Требование к интеграционной платформе ИТ КГК

# Содержание

[1. Общие требования 4](#_Toc95991618)

[2. Требование к архитектуре 4](#_Toc95991619)

[2.2. Функциональная архитектура 5](#_Toc95991620)

[2.3. Программно-компонентная архитектура интеграционной платформы 6](#_Toc95991621)

[2.4. Задачи выполняемые бэкенд разработкой и администрированием 7](#_Toc95991622)

[2.5. Компоненты интеграции 7](#_Toc95991623)

[2.6. Требования к интеграционным сервисам 8](#_Toc95991624)

[1.7. Безопасность оркестратора 9](#_Toc95991625)

[1.8. Безопасность образов 10](#_Toc95991626)

[1.9. Безопасность контейнеров 10](#_Toc95991627)

[1.10. Требования к протоколам обмена данными при интеграции информационных систем 10](#_Toc95991628)

[1.11. Операционная система – платформа 10](#_Toc95991629)

[1.12. Расположение данных 11](#_Toc95991630)

[2. ETL процесс 11](#_Toc95991631)

[3. Требование по классам критичности и отказоустойчивости 12](#_Toc95991632)

[3.1. Классификатор IT-Систем по приоритету восстановления 12](#_Toc95991633)

[3.2. Классификатор IT-Систем по типу обработки отказов 13](#_Toc95991634)

[3.3. Классификатор IT-Систем по времени восстановления и доступности за год 14](#_Toc95991635)

[3.5. Типовой архитектурный шаблон “High Speed+ (RC2)” для интеграционной платформы 17](#_Toc95991636)

[4. Требование по своду правил ИТ КГК 20](#_Toc95991637)

[4.1. Общие требования ИТ КГК 20](#_Toc95991638)

[4.2. Принципы Архитектуры данных 22](#_Toc95991639)

[4.3.1. Принципы Архитектуры прикладного программного обеспечения (информационных систем) 24](#_Toc95991640)

[4.4. Принципы Технологической Архитектуры 26](#_Toc95991641)

[5. Требование по информационной безопасности 27](#_Toc95991642)

[5.1. Общие требования к интеграционной платформе 27](#_Toc95991643)

[5.2. Требование к аутентификации и авторизации 27](#_Toc95991644)

[5.3. Требования к логированию и мониторингу 29](#_Toc95991645)

[5.4. Требование к эксплуатации приложения 30](#_Toc95991646)

[5.5. Требования к обработке и хранению данных 31](#_Toc95991647)

[5.6. Требования к администрированию 32](#_Toc95991648)

[6. Список задач для интеграции 33](#_Toc95991649)

[6.1. Список задач по интеграции для PoC (Proof of Concept) 33](#_Toc95991650)

[6.2. Список задач по интеграции после завершения проекта. 34](#_Toc95991651)

[7. Требование к обучению 38](#_Toc95991652)

# 1. Общие требования

* Интеграционная платформа должна поставляться вместе с описанием архитектуры (логическая структура приложений с модулями и их функциями; список инструментов и их версии; способы интеграции; модель данных; рекомендации по сайзингу на аппаратное обеспечение в разрез сред – разработка, тестирование, препрод, прод);
* Интеграционная платформа должна позволять масштабироваться (вертикально и горизонтально). Конкурентным преимуществом будет считаться наличие горизонтального масштабирования;
* Интеграционная платформа должна иметь встроенные инструменты мониторинга работоспособности системы (API для отслеживания состояния системы, оценка работоспособности и качества бизнес-процессов, средства конфигурирования и оперативного управления);
* Интеграционная платформа может быть предоставлена в виде «готового решения» с использованием модели лицензирования (именные, конкурентные пользователи, процессорная лицензия, лицензия на ядра серверов, лицензия на платформу, revenue sharing), а также предоставлена в виде «полуготового решения» с использованием подхода «заказной разработки» при наличии собранного и функционирующего DevOps pipeline и подтвержденного опыта успешной разработки и сборки всех компонентов, указанных в п. 1.2. Функциональная архитектура и п. 1.3. Программно-компонентная архитектура у других Заказчиков (не менее трех Заказчиков).
* При предоставлении интеграционной платформы в виде «полуготового решения» с использованием подхода «заказной разработки» исключительные авторские права на интеграционную платформу должны принадлежать КГК.

# 2. Требование к архитектуре

Архитектура интеграционной платформы ИТ КГК должна быть выстроена таким образом, чтобы учитывались принципы и требования свода правил и классов стандартов ИТ КГК, а также включала все возможности для расширения и развития в перспективе.

Основные сервисы и принципы интеграционной платформы:

* Централизованный ETL процесс
* Единая и современная визуализация данных
* Стандартизованный подход к описанию модели данных
* Легкоконфигурируемые экранные формы для пользователей
* Встроенная система управления бизнес-процессами и бизнес-правилами
* Централизованная система нормативно-справочной информации (НСИ)
* Каталог данных
* Информационная безопасность и политика доступа к данным
* Интегрированные механизмы контроля качества данных
* DevOps pipeline (Continuous Integration – CI; Continuous Delivery - CD; Continuous Deployment – CD; Continuous Feedback – CF).

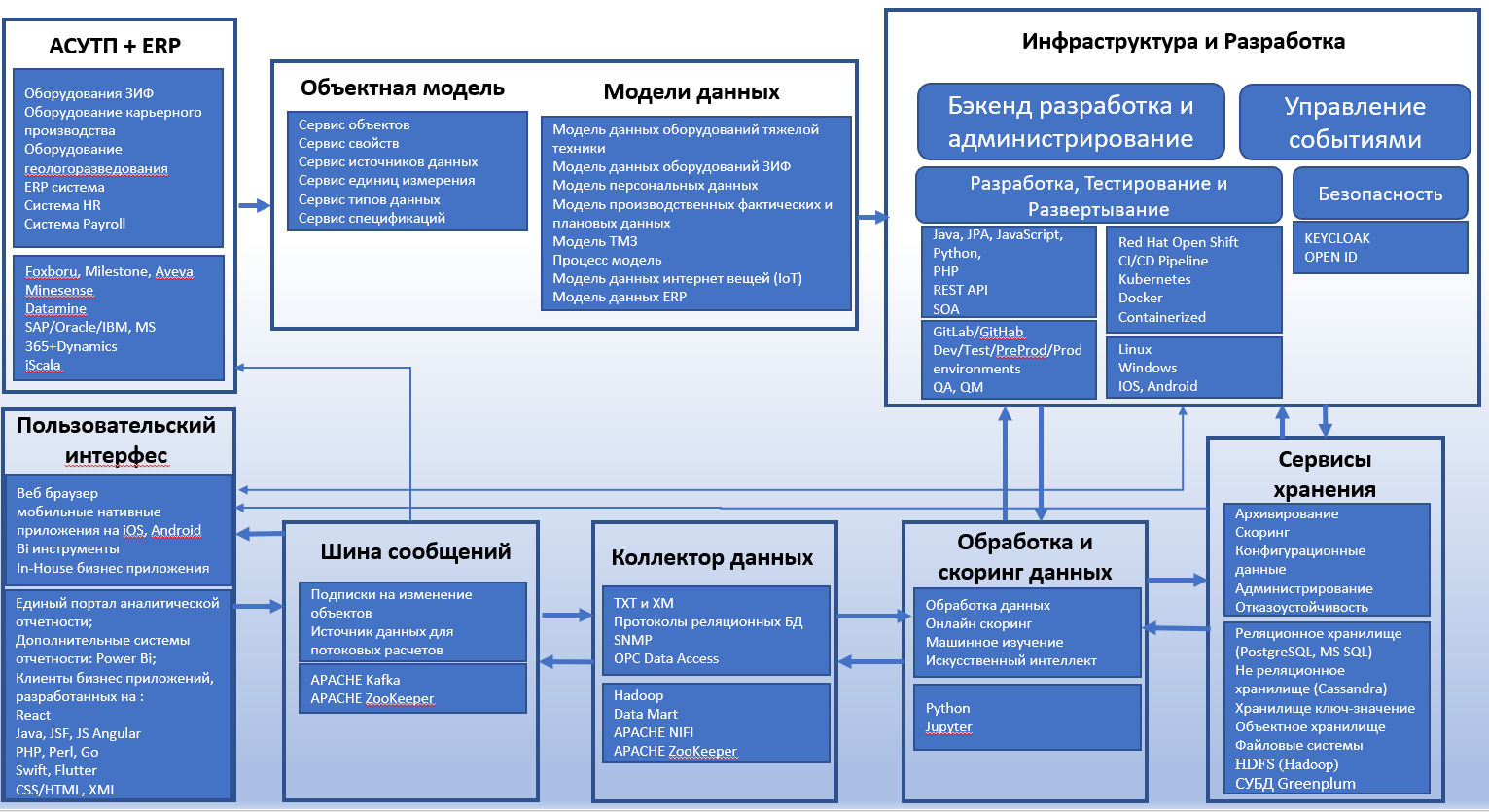
## 2.2. Функциональная архитектура

Функциональная архитектура интеграционной платформы представляет механизм управления сервисами и администрированием функциональной части процесса интеграции.



