора;
* получение списка прав пользователя;
* проверку наличия прав у пользователя;
* экспорт в файл модели авторизации;
* синхронизацию модели авторизации;
* редактирование модели авторизации;
* получение групп пользователей по коду категории пользователя.

### Компонент регистрация и долговременное хранение событий Аудита

Компонент Аудит должен обеспечивать:

* сбора и хранения событий безопасности, возникающих в процессе функционирования компонентов Платформы и прикладных приложений, написанных на её базе;
* предоставления пользовательского интерфейса для просмотра агрегированных событий безопасности уполномоченным привилегированным пользователям Платформы;
* предоставления возможности подготовки отчетов на основании информации обо всех собранных событиях безопасности, возникающих в процессе функционирования Платформы.

Компонент должен осуществлять запись, хранение и предоставление возможности для мониторинга следующих событий безопасности для компонентов Платформы и прикладных приложений, написанных на её базе:

* вход (выход) конечного пользователя в Платформу посредством аутентификации и авторизации Платформы;
* попытки доступа конечных пользователей Платформы и пользователей прикладных приложений, создаваемых на ее базе, к защищаемым объектам доступа Платформы;
* создание, изменение, редактирование конечных пользователей Платформы и пользователей модулей, создаваемых не ее базе, привилегированными пользователями Платформы;
* изменение привилегий доступа конечных пользователей Платформы и пользователей прикладных приложений, создаваемых не ее базе.

Компонент должен реализовывать следующие меры защиты согласно требованиями документа «Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах (введен в действие приказом ФСТЭК России № 17 от 11 февраля 2013 г):

* сбор, запись и хранение информации о событиях безопасности в течение установленного времени хранения (РСБ.3, усиление 1 в части генерации (сбора, записи) записей регистрации (аудита) для событий безопасности, подлежащих регистрации (аудиту) в соответствии с РСБ.1 с составом и содержанием информации, определенными в соответствии с РСБ.2);
* генерирование временных меток и (или) синхронизация системного времени в информационной системе (РСБ.6 с усилением 1, в части осуществления получения временных меток, включающих дату и время, используемых при генерации записей регистрации (аудита) событий безопасности в информационной системе посредством применения внутренних системных часов операционной системы среды функционирования);
* защита информации о событиях безопасности (РСБ.7, в части предоставления доступа к записям журналов безопасности только администратору информационной безопасности).

Реализация механизмов безопасности компонента должна быть обеспечена архитектурой компонента.

## Компоненты управления разработкой и контейнеризации

### Управление контейнерами

Компонент должен обеспечивать автоматизацию управления жизненным циклом контейнеров и сервисов. В жизненный цикл контейнера входит планирование ресурсов, управление масштабируемостью, правила балансировки нагрузки и контроль доступности, а также организация виртуальных сетей.

Компонент должен содержать:

* наличие инструментов для создания, развертывания, запуска и управления приложениями в контейнерной среде;
* поддержка приложений внутри контейнеров, написанных на языках программирования JavaScript, Node.js, Python, Java, Go, C#;
* поддержка процессов автоматического развертывания контейнеров с собранными внутри приложениями (как платформенными, так и прикладными);

### Компонент управления разработкой включая инструменты CI/CD

\*\* Работа всех участников

\*\* Языки на которых можно вести разработку

* Планирование и производственный процесс
* Управление требованиями
* Управление релизами
* Анализ качества кода
* аналитики и мониторинга производственного процесса
* контроль исходного кода и конфигураций
* управление развертыванием
* репозиторий дистрибутивов (артифактов)
* управление сборкой ПО
* Управление тестированием

### Компонент моделирования и выполнения Бизнес процессов

Компонент Платформы должен обеспечивать инструментарий для проектирования и исполнения бизнес-процессов, в том числе с поддержкой пользовательских задач. Бизнес-процессы должны формулироваться в нотации BPMN 2.0 и поддерживать вызовы сервисов в формате: REST и JSON-RPC API.

Компонент должен обеспечивать:

* визуальный редактор проектирования бизнес-процессов в нотации BPMN 2.0.
* исполнение экземпляров бизнес-процессов, спроектированных в подсистеме, с возможностью:
  + запуск на выполнение экземпляров бизнес-процессов;
  + обеспечения постоянства контекста экземпляра бизнес-процесса, в случаях перехода экземпляра бизнес-процесса в ожидание;
  + организации вызовов внешних сервисов, указанных в модели бизнес-процесса (API: REST, JSON-RPC);
  + работа с пользовательскими задачами: выполнить, назначить, взять в работу, отменить, изменить приоритет.
* управление исполнением экземпляров бизнес-процессов посредством административной панели управления: прерывание процесса, повтор сбойного шага процесса;
* управление исполнением через GUI;
* версионирование моделей бизнес-процессов.

### Компонент формирования печатных форм

Компонент должен обеспечивать генерацию отчётных и печатных форм документов по размеченным шаблонам:

* формирование отчетов на основе данных, получаемых из прикладной бизнес-области;
* ведение реестра шаблонов с поддержкой версионности;
* управление шаблонами печати (импорт, активация\деактивация шаблонов и их версий);
* генерацию документов по шаблонам в выходных форматах PDF, DOCX, RTF, HTML, XLS, CSV.

