Требование к аналитической   
платформе ИТ КГК

# Содержание

[1. Требование к архитектуре 3](#_Toc96065656)

[1.2. Функциональная архитектура аналитической платформы 4](#_Toc96065657)

[1.3. Программно-компонентная архитектура аналитической платформы 6](#_Toc96065658)

[1.4. Требования к протоколам обмена данными. 7](#_Toc96065659)

[1.5. Операционная система – платформа 7](#_Toc96065660)

[1.6. Расположение аналитической платформы 7](#_Toc96065661)

[1.7. Развертывание и обновление платформы 8](#_Toc96065662)

[2. Технологии сбора данных. 8](#_Toc96065663)

[2.1. ETL процесс. 8](#_Toc96065664)

[2.2. ELT процесс 9](#_Toc96065665)

[3. Требование по классам критичности и отказоустойчивости 12](#_Toc96065666)

[3.1. Классификатор IT-Систем по приоритету восстановления 12](#_Toc96065667)

[3.2. Классификатор IT-Систем по типу обработки отказов 12](#_Toc96065668)

[3.3. Классификатор IT-Систем по времени восстановления и доступности за год 13](#_Toc96065669)

[3.5. Типовой архитектурный шаблон “High Speed+ (RC2)” для аналитической платформы 16](#_Toc96065670)

[4. Требование по своду правил ИТ КГК 20](#_Toc96065671)

[4.1. Общие требования ИТ КГК 20](#_Toc96065672)

[4.2. Принципы архитектуры данных 22](#_Toc96065673)

[4.3.1. Принципы архитектуры прикладного программного обеспечения (информационных систем) 24](#_Toc96065674)

[4.4. Принципы Технологической Архитектуры 26](#_Toc96065675)

[5. Требование по информационной безопасности 27](#_Toc96065676)

[5.1. Общие требования к системе 27](#_Toc96065677)

[5.2. Требования к обработке и хранению данных 28](#_Toc96065678)

[5.3. Требование к эксплуатации приложения 30](#_Toc96065679)

[6. Поставщики данных КГК для аналитической платформы. 31](#_Toc96065680)

# 1. Требование к архитектуре

Архитектура аналитической платформы ИТ КГК должна отвечать принципам и требованиям свода правил и классов стандартов ИТ КГК, а также включать все возможности для расширения и развития в перспективе.

Архитектура аналитической платформы должна включать следующие основные компоненты:

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

Архитектура аналитической платформы должна обеспечивать:

* Отсутствие влияния на работу первичных (транзакционных) систем (abuse исключить);
* Возможность собирать данные со всех производственных и вспомогательных систем без ограничения в объеме передаваемой и хранимой информации, при необходимости горизонтально расширяясь (scale out);
* Поступление данных в платформу и далее к презентационному слою (web и desktop приложения, инструменты аналитики) за требуемое время (Real time, Near Real time);
* Сбор данных с датчиков на мобильной технике (самосвалы, экскаваторы, буровые и т.п.), с датчиков на фабрике, SCADA систем и пр.;
* Сбор логов с серверов, desktop’ов, мобильных устройств, SaaS приложений;
* Отсутствие потерь данных на этапе сбора и хранения;
* Хранение структурированных и не структурированных данных;
* Наличие брокера сообщений (Service Broker), с возможностью:
  + поточного анализа real-time, на этапе сбора;
  + корректного отката и пере-заливки данных с первоисточника;
  + симуляции активности в тестовой среде через транзакционные логи (replay).
* Возможность оптимально сбалансировать Data flow (pipelines) через общедоступный или кастомный ETL manager;
* Наличие Data Lake для хранения данных в оригинальном формате.

## 1.2. Функциональная архитектура аналитической платформы

Функциональная архитектура платформы должна состоять из следующих блоков (сервисов):

* Сервис администрирования платформы;
* Сервисов сбора и хранения данных;
* Сервисов структурирования и обработки данных;
* Пользовательского интерфейса;
* Управления развертыванием и разработкой.