## 2.3. Программно-компонентная архитектура интеграционной платформы

Программно-компонентная архитектура описывает компонентные составляющие программные и сервисные реализации интеграционной архитектуры ИТ КГК.



**Примечание:** Поставщик может предложить альтернативы по программно-компонентной архитектуре платформы.

## 2.4. Задачи, выполняемые бэкенд разработкой и администрированием

* Исполнение типовых задач эксплуатации приложений, включая:
  + Разработка
  + Тестирование
  + Развертывание
  + Конфигурирование
  + Мониторинг
  + Журналирование
  + Трассировка
* Среду интеграций приложений в парадигме поставщик/потребитель API по моделям запрос/ответ и публикация/подписка, а также инструменты интеграций этой среды со внешним миром.
* Возможности визуального проектирования, исполнения и мониторинга оркестрируемых технологических и бизнес-процессов.
* Эффективное достижение требуемых нефункциональных характеристик работы интегрирований системы приложений, включая:
  + Производительность
  + Надежность
  + Доступность
  + Безопасность

Использование компонентов исключает необходимость реализации и поддержки нефункциональных требований по защите информации на уровне приложений.

## 2.5. Компоненты интеграции

Компоненты интеграции предназначены для обеспечения реализации интеграции различных ИТ систем КГК, предоставляя готовые необходимые механизмы и стандарты при разработке новых интеграций, которые должны ускорить и упростить процесс разработки интеграции приложений. В то же время уровень надежности и производительности интеграции должна быть высокой для всех приложений интеграционной платформы.

Интеграционные компоненты должны выполнять следующие функции:

* Разработка сбора и обработки данных
* Разработка и администрирование хранилищ данных
* Моделирование данных
* Развертывание разработки (по DevOps)
* Администрирование сервис аккаунтов и их доступов
* Управление событиями
* Мониторинг качества.
* Администрирование безопасности
* Логирование данных и событий
* Управление пользовательскими сессиями
* Разработка и подключение пользовательского интерфейса
* сервис подключения к внешним системам с использованием различных транспортных протоколов;
* разработку и поддержку на различных форматах сообщений;
* разработку и поддержку различных режимов взаимодействия (синхронные и асинхронные);
* преобразование структуры сообщений, с возможностью настройки и перекодировки связки атрибутов;
* поддержку аутентификации и авторизации запросов между приложениями;

Кроме того, для управления данными могут использоваться следующие компоненты:

* Скоринг и обработка данных
* Архивирование данных
* Генерация уникальных идентификационных кодов
* Применение справочных данных

Также для управления интерфейсом взаимодействия необходимы следующие компоненты:

* HTTP клиент
* REST/SOAP запросы и ответы
* Сервис управления надежности
* Настройка прокси для запросов
* Шина передачи данных
* Обработка событий
* Файловая передача данных

## 2.6. Требования к интеграционным сервисам

Интеграционная автоматизированная система должна выполнять следующие функции:

* Маршрутизацию взаимодействия интегрированных систем, в том числе применение справочников
* Аутентификацию и авторизацию участников интеграции (поставщиков и потребителей)
* Мониторинг работы сервисов
* Аудит

Интеграционная платформа может выполнять следующие функции для интегрированных систем:

* Service Discovery
* Преобразование форматов, в том числе и применение справочников
* Преобразование протоколов
* Временное хранение сообщений (до момента получения адресатом) в персистирующем слое
* Фильтрация сообщений
* Агрегация сообщений
* Валидация сообщений
* Повторная отправка сообщений
* И другие интеграционные логики

Интеграционная платформа не должна:

* Реализовать бизнес-логику, не входящую в функции интеграционных систем
* Выступать владельцем данных, которые поступают от системы-поставщика.

При интеграции систем должен быть реализован следующий функционал (в зависимости от применяемой технологии):

* Таймауты на подключение
* Таймаут на ответ по запросу
* Время жизни запроса
* Запись события и нотификация при превышении согласованных пороговых значений по количеству подключений, входящих и исходящий запросов
* Запись события и нотификация при превышении таймаутов ответов и времени жизни запросов
* Запись события и нотификация при превышении согласованных значений длины очереди запросов на обработку

При интеграционных нагрузках системы, превышающих установленный максимальный уровень, необходимо обеспечить функционирование сервиса, параллельно ограничивая нагрузку сверх согласованных максимальных параметров, в том числе и в разрезе интеграционных потоков, а также направлять ответ по умолчанию (техническую квитанцию) о повышении порога производительности и, примененных в связи с этим, мерах.

## 1.7. Безопасность оркестратора

* + управление namespaces, включая распределение контейнеров по нодам в зависимости от их значимости для компании (например, все контейнеры, которые входят в скоуп PCI DSS, должны «подниматься» только на специально отведенном сервере);
  + управление pod security policy (политиками безопасности, которые определяют возможности и ограничения контейнеров);
  + управление ограничениями запускаемых контейнеров (управление квотами на использование ресурсов, применение функционала AppArmor, SELinux, cgroups);
  + обеспечение сетевой безопасности при взаимодействии контейнеров (как между собой, так и с внешними ресурсами).

## 1.8. Безопасность образов

Контейнеры создаются из образов (аналог template виртуальных машин), которые могут быть получены из внешних или внутренних реестров либо создаваться разработчиками в рамках CI/CD-pipeline. Обеспечить безопасность образов рекомендуется на всех этапах.

* + удаление «лишнего» (например, в образе контейнера в момент разработки требовался уязвимый curl, а впоследствии потребность в нем пропала, поэтому проще удалить ненужный уязвимый компонент, чем пытаться его обновить);
  + «подпись» образов, гарантирующая, что контейнер запускается только из образов, целостность которых контролируется;
  + встраивание проверок образов в CI/CD-pipeline для автоматизации сканирования на наличие уязвимостей и compliance-несоответствий (например, запуск контейнера из-под привилегированной учетной записи, незащищенные системные файловые каталоги, завышенные права системных пользователей, включенные ненужные системные службы и т. д.);
  + сканирование образов в частных и публичных реестрах, используемых компанией;
  + управление контролем доступа к реестрам.

## 1.9. Безопасность контейнеров

Необходимо контролировать то, что происходит с контейнерами в их режиме run-time:

* + настройка политик безопасности, в которых указывается, например, перечень разрешенных команд, системных вызовов, сетевых взаимодействий;
  + поведенческая аналитика: создание «модели» контейнера путем профилирования для последующей идентификации отклонений от «нормы» (например, изменение конфигурационных файлов, запись в сторонние хранилища, попытка установления несанкционированных сетевых соединений, запуск вредоносных файлов и т. д.).

## 1.10. Требования к протоколам обмена данными при интеграции информационных систем

Платформа должна поддерживать информационный обмен данными посредством API (например, использование микросервисов на Java SpringBoot).

Интеграционная технология должна поддерживать один из следующих протоколов передачи данных: JSON (RPC), gRPC, XML (RPC), HTTPS, batch import, SQL.

В качестве технологии и стандартов обмена данными должна быть поддержка технологий RESTful API и SOAP.

## 1.11. Операционная система – платформа

Платформа для клиентской части приложения должна быть предоставлена на веб браузере, мобильном устройстве, операционных системах windows и Linux. Клиентская часть приложения не должна быть зависимой от одной платформы, она должна быть кроссплатформенной. Следующие платформы должны поддерживаться в зависимости от задач:

* IOS
* Android
* Windows
* Web browsers (Chrome, MS Edge, Firefox)

Для чтения и обработки цифровых данных со счетчиков контролеров и датчиков оборудований, клиентом может выступать программная оболочка датчиков.