### Компонент исполнения фоновых заданий

Компонент Платформы должен обеспечивать функциональность асинхронного запуска пакетных вычислительных задач в порядке очереди, в том числе:

* возможность, используя API по протоколу JSON-RPC 2.0:
* создавать, обновлять, удалять очереди и создавать задачи в очередях;
* изменять параметр количества одновременно запускаемых задач в очереди;
* возможность автоматически асинхронно запускать задачи в порядке очереди;
* обеспечить возможность запуска задач в виде вызовов API по протоколам REST и JSON-RPC 2.0.

Компонент Платформы должен обеспечивать функциональность асинхронного запуска пакетных вычислительных задач по расписанию, в том числе:

* возможность, используя API по протоколу JSON-RPC 2.0:
* создавать, получать, обновлять, останавливать и удалять задания;
* принудительно запускать задания;
* возможность автоматически асинхронно запускать задания по расписанию.

## Компоненты управления инфраструктурой

### Требования к подсистеме виртуализации вычислительных ресурсов

### Требования к подсистеме управления виртуализированными ресурсами

### Требования к подсистеме управления конфигурациями и развертыванием

### Требования к подсистеме вычислительной сети

# Требования к оборудованию и системному программному обеспечению

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*]

# Требования по информационной безопасности

Подсистема ИБ Платформы обеспечивается коммерческим программным обеспечением третьих лиц и программно-аппаратными комплексами защиты инфраструктуры.

Обеспечивается защита инфраструктуры на следующих уровнях:

* на физическом уровне;
* на сетевом уровне;
* на инфраструктурном уровне;
* на уровне среды виртуализации.

На сетевом уровне обеспечена защита внутренних сетей ЦОД инфраструктуры Платформы с использованием межсетевых экранов нового поколения, обеспечивающих:

* защиту сетевого периметра инфраструктуры;
* сегментирование внутренней сети инфраструктуры;
* межсетевое экранирование сетевых потоков;
* выявление и предотвращение компьютерных атак;
* базовую защиту внешних каналов связи инфраструктуры от DDoS- атак;
* возможность доступа к подсистеме управления инфраструктурой только с использованием криптографически защищенного канала связи.

На инфраструктурном уровне обеспечено:

* управление доступом к подсистеме управления инфраструктурой с использованием средств двухфакторной аутентификации подключающихся к ней администраторов инфраструктуры и администраторов ВЦОД;
* возможность доступа к подсистеме управления инфраструктурой только администраторов инфраструктуры и администраторов ВЦОД;
* защита от несанкционированного доступа к инфраструктуре;
* доверенная загрузка серверов инфраструктуры с использованием аппаратно-программных модулей доверенной загрузки;
* контроль администраторов инфраструктуры и администраторов ВЦОД с использованием специализированных средств контроля действий привилегированных пользователей;
* антивирусная защита инфраструктуры;
* сбор и анализ событий информационной безопасности;
* контроль и анализ защищенности инфраструктуры с использованием специализированных средств по выявлению уязвимостей в используемом ПО и его некорректной конфигурации, влияющей на уровень защищенности ПО, с устранением выявленных уязвимостей и/или недостатков;
* изоляция ВЦОД разных пользователей и групп пользователей друг от друга.

Система защиты информации инфраструктуры Платформы исключает возможность доступа непривилегированных пользователей к подсистеме управления.

На уровне среды виртуализации обеспечена защита консоли управления и самообслуживания инфраструктуры с использованием межсетевых экранов уровня приложений.

Предусмотрены организационно-технические меры в составе подсистемы защиты информации инфраструктуры Платформы, обеспечивающие:

* управление доступом к инфраструктуре;
* администрирование и мониторинг подсистемой защиты информации и используемыми в её составе средствами защиты информации инфраструктуры;
* выявление и реагирование на события (инциденты) информационной безопасности;
* регулярные сканирования инфраструктуры на наличие актуальных уязвимостей с устранением выявленных уязвимостей.

# Установка Платформы, обучение персонала

* 1. Исполнитель должен иметь успешно реализованные проекты по созданию, внедрению, установке аналогичных систем.
  2. Деятельность Исполнителя, связанная с ознакомлением с необходимой информацией, сбором данных, разработкой документов, вводом программно-технических средств в эксплуатацию, а также их сопровождением, выполняемая на объектах Общества, должна осуществляться в соответствии с согласованными с Обществом порядком предоставления доступа специалистам Исполнителя на объекты Общества.
  3. Привлечение Исполнителем соисполнителей согласовывается с Обществом в рамках технико-коммерческого предложения и указывается в договоре на выполнения работ.
  4. Работы по установке и настройке Платформы должны осуществляться Исполнителем под контролем работников Общества.
  5. Исполнитель обязан в течение 1 месяца после заключения контракта представить на согласование Заказчику программу приемо-сдаточных испытаний Платформы. В программу приемо-сдаточных испытаний должны быть включены тесты, имитирующие выход из строя серверов, дисковых подсистем, коммуникационного оборудования, ЦОД целиком и сбои в системном программном обеспечении, тесты работоспособности системы резервного копирования и восстановления, системы мониторинга. Все тесты, имитирующие аварийные ситуации, должны отслеживаться системой мониторинга.
  6. Предоставить требования к квалификации персонала Заказчика для выполнения полного спектра работ по развертыванию, эксплуатации, разработке приложений на стороне Заказчика. Предусмотреть обучение 200 специалистов Заказчика согласно данным требованиям. Исполнитель должен предложить рекомендуемый перечень курсов. Окончательный состав курсов и количество слушателей, расписание курсов будет определено в рамках подготовки тендера. Окончательное расписание курсов должно быть согласовано не позднее 1 месяца после заключения контракта. Обучение должно производиться в учебных центрах производителя программного обеспечения, удаленно или на территории Заказчика.