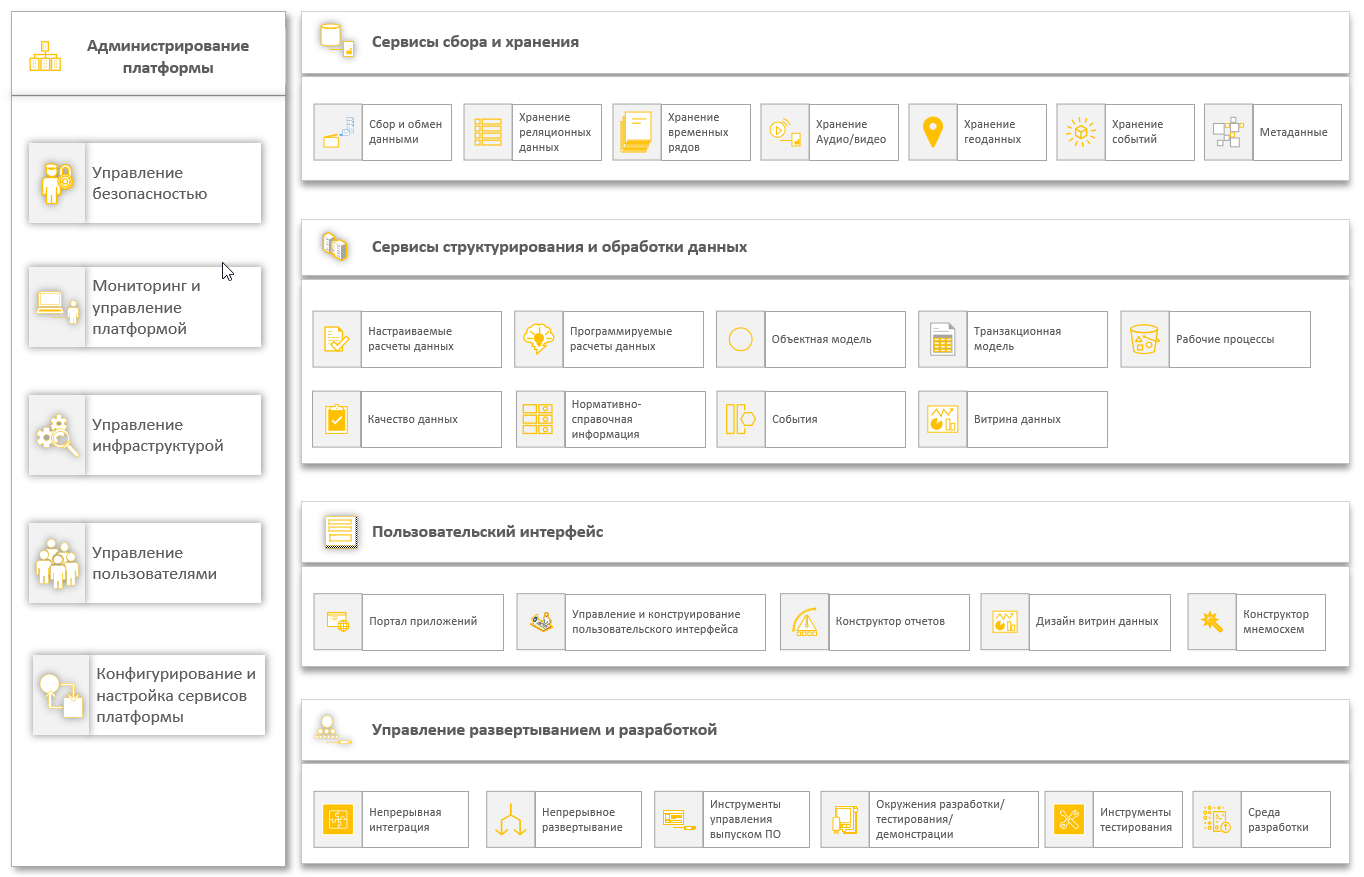
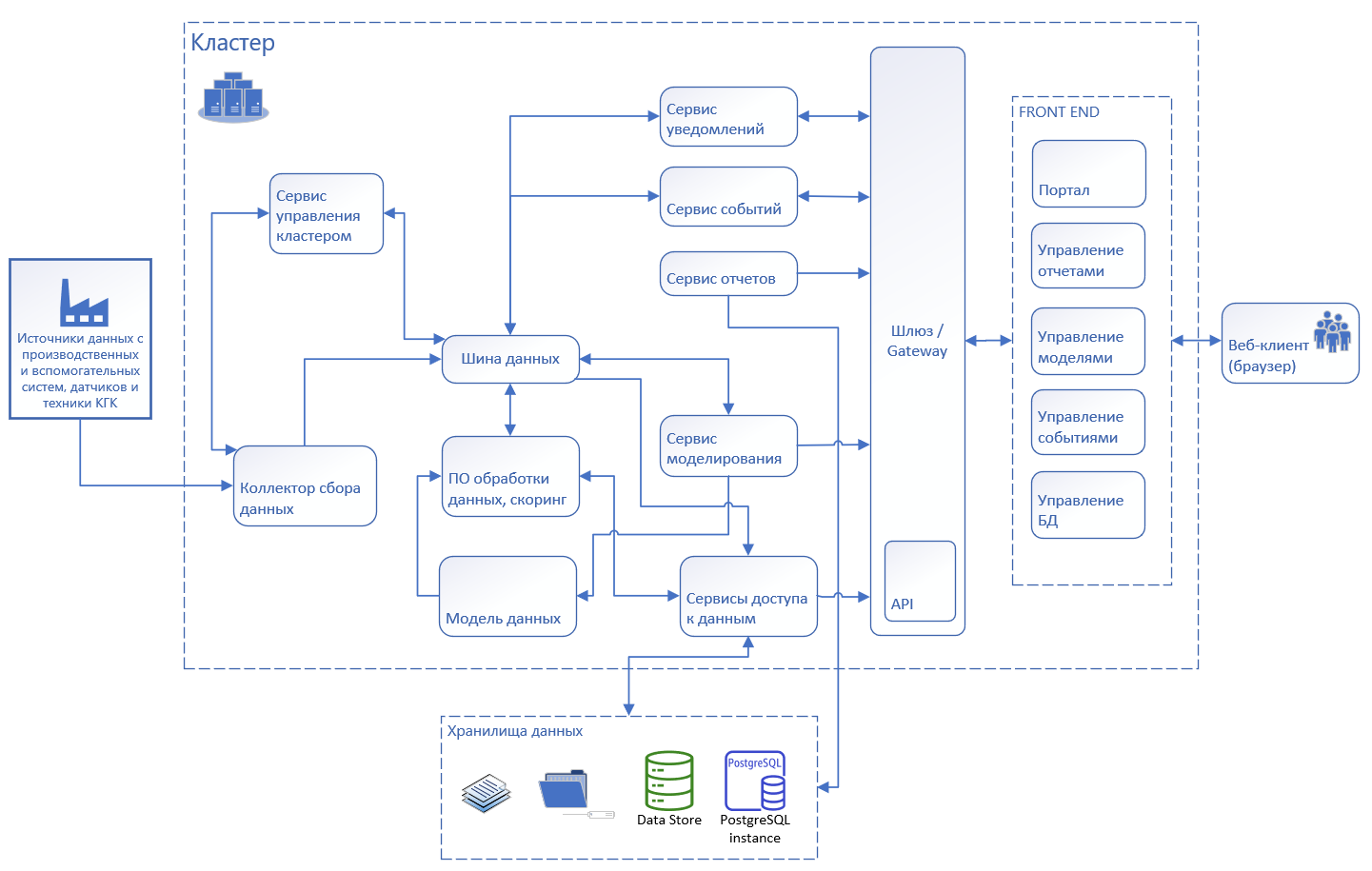