Серверная часть платформы также должна быть кроссплатформенной, т. е. поддерживать следующие операционные системы:

* Linux/UIX/AIX
  + Oracle (Salaries),
  + Centos,
* Windows,
* Mac – Допускается для решений приобретаемых закупками по проектам.
* IBM (UIX/OS5) – Допускается для решений приобретаемых закупками по проектам.

## 1.12. Расположение данных

Платформа должна быть On-Premise решением и должна быть развернута в Центрах Обработки данных (ЦОД), арендуемых или приобретенных КГК.

Основной центр обработки данных будет находиться в ЦОД Рудника КГК. Резервный ЦОД будет располагаться в Бишкеке, арендуемым КГК.

# 2. ETL процесс

ETL (Extract, Transfer and Load) – относится к процессу извлечения, преобразования и загрузки. Это своего рода этап интеграции данных, когда данные, поступающие из разных источников, извлекаются и отправляются в хранилища данных. Данные, извлеченные из различных ресурсов, должны преобразоваться в определенный формат в соответствии с требованиями бизнес-процесса для дальнейшей загрузки в хранилище данных.

ELT (Exttract, Load, Transfer) – извлечение, загрузка и далее преобразование. Данный подход также может использоваться в зависимости от специфики задачи. В этом случае данные сначала сохраняются в хранилище данных и потом проходят преобразование согласно требованиям к формату по отображению в пользовательском интерфейсе.



Для построения ETL процесса могут применяться различные методы и инструменты в зависимости от задач к отчетам и данным. Для интеграции и построения ETL на уровне микросервисов должны применяться следующие технологии и стандарты:

* Брокер сообщений **Apache Kafka**
* Контейнеризация: технология Containerized от решения Kubernetes, технология с открытым кодом - Red Hat OpenShift
* Поддержка DataOps и DevOps, что обеспечивает взаимодействия внутри команд и отделов по развертыванию изменений
* Технология и протоколы передачи – HTML, XML, JSON, SAOP, REST, API

**Примечание:** для компонентов могут рассматриваться альтернативные варианты/аналоги по согласованию с КГК.

# 3. Требование по классам критичности и отказоустойчивости

Целью применения стандартов классов IT-Систем является сокращение времени на проектирование информационных систем за счет применения системы классификаторов и типовых архитектур на ранних стадиях проектов, снижение затрат на эксплуатацию информационных систем за счет применения типовых архитектур. Настоящий стандарт описывает классы IT-Систем, а также технические, технологические и организационные требования, предъявляемые к каждому классу.

## 3.1. Классификатор IT-Систем по приоритету восстановления

|  |  |  |
| --- | --- | --- |
| **Код класса Системы** | **Имя класса Системы** | **Описание классификаторов** |

|  |  |  |
| --- | --- | --- |
| BC | Business Critical | Системы, недоступность которых оказывает немедленное негативное влияние на способность компании получать доходы или оказывает критическое влияние на эффективность работы большого количества сотрудников компании КГК. Иными словами, это системы, обеспечивающие поддержку бизнес-деятельности компании КГК. Недоступность этих Систем в течение 1 сутки приводит к значительным косвенным производственным потерям, в результате которого компания КГК понесет существенные экономические потери в бизнесе. |

## 3.2. Классификатор IT-Систем по типу обработки отказов

|  |  |  |
| --- | --- | --- |
| **Код классифи**  **катора** | **Имя классификатора** | **Описание** |
| DR | Disaster Recovery | Обеспечивает полное дублирование оборудования на удаленной площадке и ручное восстановление работоспособности IT-Системы в **резервном** ЦОД (ручное  переключение на резервное оборудование) |

\***полуавтоматическое восстановление –** относится только к **системам управления базами данных** (далее – СУБД), на которых в автоматическом режиме идёт постоянный накат архивных логов, но сам процесс переключения СУБД с основного на резервный осуществляется ответственным человеком.

## 

## 3.3. Классификатор IT-Систем по времени восстановления и доступности за год

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код шаблона отказоус тойчивос ти | Имя шаблона отказоус тойчивос ти | Описание шаблона | Код класса Системы использую щий данный шаблон | Регламентир ованный % доступности Системы за год (SLA) | Макс. допустимое время простоя Системы за год | \*Восстановление в случае локального сбоя  системы | | \*Восстановление в случае падения основного ЦОДа | | Количество необходимых комплектов оборудования для обеспечения заявленной отказоустойчивости |
| RTO | RPO | RTO | RPO |
| RC2 | High Speed+ | Системы, недоступность которых оказывает немедленное негативное влияние на способность компании получать доходы или оказывает критическое влияние на эффективность работы большого количества сотрудников компании КГК. Иными словами, это системы, обеспечивающие поддержку бизнес-деятельности компании КГК. Недоступность этих Систем в течение 1 сутки приводит к значительным косвенным производственным потерям, в результате которого компания КГК понесет существенные экономические потери в бизнесе. | BC | 99,7% | до 1д 2ч 17м | до 30 мин | 0-30  мин | до 60 мин | 0-30 мин | Два комплекта оборудования (серверов + СХД) в основном ЦОДе и один комплект в рЦОДе  (всего 3 комплекта оборудования) |

**Восстановление в случае локального сбоя системы** – данная классификация определяет скорость восстановления IT-Систем при следующих вариантах сбоя:

* 1. выход из строя IT-Системы при общей работоспособности ЦОД;
  2. неработоспособность ЦОД в целом.

**Восстановление в случае падения основного ЦОД** – данная классификация определяет время доступности IT-Систем в год при следующих вариантах сбоя:

1. выход из строя IT-Системы при общей работоспособности ЦОД;
2. неработоспособность ЦОД в целом;
3. выход из строя любого серверного или сетевого оборудования;
4. пропадания канала связи в основном ЦОД.

### 3.4. Дополнительные классификаторы

### 3.4.1. Классификатор IT-Систем по звенности

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| 4U | Распределенная | Все компоненты могут быть размещены на множестве серверов или сервис элементах как облачных/виртуальных, так и на физических устройствах |

### 3.4.2. Классификатор IT-Систем по избыточности

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| FR | Полностью резервированная | Все компоненты IT-Системы имеют полностью  симметричное резервное оборудование («симметричный резерв»). |

### 3.4.3. Классификатор IT-Систем по режиму поддержки

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| S12x7 | S12x7 | IT-Система, сопровождаемая IT в режиме 12 часов в сутки и 7  дней в неделю. |

## 

### 3.4.4. Классификатор IT-Систем по способу обработки отказов

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| AR | Автоматическое  восстановление | Восстановление IT-Системы происходит автоматически без  вмешательства инженерного состава. |

### 3.4.5. Классификатор IT-Систем по этапам (статусам) жизненного цикла платформы и архитектуры

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| NEW | Новая | Регистрация новой ИТ системы в каталоге ИТ систем КГК |

### 3.4.6. Классификатор IT-Систем по производителям оборудования и системного ПО

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| PILOT | Пилотная система | Вновь внедряемая IT-Система, системные компоненты которой не удовлетворяют текущим IT Стандартам компании, но являются перспективными и учитывают тенденции развития IT-индустрии на долгосрочную перспективу.  Данный класс отражает временное состояние системы. По истечении срока пилотирования (устанавливается на АК) возможен переход в следующие состояния FC или  NC. |

### 3.4.7. Классификатор IT-Систем по уровню мониторинга

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| FM | Доступна для пользователей | IT-Система является объектом комплексного автоматического мониторинга, способного с минимальной задержкой определить недоступность системы для конечного пользователя и предпринять корректирующие  воздействия. |
| CM | CUSTOM | IT-Система, для мониторинга которой применяются  нестандартные средства, требующие привлечение сторонних специалистов или собственных разработок. |
| FM+CM | Гибридная | IT-Система является объектом комплексного гибридного мониторинга на основе FM+CM кодов. |

Для систем классов CO, MC, BC обязательным уровнем мониторинга является FM, при этом допускаются комбинации FM+SLAM, FM+CM.