# Требования к гарантийному обслуживанию и технической поддержке

* 1. Техническая поддержка системы и ее гарантийное обслуживание должно осуществляться в течении 3-х лет с момента ввода в эксплуатацию.
  2. Инцидентная поддержка на все компоненты платформы PROD-стенда должна осуществляться 7 дней в неделю 24 часа в сутки. Для остальных стендов и компонент ландшафта поддержка осуществляется в рабочие дни 5 дней в неделю с учетом праздников.
  3. Отношение времени работоспособности Системы к общему времени (за календарный год) должно составлять не ниже 0,994 без учёта времени, затраченного на проведение плановых профилактических работ на Системе. Периодичность плановых профилактических работ не должна быть чаще одного раза в месяц и не должна превышать 12 часов. Время проведения плановых профилактических работ определяется Заказчиком.
  4. Регламент гарантийных взаимоотношений с Исполнителем работ должен быть подготовлен и оформлен приложением к договору.
  5. Условия SLA оформляются приложением к договору.
  6. Условия SLA и гарантии на компоненты Платформы единые независимо от того компонент это производства поставщика, СПО компонент, или проприетарный компонент, предоставляемый на условиях сублецинзионного соглашения;
  7. Период выхода обновлений программного обеспечения определяется поставщиком Платформы, но не реже выхода минорных версий СПО на основе которых разработан компонент Платформы.
  8. Каналы и способ предоставления обновлений определяются приложением к договору.
  9. В рамках технической поддержки предоставляется:
* плановые обновления программного обеспечения;
* обновления документации;
* консультирование по компонентам Платформы.
  1. В рамках инцидентной поддержки предоставляются:
* услуги по анализу и исправлению сбоев, отказов и иного нештатного функционирования компонент Платформы;
* выдача рекомендаций по настройкам и использованию компонент Платформы;
* внеплановое (экстренное) изменения ПО Платформы в т ом числе патчи безопасности;

# Расходы на эксплуатацию Платформы

Включить в коммерческое предложение расходы до 31.12.2027 с разбивкой по годам по следующим статьям:

* расходы на поддержку (сопровождение) Платформы;
* расходы на оплату лицензий программного обеспечения.

# Проект реализации предложенного решения

Исполнитель обязан в срок не позднее 1 месяца до начала работ представить и согласовать с Заказчиком проектную документацию по предлагаемому решению. Проектная документация должна содержать:

* пояснительную записку к проекту;
* руководство по развертыванию и общее описание Платформы;
* план аварийного восстановления компонентов Платформы после катастрофы с детальным описанием всех действий по устранению последствий аварии и восстановлению данных;
* график выполнения работ.

Лист согласования

РАЗРАБОТАНО

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Инициалы, фамилия | Подпись | Дата |
| Заместитель начальника технического управления ПАО «Сургутнефтегаз» – начальник отдела внедрения средств вычислительной техники, автоматизированных систем управления и телекоммуникаций | А.В.Полунин |  |  |
| Начальник отдела мониторинга и безопасности информационных систем управления информационных технологий ПАО «Сургутнефтегаз» | А.М.Шевчук |  |  |
| Заместитель начальника управления информационной безопасности ПАО «Сургутнефтегаз» | А.Н.Шубин |  |  |
| Главный инженер  ПУ «СургутАСУнефть» | Е.А.Шарф |  |  |
| Заместитель главного инженера ПУ «СургутАСУнефть» | К.А.Ермаков |  |  |
| Начальник комплекса обеспечения автоматизации вычислительных работ  ПУ «СургутАСУнефть» | В.В.Шумов |  |  |
| Заместитель начальника комплекса обеспечения автоматизации вычислительных работ  ПУ «СургутАСУнефть» | Г.В.Киселев |  |  |

СОГЛАСОВАНО

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Инициалы, фамилия | Подпись | Дата |
| Заместитель главного инженера – начальник технического управления  ПАО «Сургутнефтегаз» | Д.Н.Олейник |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Начальник управления информационных технологий  ПАО «Сургутнефтегаз» | Р.Д.Гимранов |  |  |
| Начальник управления информационной безопасности ПАО «Сургутнефтегаз» | А.Е.Григоров |  |  |
| Начальник  ПУ «СургутАСУнефть» | А.С.Шумцов |  |  |