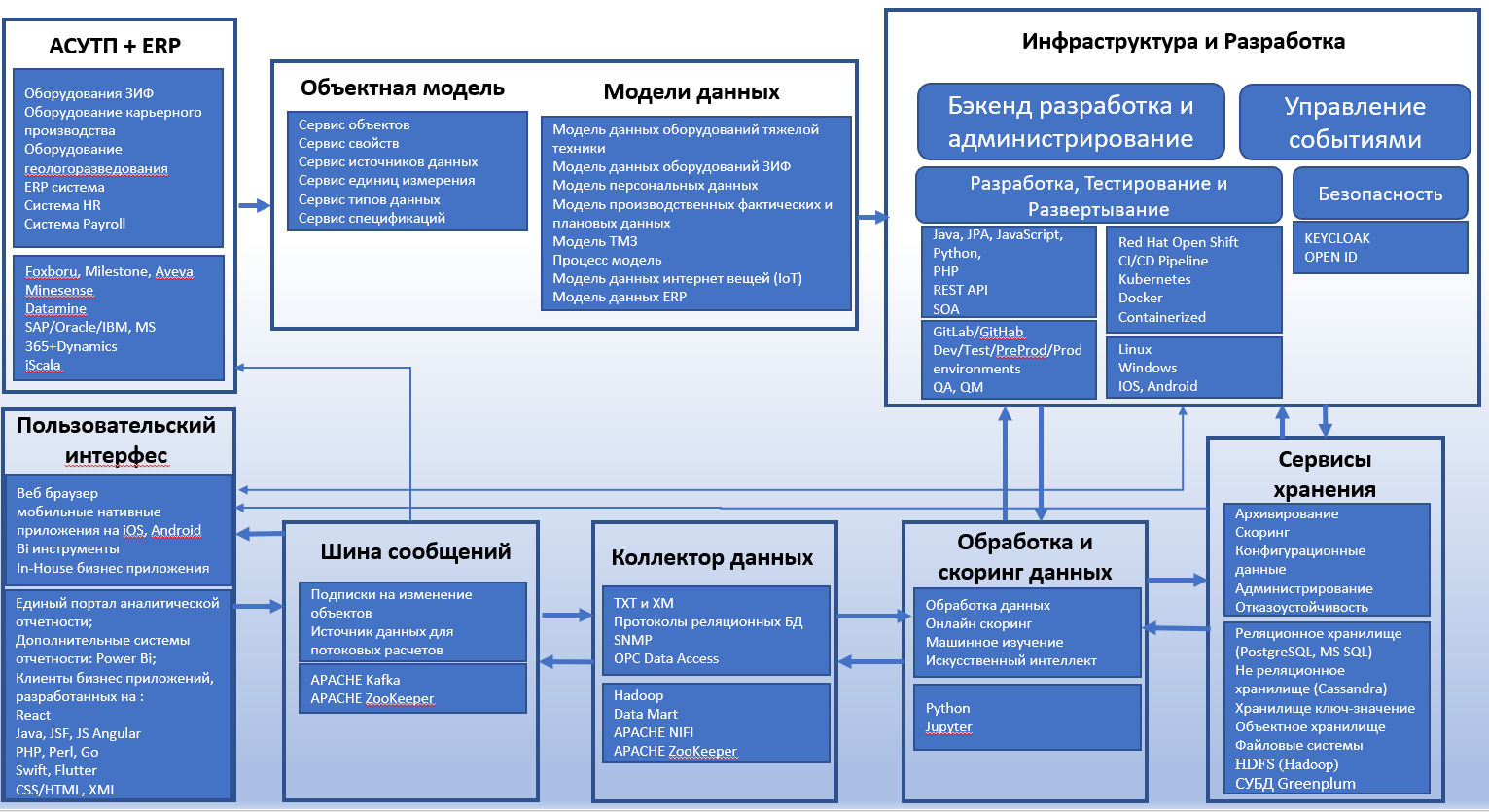