### 3.4.8. Классификатор IT-Систем по категориям пользователей

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| AB+ EC | Внешние и внутренние  пользователи | IT-Система, пользователями которой являются, как и  внешние так и внутренние пользователи. |

### 3.4.9. Классификатор IT-Систем по типу масштабирования

|  |  |  |
| --- | --- | --- |
| **Код** | **Имя** | **Описание** |
| GM | Горизонтальное масштабирование | Архитектура IT-Системы позволяет увеличивать ее производительность путем добавления параллельных  компонентов с возможностью балансировки нагрузки между компонентами. |

## 3.5. Типовой архитектурный шаблон “High Speed+ (RC2)” для интеграционной платформы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Код шаблона отказоус тойчивос ти** | **Имя шаблона отказоуст ойчивост и** | **Описание шаблона** | **Восстановление в случае локального сбоя системы** | | **Восстановление в случае падения основного**  **ЦОДа** | |
| **RTO** | **RPO** | **RTO** | **RPO** |
| RC2 | High Speed+ | Системы, недоступность которых оказывает немедленное негативное влияние на способность компании получать доходы или оказывает критическое влияние на эффективность работы большого количества сотрудников компании КГК. Иными словами, это системы, обеспечивающие поддержку бизнес-деятельности компании КГК. Недоступность этих Систем в течение 1 сутки приводит к значительным косвенным производственным потерям, в результате которого компания КГК понесет существенные экономические потери в бизнесе. | 30 мин | 0-30 мин | 60 мин | 0-30 мин |

Системы класса RC2 по приоритету восстановления — это Системы **MC (Mission Critical)**, а по типу обработки отказов – **DR (Disaster Recovery)**.

### 3.5.1. Технологическое решение для RC2 IT-Систем:

Для обеспечения целевого времени восстановления, традиционных процессов и процедур резервного копирования на ленты недостаточно, поэтому необходимо иметь актуальную копию данных приложения на резервном диске сразу же после потери данных. Это возможно только при репликации данных приложения в реальном времени на несколько дисковых копий.

Для защиты данных от потери и логического искажения будет применяться стратегия РКД и восстановления с лент и/или дисков. При этом необходимо будет еженедельно выполнять полное РКД на магнитную ленту и каждый день проводить инкрементное РКД (а не просто архивировать резервные копии журнала) на магнитную ленту.

Все резервное копирование на магнитную ленту должно быть перекрестным между территориями, чтобы магнитная лента сразу оказывалась в центре восстановления данных и не пропала в случае чрезвычайной ситуации.

Хотя для этого класса Систем допустим более долгий срок восстановления, при использовании традиционных методов восстановления с магнитных лент его было бы сложно соблюсти, учитывая объемы данных, которые необходимо восстановить после аварийной ситуации для приложений, имеющих >1 ТБ данных. Для таких приложений будет применяться та же стратегия защиты данных, что и для класса восстановления RC1.

Могут применять следующие технологии резервирования данных:

1. РКД по сети LAN или SAN;
2. РКД данных на дисковую память;
3. РКД на магнитные ленты.

Схема восстановления Систем класса RC2 (High Speed+)



**Основной ЦОД**

**Резервный ЦОД**

SD

SD

**HEWLETT -Cla s**

**PA CKA RD**

**HEWLETT PA CKA RD**

POWERED

POWERED

Система Active

Система Active

Система Standby

Ленточная библиотека

Ленточная библиотека

**PA CK AR D**

**PA CK AR D**

**H EWLE TT**

**P AC KA RD**

**H EWLE TT**

**P AC KA RD**

**H EWLE TT**

**P AC KA RD**

Реплика СХД

СХД №1

СХД №2

СХД №3

**HE WLET T**

**HE WLET T**

Для Cистем класса RC2 должны быть учтены следующие требования:

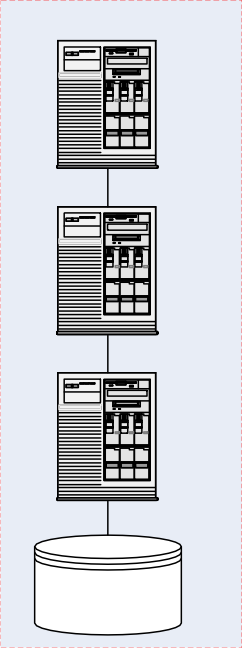
* 1. необходимо обеспечить мониторинг системы и ее компонентов (включая сетевое оборудование) в режиме 24/7;
  2. управление системой должно осуществляться только через автоматизированные средства контроля действий администраторов;
  3. необходимо в обязательном порядке настроить и использовать технологию

«кластеризации» (минимум - в разных помещениях, предпочтительно на разных ЦОД) или входить в состав приложения, обладающего внутренней устойчивостью. Если кластерные узлы находятся в одном ЦОД, они должны располагаться в разных стойках;

* 1. для данных Систем необходимо использовать подключенные через SAN СХД верхнего\среднего уровня (не внутренние диски и не DAS);
  2. одна копия данных приложения должна располагаться на дисковом массиве на другой площадке, т. е. настроена СХД реплика;
  3. для разных копий данных необходимо использовать разные СХД;
  4. необходимы решения PVG striping /Secure Path;
  5. необходимо использовать минимум 2 (два) выделенных сетевых адаптера для передачи данных с внешними сервисами;
  6. необходимо выделенное сетевое подключение, используемое для целей РКД для серверов приложений =>2Tb;
  7. необходимо полное, документированное решение по восстановлению после сбоев (Disaster Recovery);
  8. необходимо проводить тестирование восстановления данных согласно утверждённого и подписанного плана тестирования РКД;
  9. необходим действующий контракт(техподдержка) на обслуживание ПО и аппаратных средств со стороны Вендора ПО\оборудования (время реагирования 24 часа или меньше);
  10. не менее одного раза в неделю\день (в зависимости от приложения) следует производить полное РКД, а также не реже раз в сутки проводить инкрементальное РКД;
  11. РКД следует осуществлять в другой ЦОД (возможно, расщеплением зеркала в SAN или через локальную сеть на медиа-сервер);
  12. каналы связи между комплектами оборудования должны быть шифрованными.

Для Систем RC2 класса применяется следующая типичная схема резервирования аппаратного обеспечения:

**Disaster recovery**



**Резервный ЦОД**

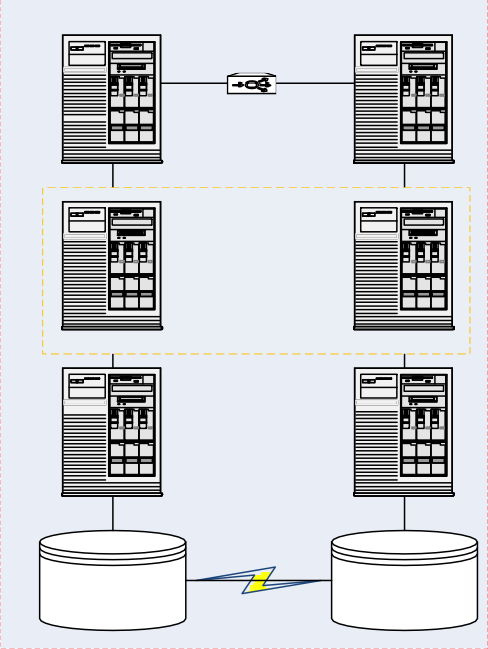
Web Server №3

Application Server №3

Standby Database

Server №3

СХД №3



**Основной ЦОД**

Web

Server №1

Web

Server №2

Web балансировщик

Application

Server №1

Кластеризация

Application

Server №2

Active DataBase

Server №1

Active DataBase

Server №2

СХД №1

Реплика СХД

СХД №2

Ручное переключение

### 3.5.2. Стоимость решения для RC2 IT-Систем:

Автоматическое переключение

Системное программное обеспечение (лицензии):

Предусмотрено специализированное ПО кластеризации для обеспечения повышенной отказоустойчивости.

Серверное оборудование:

При данном классе восстановление необходимо приобретать +2 комплект серверов. Сумму за оборудование необходимо умножить на K=3.

Дисковая подсистема:

При данном классе восстановление необходимо приобретать +2 объем необходимого дискового пространства.

Сумму за СХД необходимо умножить на K=3.