Схема взаимодействия сервисов аналитической платформы:



## 1.3. Программно-компонентная архитектура аналитической платформы

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



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

## 1.4. Требования к протоколам обмена данными.

Аналитическая платформа должна обеспечивать сбор данных с любых источников и, соответственно, располагать возможностью подключения к этим источникам как посредством Native Clients drivers, так и посредством API.

Внутренний обмен данными в платформе должен осуществляться посредством API (например, использование микросервисов на Java SpringBoot).

В платформе может быть использован один из следующих форматов для передачи данных: JSON, XML, TXT\CSV.

В качестве технологий и стандартов обмена данными должна быть поддержка технологий RESTful API, gRPC, SQL.

Передача данных между системами должна быть осуществляться через шифрованные каналы (SSL\TLS).

## 1.5. Операционная система – платформа

Клиентская часть приложения должна быть кроссплатформенной. Предпочтительная платформа – web клиент (Chrome, MS Edge, Firefox), но в зависимости от задач следующие платформы должны поддерживаться: IOS, Android, Windows, Mac OS, Linux.

Серверная часть платформы должна быть основана на облачных технологиях и поддерживать технологию IaC (Infrastructure as Code) и при необходимости легко мигрировать или объединять разные облачные платформы (cloud native vendor neutral). Использование контейнеризации является основой для реализации компонентов платформы. В качестве оркестровки контейнеризированных приложений необходимо использовать Kubernetes.

## 1.6. Расположение аналитической платформы

Аналитическая платформа может быть развернута на серверах КГК или в центрах обработки данных (арендованных или находящихся в собственности КГК), физически расположенных на территории Кыргызской Республики.

**Примечание**: Приоритетным направлением для КГК является постройка и использование собственных центров обработки данных. КГК рассматривает возможность наличия двух центров обработки данных: основного, расположенного непосредственно на руднике Кумтор и резервного в г. Бишкек.

## 1.7. Развертывание и обновление платформы

Развертывание и обновление платформы необходимо реализовать через CI\CD pipeline с автоматическим тестированием и последующим развертыванием \ откатом изменений через Version Control System (VCS). Развертывание тестовых сред также должно реализовываться через VCS, через отдельный репозиторий (для защиты от несанкционированного изменения). Предпочтительно использование удобного инструментария на основе Gitlab, Jenkins и подобных.

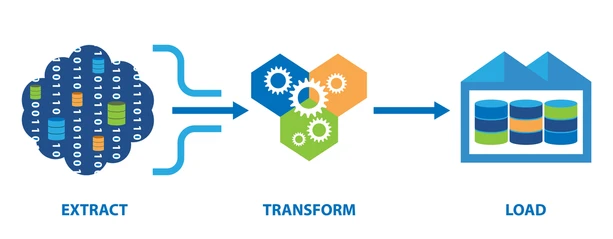
# 2. Технологии сбора данных.

При росте количества источников данных становится сложно вручную загружать и трансформировать данные. Именно для этих задач используются ETL решения.

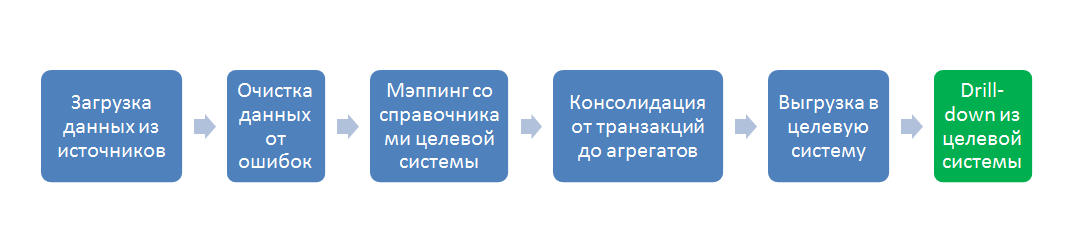
## 2.1. ETL процесс.

ETL (Extract, Transform, Load) – это совокупность процессов управления хранилищами данных, включая:

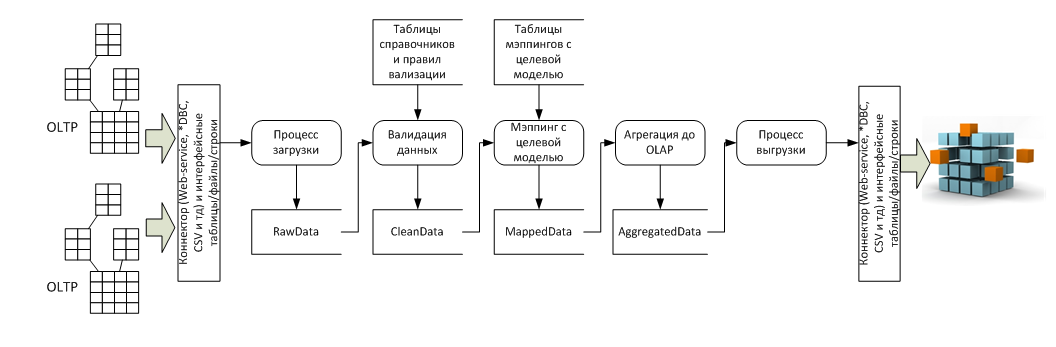
* извлечение данных из внешних источников (таблицы баз данных, файлы);
* преобразование и очистка данных согласно бизнес-потребностям;
* загрузка обработанной информации в корпоративное хранилище данных (КХД).



При разработке ETL системы КГК следует придерживаться следующих основных функций ETL процессов, представленных на блок схеме:



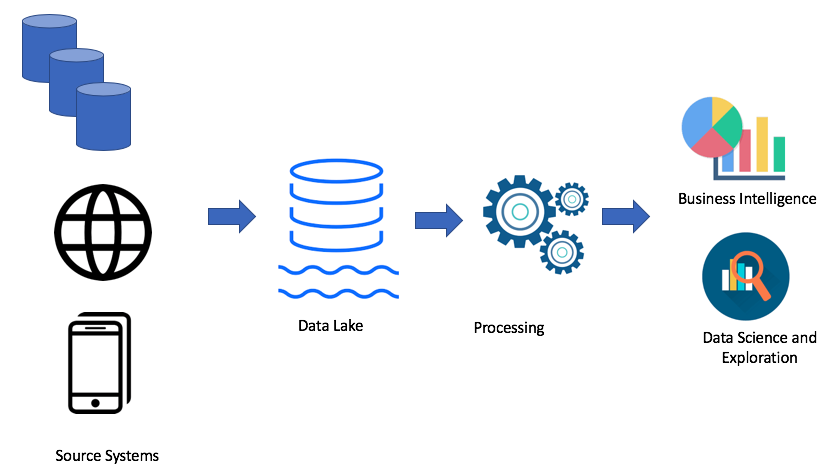
В разрезе потока данных это несколько систем-источников (обычно OLTP) и система приемник (обычно OLAP), а также пять стадий преобразования между ними:



1. Процесс загрузки – его задача затянуть в ETL данные произвольного качества для дальнейшей обработки, на этом этапе важно сверить суммы пришедших строк, если в исходной системе больше строк, чем в RawData, то значит, загрузка прошла с ошибкой;
2. Процесс валидации данных – на этом этапе данные последовательно проверяются на корректность и полноту, составляется отчет об ошибках для исправления;
3. Процесс мэппинга данных с целевой моделью – на этом этапе к валидированной таблице пристраивается еще n-столбцов по количеству справочников целевой модели данных, а потом по таблицам мэппингов в каждой пристроенной ячейке, в каждой строке проставляются значения целевых справочников. Значения могут проставляться как 1:1, так и \*:1, так и 1:\* и \*:\*, для настройки последних двух вариантов используют формулы и скрипты мэппинга, реализованные в ETL-инструменте;
4. Процесс агрегации данных – этот процесс нужен из-за разности детализации данных в OLTP и OLAP системах. OLAP-системы — это, по сути, полностью денормализованная таблица фактов и окружающие ее таблицы справочников (звездочка/снежинка), максимальная детализация сумм OLAP – это количество перестановок всех элементов всех справочников. А OLTP система может содержать несколько сумм для одного и того же набора элементов справочников. Можно было-бы убивать OLTP-детализацию еще на входе в ETL, но тогда мы потеряли бы «аудиторский след». Этот след нужен для построения Drill-down отчета, который показывает — из каких строк OLTP, сформировалась сумма в ячейке OLAP-системы. Поэтому сначала делается мэппинг на детализации OLTP, а потом в отдельной таблице данные «схлопывают» для загрузки в OLAP;
5. Выгрузка в целевую систему — это технический процесс использования коннектора и передачи данных в целевую систему.