# 4. Требование по своду правил ИТ КГК

Целью свода правил ИТ является обозначение основных ценностей, требований и ограничений, применяемых при разработке, эксплуатации и развитии ИТ решений в бизнес-процессах компании КГК. А именно, это

- помощь в принятии решений подразделениями КГК при анализе / оценке идей, внедрении / изменении / эксплуатации ИТ систем/ обеспечении информационной безопасности, в соответствии с миссией, видением, бизнес и ИТ стратегиями КГК;

- использование стандартизированного подхода по внедрению / изменению / эксплуатации ИТ, с целью минимизации рисков ИТ, информационной безопасности и операционных рисков компании и получения максимальной выгоды от ИТ КГК

## 4.1. Общие требования ИТ КГК

### 4.1.1. Максимизация выгоды для КГК (многократное использование ИТ систем)

Необходимо выявлять и строить ИТ системы так, чтобы компоненты ИТ систем можно было использовать многократно, будь то бизнес-функция, сервис, приложение, информация, системное программное обеспечение или инфраструктура.

Архитектурный процесс должен быть построен таким образом, чтобы обеспечивать многократное использование уже существующих компонентов везде, где только возможно;

Необходимо искать любые возможности сокращения количества дублирований в бизнес-процессах, базах данных, операционных системах, в серверах приложений и в самих приложениях, чтобы максимально сократить и лицензии, и усилия по текущему управлению, и расходы на сопровождение.

### 4.1.2. Минимальная совокупная стоимость владения ИТ системами

Необходимо при построении Архитектуры КГК анализировать и выбирать ИТ системы и технологии на рынке, которые имеют минимальную совокупную стоимость владения ИТ системами и технологиями.

Использование в работе лучших и проверенных ИТ систем и технологий на рынке, гарантирующих удовлетворительное качество их работоспособности, при минимальной стоимости владения ИТ системами и технологиями.

### 4.1.3. Соответствие законодательству

При построении Архитектуры КГК необходимо руководствоваться законодательством Кыргызской Республики и внутренними нормативными документами КГК.

### 4.1.4. Защита интеллектуальной собственности

Приобретение и использование в работе официально разрешенного и лицензионного программного обеспечения (проприетартного и Open Source) и только от официальных уполномоченных Поставщиков и Вендоров ИТ систем.

Необходимо при анализе и выборе программного обеспечения выбирать и приобретать только официальное лицензионное программное обеспечение (проприетарное от Вендоров, либо официально разрешенное Open Source с GPL).

### 4.1.5. Проектирование адаптивности и гибкости

Необходимо активно отстаивать разработку и инвестиции в изменения, которые увеличивают адаптивность и гибкость ИТ архитектуры КГК.

Необходимо культивировать представление, что ИТ архитектура КГК должна быть автономной и слабо связанной. С течением времени, необходимо активно замещать компоненты, являющиеся не гибкими.

Увеличивать адаптивность через использование гибко настраиваемых узлов.

Необходимо контролировать проекты на предмет негативного влияния на адаптивность и гибкость поставляемых решений (например, если всегда принимать вариант архитектуры с минимально предложенной ценой).

### 4.1.6. Масштабируемость

ИТ архитектура КГК должна поддерживать высокий рост объемов операций, быть расширяемой и масштабируемой.

Необходимо, чтобы ИТ архитектура КГК обеспечивала поддержку требуемого уровня нагрузки в соответствии с долгосрочными прогнозами роста всех направлений бизнеса. Необходимо учитывать рост бизнеса при проектировании и разработке ИТ архитектуры КГК, чтобы не возникало также ограничений приложений, не связанных с производительностью, таких как завершение тех или иных внутренних счетчиков, адресов и пр.

### 4.1.7. Инновационность

При анализе и выборе решений важно учитывать, что инновации могут предоставить КГК уникальность и конкурентное преимущество.

Необходимо анализировать и учитывать потенциальные выгоды от новых технологий, которые могут предоставить новый уникальный пользовательский опыт (впечатления), оптимизировать бизнес - процессы и сократить совокупную стоимость владения. Самое главное предоставить конкурентное преимущество на рынке.

## 4.2. Принципы Архитектуры данных

### 4.2.1. Управление данными (информацией) как общим активом КГК

Необходимо организовать хранение и использование информации таким образом, чтобы наиболее эффективно получать прибыль, повышать конкурентные преимущества и ускорять принятия решений по всем направлениям бизнеса, в соответствии с требованиями информационной безопасности КГК.

Управление данными/информацией должно быть унифицировано.

Должны быть определены официальные источники информации.

Чтобы обеспечить доступность информации для принятия решений, необходимо разработать хранилища данных.

Необходимо ускорять скорость доступа к данным посредством STP.

Необходимо установить политики доступа к информации с учетом требований по конфиденциальности, целостности и доступности.

Различные направления бизнеса должны предоставлять доступ к имеющейся у них информации для использования всей организацией КГК.

### 4.2.2. Разделение транзакционных и аналитических данных КГК

С точки зрения обеспечения хранения, размещения и обработки данные следует разделять на транзакционные и аналитические.

Все объекты данных и базы данных должны быть классифицированы и соотнесены по признаку транзакционной или аналитической обработки данных.

Решения, в рамках которых транзакционная система становится хранилищем данных или местом расчета данных для принятия решений – неприемлемы.

Решения, в рамках которых в системе принятия решений начинают работать пользователи, меняющие исходные данные (т. е. выполняющие транзакции) – неприемлемы.

Необходимо обеспечить синхронизацию данных в транзакционной системе и в хранилищах в соответствии с требованиями бизнеса.

Системы принятия решений и аналитические системы должны получать данные из транзакционных систем через интерфейсы в таких режимах, которые не влияют на производительность транзакционных систем (например, журнальные репликации, запуск ETL в технологических паузах и т. д.).

Изменения данных напрямую в СУБД транзакционных и аналитических системах, минуя интерфейсы недопустимы и должны всячески запрещаться.

В задачах / проектах миграции данных и документов обязательно должны быть выполнены задачи по переносу исторических данных и документов со старых ИТ систем в новые ИТ системы.

### 4.2.3. Информация должна иметь владельца и мастер – систему

Для каждой информационной сущности должен быть определен владелец и мастерсистема, где эта сущность вводится и изменяется. Только полученное из этой мастерсистемы значение может считаться достоверным для всех бизнес-процессов КГК.

Все данные, имеющие значение для бизнеса, должны иметь владельца, определяющего и контролирующего политики ввода, изменений и жизненного цикла.

Для каждого объекта данных должна быть определена мастер-система. Только полученное из этой мастер-системы значение может считаться достоверным для всех бизнес-процессов КГК.

При необходимости изменения данных делается соответствующий запрос к мастер-системе, чтобы изменение производилось средствами и на стороне мастер-системы.

Запрещается реализация постоянных решений, допускающих возможность дублирования мастер-данных.

Реализация решений, допускающих возможность дублирования мастерданных, возможна только в качестве временного решения для быстрого внедрения бизнес-продукта. Для такого решения обязательно должен быть запланирован срок окончания временного использования и план постоянного решения.

При заказе разработки решения у внешнего поставщика требования по архитектуре данных фиксируются на этапе согласования архитектурного решения;

Нужен стандартный процесс согласования и контроля архитектурного решения.

Запрещается использование мастер-данных в целях тестирования

### 4.2.4. Корректность и актуальность данных.

Ценность представляет только корректная и актуальная информация. Необходимо управлять жизненным циклом хранимой информации, а также обеспечивать надлежащий контроль их корректности.

Для всех сущностей и атрибутов информации, актуальность которых может изменяться со временем, должны быть определены политика их применения и регламент актуализации.

Хранимые пользовательские данные должны иметь маркировку времени начального ввода и истории актуализации.

Должны быть организованы процессы проверки актуальности пользовательских данных и процессы обновления этих данных.

Данные вводятся в информационные системы один раз, и тут же выполняется проверка их корректности.

Должны быть реализованы механизмы проверки целостности данных.

По всем объектам и атрибутам данных, которые могут трактоваться и обрабатываться неоднозначно (например, персональные данные КГК), должны быть определены регламенты контроля качества ввода (проверка форматов, проверка по справочникам и т. д.) в момент сбора.

Необходимо следовать стандартам по взаимодействию, а также промышленным стандартам, во всех случаях, когда нет непреодолимых бизнес-причин внедрять нестандартное решение.

Необходимо установить процесс утверждения стандартов, их периодического пересмотра и корректировки, а также порядка согласования отклонений от стандартов.

Необходимо обеспечить, чтобы предоставляемая информация была одинаковой, вне зависимости от способа или канала доставки.

Для всех объектов транзакционных данных должны разрабатываться универсальные функции обработки, например, ввод нового объекта, изменение объекта, удаление объекта.