## 2.2. ELT процесс

В ряде случае при работе с огромным объемом данных стоит рассмотреть возможность использования ELT процессов.



ELT означает «Извлечь, загрузить и преобразовать». В этом процессе данные для основных преобразований используются через хранилище данных. Это означает, что нет необходимости в промежуточном размещении данных. Для всех типов данных, включая структурированные, неструктурированные, полу-структурированные и даже необработанные данные ELT использует облачные решения для их хранения.

**Процесс ELT работает с озерами данных**. «Озера данных» – это особые виды хранилищ данных, которые, в отличие от хранилищ OLAP, принимают любые структурированные или неструктурированные данные. Озера данных не требуют преобразования данных *перед* их загрузкой. В озеро данных можно загрузить любой **тип** *необработанной* информации, независимо от формата или его отсутствия.

Однако перед анализом данных с помощью аналитической платформы все еще необходимо преобразование данных. Очистка, обогащение и преобразование данных происходят после загрузки данных в озеро данных. Ниже приведены факты о работе ELT и озерах данных:

* **Новая технология, ставшая возможной благодаря высокоскоростным облачным серверам:** ELT – это относительно новая технология, ставшая возможной благодаря современным облачным серверным технологиям. Облачные хранилища данных предлагают практически безграничные возможности хранения и масштабируемую вычислительную мощность. Например, такие платформы, как Amazon Redshift и Google BigQuery, делают возможными конвейеры ELT благодаря своим невероятным возможностям обработки.
* **Принимайте данные, как только они становятся доступными:** ELT в сочетании с озером данных позволяет сразу же принимать постоянно расширяющийся пул необработанных данных, как только они становятся доступными. Нет необходимости преобразовывать данные в специальный формат перед их сохранением в озере данных.
* **Преобразует только те данные, которые вам нужны:**ELT преобразует только данные, необходимые вам для конкретного анализа. Хотя оно может замедлить процесс анализа, озеро данных предлагает вам большую гибкость, потому что вы можете преобразовывать данные разными способами на лету и так создавать различные типы метрик, прогнозов и отчетов. И наоборот, с ETL весь конвейер ETL, и структура данных в хранилище OLAP, может потребовать модификации, если ранее принятая структура не позволяет проводить новый тип анализа.

# 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.

***При разработке требований к внедрению классы критичности и отказоустойчивости могут быть пересмотрены согласно требованиям к доступности той или иной задачи.***

## 3.1. Классификатор IT-Систем по приоритету восстановления

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

Примечание: **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 отказоустойчивости | Имя шаблона отказоустойчивости | Описание шаблона | Код класса системы исполь-зующий данный шаблон | Регламенти-рованный % доступности системы за год (SLA) | Макс. допустимое время простоя системы за год | \*Восстановление в случае локального сбоя системы | | \*Восстановление в случае падения основного ЦОДа | | Кол-во 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 суток 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 | Все сотрудники компании | IT-Система, пользователями которой являются все сотрудники компании. |

### 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 аналитической 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 суток 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.

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**

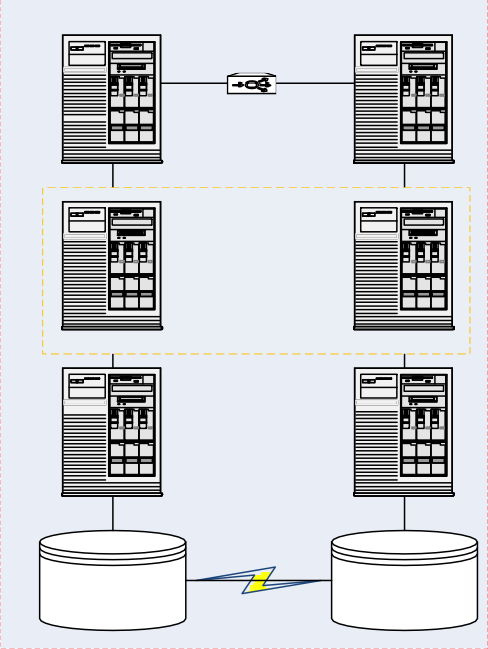
**HE WLET T**

Для систем 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
4. «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;
5. для данных Систем необходимо использовать подключенные через SAN СХД верхнего\среднего уровня (не внутренние диски и не DAS);
6. одна копия данных приложения должна располагаться на дисковом массиве на другой площадке, т.е. настроена СХД реплика;
7. для разных копий данных необходимо использовать разные СХД;
8. необходимы решения PVG striping / Secure Path;
9. необходимо использовать минимум 2 (два) выделенных сетевых адаптера для передачи данных с внешними сервисами;
10. необходимо выделенное сетевое подключение, используемое для целей РКД для серверов приложений =>2Tb;
11. необходимо полное, документированное решение по восстановлению после сбоев (Disaster Recovery);
12. необходимо проводить тестирование восстановления данных согласно утверждённого и подписанного плана тестирования РКД;
13. необходим действующий контракт(техподдержка) на обслуживание ПО и аппаратных средств со стороны Вендора ПО\оборудования (время реагирования 24 часа или меньше);
14. 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;
15. РКД следует осуществлять в другом ЦОД (возможно, расщеплением зеркала в SAN или через локальную сеть на медиа-сервер);
16. каналы связи между комплектами оборудования должны быть шифрованными.

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

**Disaster recovery**



**Основной ЦОД**

Web

Server №1

Web

Server №2

Web балансировщик

Application

Server №1

Кластеризация

Application

Server №2

Active DataBase

Server №1

Active DataBase

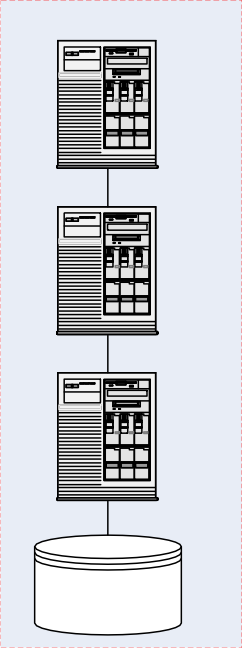
Server №2

СХД №1

Реплика СХД

СХД №2

Автоматическое переключение



**Резервный ЦОД**

Web Server №3

Application Server №3

Standby Database

Server №3

СХД №3

Ручное переключение

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

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 КГК, а именно:

* 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 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 бизнеспричин 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 сервисытаким 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 бизнеспроцессах 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 системе

Система должна иметь функцию авто логирования и журналирования всех изменений / событий на стороне сервера. Такая функция должна хранить следующих данных как минимум: Идентификатор пользователя, время, имя запущенной функции, имя команды, входные/выходные параметры, возможные ошибки и т.д.

Система должна иметь админ приложение для управления и администрирования доступов, ролей и групп пользователей в разрезе функций, приложений и модулей ПО. Модули и приложения системы должны быть доступны конечным пользователям согласно матрице доступа (CRUD). Кроме этого, необходимо иметь наличие консоль приложения на стороне сервера для управления супер конфиденциальными настройками и доступами.

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.

Аутентификация пользователей в системе должна иметь возможность подключения к Active Directory (AD) компании и выполнить аутентификацию по SSO с применением MFA.

Авторизация пользователей должна осуществляться согласно настройкам модуля администрирования пользователей и прав доступа.

Аутентификация и авторизация пользователей должна быть многофакторной, точнее должна быть двухфакторной.

## 5.2. Требования к обработке и хранению данных

### 5.2.1. Персональные данные

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).

### 5.2.2. 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.

### 5.2.3. 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).

### 5.2.4. Авторизация и аутентификация

Система должна предоставлять механизм, позволяющий администратору создавать, изменять и удалять учетные записи пользователей в выделенном собственном каталоге пользовательских учетных записей, и должна отвечать следующим требованиям:

* Предоставлять единую точку входа с возможностью настройки MFA для конечного пользователя.
* Аутентификация пользователей должна выполняться средствами платформы, желательно интеграция с Active Directory.
* Предоставлять доступ на основе Role Based Access Control (RBAC).
* По требованию бизнес-процесса обеспечивать возможность настройки Row Level Security (RLS) на уровне пользователя, группы, отдела, компании.
* Для решения поставленных задач должны предоставляться минимально необходимые права доступа.
* Функциональные компоненты платформы, выполняющие фоновые задачи (aka services or daemons) должны работать под выделенными сервисными аккаунтами.

### 5.2.5. Изолирование данных по авторизации и аутентификации

Изоляция данных и исполнение нормативных требований пользовательских данных для размещения в выделенной базе данных. Система должна предоставлять механизм аутентификации и авторизации пользователей в изолированной области базы данных, настроенной специально по требованиям, которые принадлежат пользователям. База данных клиента должна быть изолирована от баз данных других клиентов. Обработка данных клиента должна происходить в выделенном сегменте, изолированном от других систем провайдера. Данные должны храниться только в зашифрованном виде. Передача данных должна происходить только по защищенным каналам и протоколам. Доступ в базу данных должен предоставляться авторизированным и аутентифицированным пользователям.