Для репликации и массовой передачи транзакционных данных между различными системами должны реализовываться универсальные сервисы таким образом, чтобы минимизировать общее количество физических потоков данных.

Универсальные сервисы должны быть опубликованы.

При реализации проектов необходимо максимально применять готовые универсальные сервисы обработки данных.

При реализации проектов необходимо выявлять новые универсальные функции обработки данных и реализовывать их как сервисы.

## 4.3.1. Принципы Архитектуры прикладного программного обеспечения (информационных систем)

### 4.3.1. Использование веб браузеров, «тонких» клиентов, мобильные приложения на iOS, Android

Необходимо избегать внедрения приложений с использованием «толстого клиента», отдавая предпочтение архитектурным решениям с «тонким клиентом», веб браузерам, мобильным приложениям на iOS, Android. Это касается в первую очередь front-end систем, а также других систем массового и/или распределенного использования. В тех случаях, когда использование «тонкого клиента» невозможно в силу ограничений конкретной прикладной системы (например: наследуемые устаревшие приложения, «коробочные» продукты, работающие в двухзвенной архитектуре, усиленные требования по информационной безопасности), должна быть обеспечена работоспособность данной системы при развертывании с использованием терминального решения.

### 4.3.2. Модульность прикладного программного обеспечения

Архитектура систем должна строиться из максимально независимых модулей, интегрированных между собой через универсальные интерфейсы (API) и сервисы, реализующий функционал и приемо / передачу данных.

Модули и компоненты ИТ систем должны строиться с учетом группировок сильно связанных бизнес-функций таким образом, чтобы обеспечить максимальную универсальность и многократное использование данного модуля их в автоматизации различных процессов.

Слабосвязанные функции не должны размещаться в одном физическом модуле или компоненте.

Интерфейсы (API) между модулями и компонентами должны строиться максимально универсальными как с точки зрения их функциональности, так и архитектуры их реализации.

Микросервисная архитектура в сочетании с SOA (Service Oriented Architecture) наиболее полно обеспечивает слабосвязанность информационной среды.

### 4.3.3. Принцип «Единое Окно»

Работа сотрудника по обслуживанию пользователей или при работе в бизнес-процессах должна быть максимально автоматизирована в рамках «единого окна»: интегрированного пользовательского интерфейса, обеспечивающего все необходимые сервисы и информацию без необходимости переключения между несколькими приложениями.

«Единое окно» не означает, что все приложения КГК должны быть реализованы на одной платформе. Речь идет о едином презентационном слое фронтальных приложений пользователей, обслуживающих внутренних или внешних клиентов КГК (или пользователей, задействованных в сквозных бизнес-процессах), интегрированном с другими системами на основе принципов SOA-интеграции.

При реализации «единого окна» необходимо избегать применения «лоскутной» сшивки пользовательского интерфейса разных систем (т. е. применять технологию встраиваемых фреймов), т.к. это, как правило, приводит существенному снижению функциональной и презентационной целостности интерфейса, усложняет администрирование доступа пользователей, а также может приводить к нестабильности работы интерфейса, которую сложно контролировать со стороны сопровождения.

Необходимо использовать единый механизм аутентификации и авторизации сотрудников в системах КГК, в т. ч. с использованием централизованного LDAPкаталога, унифицировать алгоритмы передачи контекста аутентификации/авторизации через веб-сервисы.

При предоставлении доступа к информационным системам из-за пределов периметра защиты осуществлять только с использованием методов двухфакторной аутентификации (использованием двух из трех факторов: «что я знаю», «что я имею», «что я есть сам»).

При проектировании решений необходимо соблюдать принципы архитектуры данных для упрощения создания систем в парадигме “Единого окна”.

Все middle и back-приложения должны публиковать веб-сервисы получения/изменения данных для обеспечения работы интегрированных решений “единого окна”.

### 4.3.4. Омниканальность

Необходимо предоставлять продукты и услуги КГК максимально удобным, единообразным для пользователей способом во всех возможных каналах взаимодействия (отделения, веб браузеры (интернет банкинг, сервисы на сайте), Call Centre, мобильные приложения, IVR, SMS, соц. сети и проч.) с единым универсальным механизмом доступа, при котором клиент вводит логин и пароль один раз в одной сессии взаимодействия с сервисом КГК и получает доступ ко всем доступным для него функциям удаленного обслуживания.

## 4.4. Принципы Технологической Архитектуры

### 4.4.1. Соответствие затрат бизнес – задаче

Необходимо обеспечивать соответствие затрат бизнес – задаче.

Необходимо проводить классификацию внедряемого прикладного программного обеспечения по уровню критичности для бизнеса – это позволит точнее обеспечить потребности заказчика, посредством использования типовых решений.

Совместное использование вычислительных мощностей и ресурсов аппаратного обеспечения в КГК. Это позволяет выделять ресурсы прикладному программному обеспечению в соответствии с их фактической необходимостью, а также более эффективно использовать аппаратные ресурсы. Также оплата за фактически используемые мощности позволит снизить совокупную стоимость владения ИТ системы.

### 4.4.2. Управляемость

ИТ система должна обеспечивать предоставление оперативной информации о своей работе и обеспечивать средства управления, позволяющие воздействовать на параметры работы системы.

### 4.4.3. Минимальное количество разных технологий

Необходимо при построении ИТ Архитектуры КГК анализировать и выбирать ограниченное количество технологий для удовлетворения потребностей КГК, исходя из возможностей по развитию и технической поддержке Блока ИТ.

### 4.4.4. Общие ресурсы и мощности ИТ инфраструктуры КГК

Необходимо при построении ИТ Архитектуры КГК анализировать и выбирать ИТ системы и технологий, которые поддерживают возможности и работу при совместном использовании мощностей (RAM, дисковое пространство и т. д.) для удовлетворения потребностей КГК.

### 4.4.5. Многократное использование аппаратного обеспечения

Архитектурные решения должны обеспечивать максимальную отдачу от применяемого оборудования в сочетании с обеспечением «пиковых» требований к аппаратным ресурсам и требуемого уровня отказоустойчивости.

Для оптимального использования аппаратных ресурсов необходимо использовать концепцию консолидации ИТ систем в сочетании с виртуализацией информационной инфраструктуры, которая может быть реализована следующими методами: • Частное облако для IaaS, PaaS, SaaS; • Публичное облако для PaaS; • Гибридное облако (Private Cloud интегрируется с внешними Public Cloud такими как Azure, Amazon, Google и т. д.).

# 5. Требование по информационной безопасности

## 5.1. Общие требования к интеграционной платформе

Для работы с интеграционными сервисами КГК аутентификация должна осуществляться в соответствии с требованиями спецификации WS-Security.

Информационная система, взаимодействующая с информационными системами КГК, должна соответствовать требованиям информационной безопасности согласно государственным стандартам Кыргызской Республики.

Все каналы связи, должны быть защищены с помощью сертифицированных средств криптографической защиты информации, соответствующих требованиям, установленным Стандартом КР по криптографической защиты информации.

Доступ третьих лиц ко всем техническим средствам, каналам связи и поддерживающим системам (электропитания, вентиляции, кондиционирования и т.п.) взаимодействия должен быть исключен.

## 5.2. Требование к аутентификации и авторизации

Работа сервисов (микросервисов) по интеграционным процессам должна выполняться под отдельно выделенным сервис аккаунтом – учетной записью, предназначенной только для интеграционного сервиса взаимодействующих между собой ИТ систем. Для каждой предметной области (один бизнес-процесс или совокупность бизнес-процессов) должен быть свой интеграционный сервис аккаунт, который будет иметь ограниченный доступ только для взаимодействующих систем на уровне интеграционного сервиса, т.е. кроме ограниченных интеграцией сервисов никуда он не должен иметь доступ и права на обработку данных должен быть в соответствии требованиям интеграции.

Интеграционный сервис аккаунт должен иметь права только на запуск объявленной в интеграции сервис (микросервис) процедуры.

Кроме этого, доступ сервис аккаунта к интеграционным сервисам ИТ систем участников взаимодействия должен осуществляться с использованием сертифицированных средств межсетевого экранирования.

Доступ пользователя или агента интеграционного сервиса на запуск интеграционной процедуры (сервиса) должен определяться также как и к другим функциям ИТ системы с помощью матрицы CRUD.

### 5.2.1. Управление доступом и секретами

На каждом из «уровней» — от ноды до контейнера — присутствует множество разных сущностей, взаимодействующих между собой. Для них необходимо выполнить следующее:

* разработать ролевую модель, описывающую, кто и с какими правами имеет доступ, например, к оркестратору, реестру с образами контейнеров, к конкретному namespace и т.д.;
* управлять секретами — паролями пользователей, сервисными учетными записями, токенами для аутентификации в сторонних сервисах, выстроить корректное управление инфраструктурой открытых ключей.

Diagram

Description automatically generated with low confidence

**Secret Management.**

* API-токены;
* SSH-ключи;
* технологические учетные записи;
* данные аутентификации облачных сервисов;
* 509-сертификаты и другие чувствительные данные (например, пароли).

Рисунок 4. Общий вид жизненного цикла процесса разработки ПО с ИБ-решениями

Graphical user interface, application

Description automatically generated

## 5.3. Требования к логированию и мониторингу

Программными средствами должны протоколироваться (логироваться) факты приема и отправки каждого информационного сообщения в рамках системы взаимодействия с указанием уникального в рамках сервиса идентификатора сообщения, направления сообщения, даты, времени, адресата и контрольной суммы сообщения.

В рамках процедуры мониторинга состояния и использования сервисов, зарегистрированных в интеграционной шине, для каждого взаимодействия автоматически должны регистрироваться следующие данные:

* запрашиваемый электронный сервис;
* пользователь (для авторизованных запросов);
* IP-адрес пользователя;
* время отклика сервиса;
* содержимое запроса;
* содержимое ответа;
* объем передаваемых данных в запросе (в байтах);
* объем передаваемых данных в ответе (в байтах);
* при возникновении ошибки ее описание.

В рамках процедуры мониторинга состояния и использования сервисов, зарегистрированных в интеграционной шине, должны выполнять следующие действия:

* в автоматическом режиме осуществляется регулярный опрос зарегистрированных сервисов, анализируется их состояние и формируется автоматическая рассылка уведомлений оператору системы взаимодействия и поставщику сервиса при диагностировании ошибок;
* в автоматизированном режиме выполняются задачи предоставления аналитических отчетов по результатам работы интеграционной шины с возможностью группировки, сортировки и фильтрации данных.

## 5.4. Требование к эксплуатации приложения

**WAF.** Web Application Firewall — межсетевой экран прикладного уровня, основной задачей которого является защита приложений, доступных online. Сперва может показаться, что это очень «ограниченное» решение, направленное на защиту одного класса приложений. Однако практически каждый слышал о переходе в digital, одним из аспектов которого является предоставление сервисов клиентам по всем возможным каналам в любое время, в том числе через веб.

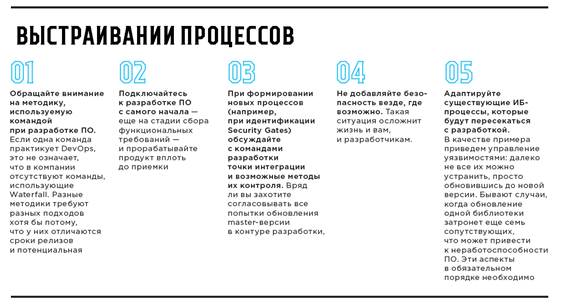
**Задача WAF — контроль http/https-трафика между пользователем и веб-приложением.**

Он позволяет идентифицировать и защищать приложения от наиболее известных атак, информация о которых регулярно публикуется OWASP (Open Web Application Security Project), например:

* инъекции, они же «внедрение кода»;
* некорректная аутентификация;
* ошибки в настройке;
* межсайтовый скриптинг (XSS);
* подделка межсайтовых запросов (CSRF);
* использование компонентов с известными уязвимостями;
* недостаточные логирование и мониторинг.

WAF выступает в роли щита между веб-приложением и пользователем, что позволяет перехватывать http/https-трафик и реализовывать действия согласно настроенным политикам — например, блокировать трафик, если была идентифицирована попытка SQL-инъекции.  В таком случае трафик не дойдет до конечного потребителя (веб-приложения) и работоспособность сервиса не нарушится или не будет совершена кража конфиденциальных данных компании. В другом сценарии WAF разрешает только «доверенный» трафик, который был заранее определен (whitelisting-подход).

Ниже приведена модель требований для разработки ПО согласно методологии информационной безопасности.



## 5.5. Требования к обработке и хранению данных

### 5.5.1. Персональные данные

Передаваемые данные между сервером и клиентом, а также между сервер приложением и СУБД, должны быть зашифрованы. Алгоритм и ключ шифрования должна быть максимально секретным и иметь длину ключа не менее 1024 Бит.

Все данные системы, которые относятся к категориям персональных данных работников КГК, должны физически храниться и обрабатываться только на территории КР согласно закону о персональных данных КР. К таким категориям данных относятся данные, обрабатываемые в системах управления персоналом (HR) и расчета заработных плат работников (Payroll).

Персональные данные не должны передаваться во внешние системы или во внутренние системы с доступом внешних пользователей через интеграционную платформу.

### 5.5.2. Коммерческая тайна

Все данные системы, которые относятся к категориям информации и данным коммерческого характера в системе, должны храниться и обрабатываться согласно закону КР о коммерческой тайне.

### 5.5.3. Защита машинных носителей информации

Для защиты машинных носителей информации должны выполняться следующие требования:

* Учет машинных носителей информации.
* Управление доступом к машинным носителям информации.
* Контроль перемещения машинных носителей информации за пределы контролируемой зоны.
* Исключение возможности несанкционированного ознакомления с содержанием информации, хранящейся на машинных носителях, и (или) использования носителей информации в иных информационных системах.
* Контроль использования интерфейсов ввода (вывода) информации на машинные носители информации.
* Контроль ввода (вывода) информации на машинные носители информации. Контроль подключения машинных носителей информации.

Уничтожение (стирание) информации на машинных носителях при их передаче между пользователями, в сторонние организации для ремонта или утилизации, а также контроль уничтожения (стирания).

## 

## 5.6. Требования к администрированию

Администрирование и сопровождение оборудования, обеспечивающего криптографическую защиту каналов связи, должно производиться только участником взаимодействия либо уполномоченными им лицами.

В целях обеспечения защиты информации, содержащейся в информационных системах, подключенных к системе взаимодействия, участники информационного взаимодействия:

* обеспечивают при обслуживании информационных систем, подключенных к интеграционной шине, исполнение установленных требований по информационной, производственной, технологической и противопожарной безопасности;
* осуществляют контроль доступа посторонних лиц к техническим средствам и каналам связи в контролируемой зоне участника взаимодействия, включая время проведения ремонтных работ и уборки помещений;
* обеспечивают обслуживание информационных систем, подключенных к интеграционной шине, только лицами, имеющими право доступа к информации, содержащейся в указанных информационных системах;
* принимают необходимые и достаточные меры, исключающие доступ посторонних лиц к защищаемой информации, в том числе парольной и ключевой информации, хранящейся на используемых и отчуждаемых носителях информации;
* осуществляют учет лиц, имеющих доступ к конечному оборудованию, обеспечивающему криптографическую защиту каналов связи интеграционной шины, расположенной в контролируемой зоне участника взаимодействия, а также лиц, имеющих возможность изменения конфигурации информационных систем данного участника взаимодействия, подключенных к интеграционной шине.

В целях обеспечения полноценного функционирования интеграционной шины и подключенных к ней информационных систем каждый участник взаимодействия:

* обеспечивает возможность оперативного переключения на резервный канал с сохранением функций обеспечения безопасности информации для всех каналов связи, выход из строя которых может существенно повлиять на доступность информационных систем, подключенных к интеграционной шине;

обеспечивает возможность оперативной замены оборудования, обеспечивающего криптографическую защиту каналов связи, используемых участником взаимодействия для осуществления информационного обмена в рамках интеграционной шины, в случае выхода такого оборудования из строя.

# 6. Список задач для интеграции

## 6.1. Список задач по интеграции для PoC (Proof of Concept)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Потребитель**  **(Информационная система, которая запрашивает / потребляет данные)** | **Поток данных / Сущность**  **(Сущность, если известно атрибутный состав)** | **Поставщик**  **(Информационная система – источник данных для потребителей** | **Способ интеграции**  **(синхронный, асинхронный)** | **Частота обновления данных** | **Механизм интеграции**  **(ETL – сервис, наименование API, MQ, dB link, xml, soap ui)** | **Цель интеграции** |
| 1 | ERP система, профайл пользователей и работников | HR данные работника: Табельный номер (ID), ФИО, должность, тип работника (штатный/по контракту), ID руководителя, контакты, кост центр, подразделение, Дата приема, Дата увольнения (если по контракту). | Система управления человеческими ресурсами | Асинхронный | Событийно | API (через Json или XML) | Обеспечить HR информацию по сотрудникам ERP для оформления документов, транзакций, заявок, утверждений и тд |
| 2 | Банк/контрагент – внешняя система Интернет-Банкинг | Банковские платежи: Наименование получателя (юр лица), наименование банка контрагента получателя, счет банка, назначение платежа, сумма платежа, валюта платежа, наименование отправителя, наименование банка отправителя, счет отправителя, ЭЦП | ERP система, Финансовый модуль, Учет кредиторской задолженности | синхронно | Событийно | API в XML формате | Для проведения банковского платежа поставщикам товаров и услуг, предоставляемых для КГК системе интернет-банкинга |
| 3 | ERP система, Финансовый модуль, Бухгалтерия | Транзакционные данные:  ID журнала, ID линии, дата транзакции, период, счет/кост центр, природа счетов, номер (ID) работника, сумма, назначение транзакции, | Payroll система | Синхронно | Событийно | API в XML формате | Для проведения бухгалтерских проводок в главной книге ERP системы по начисленным заработным платам |

## 6.2. Список задач по интеграции после завершения проекта.

Данные задачи не планируется реализовать в рамках проекта построения интеграционной платформы. Они представлены в данном документе лишь для представления о том какие еще задачи предстоят разработать в построенной интеграционной платформе.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Потребитель**  **(Информационная система, которая запрашивает / потребляет данные)** | **Поток данных / Сущность**  **(Сущность, если известно атрибутный состав)** | **Поставщик**  **(Информационная система – источник данных для потребителей** | **Способ интеграции**  **(синхронный, асинхронный)** | **Частота обновления данных** | **Механизм интеграции**  **(ETL – сервис, наименование API, MQ, dB link, xml, soap ui)** | **Цель интеграции** |
| 4 | ERP система, модуль Техобслуживания и ремонта | Мото часы, километражи и поломки по техникам: ID техники, наименование техники, дата и время, текущие моточасы или километраж, время простоя, причина поломки | Система диспетчеризации производтсвенного отдела “Dispatch/Minesense” | Асинхронно | По расписанию | DB Link или API | Импорт мото-часов и километража по оборудованиям из производственных систем в ERP |
| 5 | ERP система, модуль планирования и бюджетирования | Фактические и плановые показатели производства: Наименование карьера, тоннаж выкопанного грунта, тоннаж отхода, тоннаж низкосортной руды, тоннаж руды с низким содержанием золота, тоннаж всей произведенной руды, тоннаж отработанной руды и тд | Система планирования и управления горным производством горного отдела (DataMine). | Асинхронно | По расписанию | API в XML формате | Передача фактических и плановых данных из производственных систем в ERP систему в финансовый модуль для составления бюджета и планирования |
| 6 | ERP система, модуль планирования и бюджетирования | Фактические и плановые показатели производства: Кол-во извленного и выплавленного золота,  Содержание золота в руде  Время работы и простоя фабрики  Статистика по шаровой мельнице  Флотация  Кол-во затраченных реагентов | ЗИФ – производственная система фабрики (MileStone, FoxBoru) | Асинхронно | По расписанию | API в XML формате | Передача фактических и плановых данных из производственных систем в ERP систему в финансовый модуль для составления бюджета и планирования |
| 7 | ERP система, Финансовый модуль, Учет кредиторской задолженности | Электронный инвойс:  Инвойс номер, наименование поставщика товаров/услуг, наименование банка поставщика, номер счета поставщика, сумма, дата инвойса, наименование нашей организации, номер счета контрагента, Бенефициар, ИНН | Система электронного документооборота КР | Синхронно | Событийно | API в XML формате | В рамках единого электронного документооборота по указу КР, все эл инвойсы должны пройти через единую службу правительства КР. |
| 8 | Система электронного документооборота КР | Электронный инвойс:  Инвойс номер, наименование поставщика товаров/услуг, наименование банка поставщика, номер счета поставщика, сумма, дата инвойса, наименование нашей организации, номер счета контрагента, Бенефициар, ИНН | ERP система, Финансовый модуль, Учет дебиторской задолженности | Синхронно | Событийно | API в XML формате | В рамках единого электронного документооборота по указу КР, все эл инвойсы должны пройти через единую службу правительства КР. |
| 9 | ERP/Профайл пользователя | Профайл пользователя: EmployeeID с AD Account Name (samAccountName),  Почтовые адреса  Отдел, Локация, Менеджер, Статус (активный \ заблокированный) | ИТ Система учетных записей пользователей (Active Directory) | Синхронно | Событийно | API | Для аутентификации и авторизации пользователя в ERP через AD нужно интегрировать его через AD SSO и MFA |
| 10 | Payroll система | Табеля работников:  ID работника, дата и время, часы или дни, единица учета времени (в часах или днях), количество часов или дней, код оплаты, код премиального, код локации, кост центр или проект код, наряд, задача наряда | HR система | Синхронно | Событийно | API через JSON | Для расчета заработной платы необходимо импортировать табеля (учет раб времени) работников из системы HR в систему расчета ЗП |
| 11 | Фабрика данных (Data Factory) – хранилище данных аналитической платформы | Производственные данные и финансовые данные со всех систем, которые требоуются показать в едином портале аналитической и статистической отчетности КГК | Все производственные и финансовые системы (Datamine,  Woodgrove,  Milestone,  Minesense,  ERP,  HR,  Payroll и тд) | Асинхронно | По расписанию | REST API, SOAP | В рамках интеграционной и аналитической платформы планируется сбор, обработка, хранение и показ всех данных, необходимых для отчетности. Предполагается цифровой двойник/копия всех систем. |
| 12 | Система расчета заработной платы работников (Payroll) | HR данные работника:  ID работника, ставка/оклад, кост центр, локация, грейд по окладу, дата приема, и тд | Система управления человеческими ресурсами (HR) | Синхронно | Событийно | API REST/SOAP | HR данные необходимо передавать в Payroll по мере добавления или изменения для правильного и своевременного расчета ЗП работнику. |
| 13 | Система управления человеческими ресурсами (HR) | Данные сотрудника о посещении Рудника: ID работника, Вход/выход в Рудник, дата/время | Система управления охраны (Security system GuardPoint) | Асинхронно | По расписанию | API через JSON | Необходимо интегрировать систему охраны с системой управления человеческими ресурсами для сверки внесенных табелей по отношению к фактическим данным, зарегистрированных в системы охраны Рудника |
| 14 | Active Directory | Active Directory информация пользователя в системе учетной записи AD КГК:  ID работника, ФИО, ID руководителя, локация, наименование должности, статус, дата приема, наименование отдела/управления, и тд | Система управления человеческими ресурсами (HR) | Асинхронно | Событийно | API через SOAP | Необходимо интегрировать системы HR с AD КГК чтобы пользователь в системе имел актуальные HR данные и статус. Данная интеграция актуальна особенно для блокировки пользователя если статус в HR системе не активный. |

# 7. Требование к обучению

Необходимо иметь план и методологию обучения следующих категорий пользователей:

1. Разработчики (Бэкенд, Фронтенд, Интеграторы, по отчетам)
2. Администраторы (СУБД, Инфраструктуры)
3. Конечные пользователи

В виде показательного примера, требуется разработать несколько задач по интеграции и обработке данных вместе с разработчиками ИТ КГК. При необходимости нужно организовать курсы по основным технологиям.

Отдельно требуется обучение администраторов и обеспечить демонстрационные примеры выполнения задач. При необходимости требуется организовать курсы.

По конечным пользователям, необходимо разработать план первоначального обучения и регулярного обучения по обновлению знаний. Можно применять специальные информационные системы или ПО по обучению пользователей через веб портал, где тренеры могут разработать и выложить видео и другие материалы, по которым пользователи могут самостоятельно ознакомиться, пройти тесты, и тем самым получить право на доступ к системе.