### 5.2.6. 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. 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.

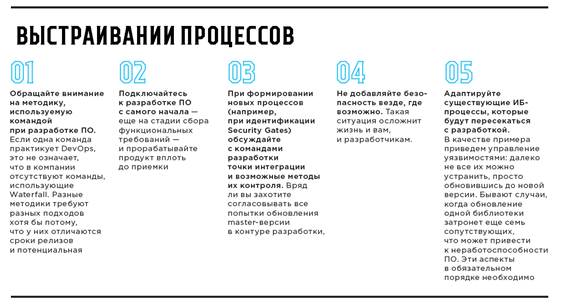
**Задача WAF — контроль http/https-трафика между пользователем и веб-приложением.**

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.

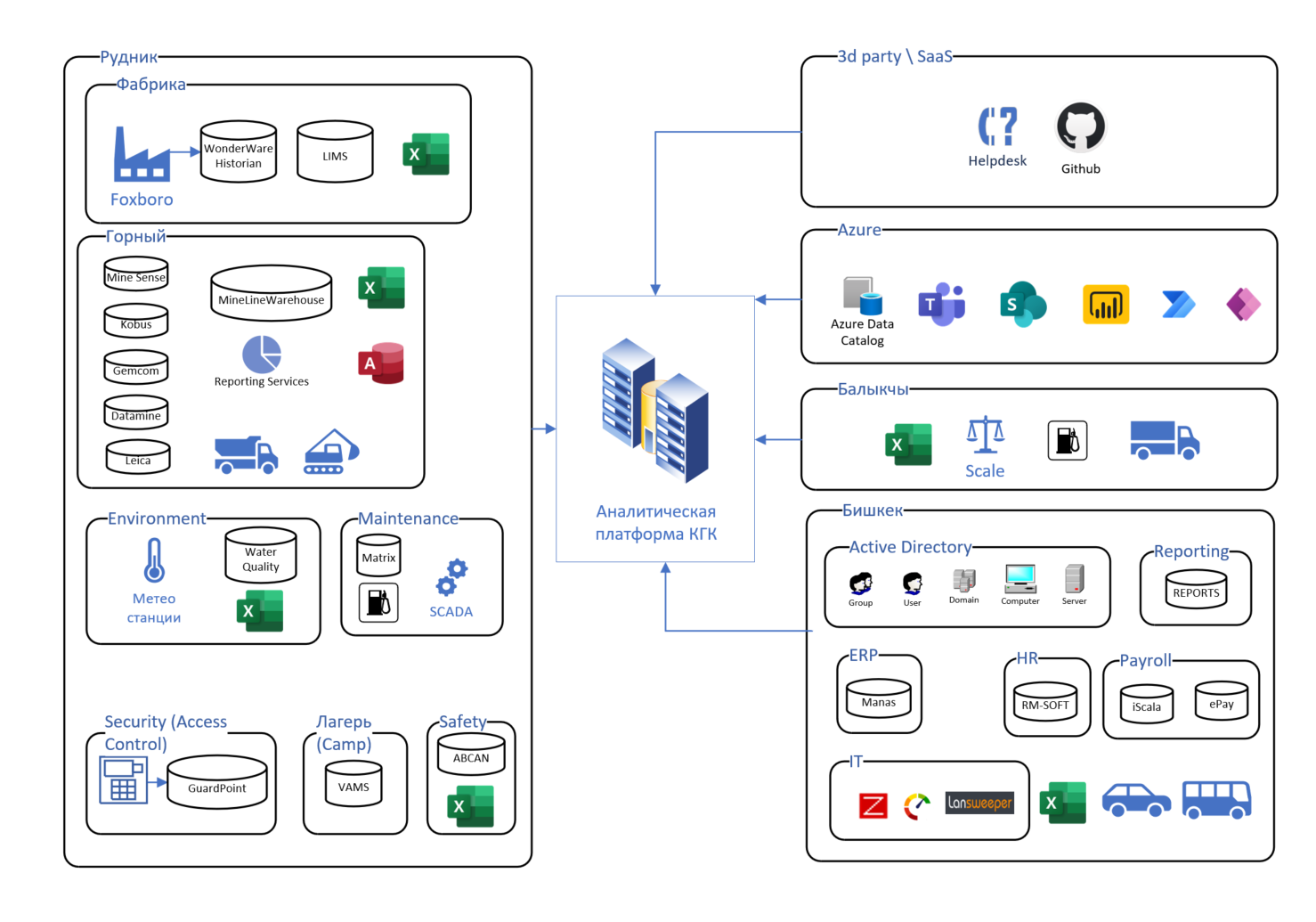
WAF выступает в роли щита между веб-приложением и пользователем, что позволяет перехватывать http/https-трафик и реализовывать действия согласно настроенным политикам — например, блокировать трафик, если была идентифицирована попытка SQL-инъекции.  В таком случае трафик не дойдет до конечного потребителя (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.



# 6. Поставщики данных КГК для аналитической платформы.

Ниже представлена диаграмма основных поставщиков данных КГК для аналитической платформы:



В настоящий момент в качестве источников данных могут выступать файлы, однако задача состоит в отказе в хранении данных в файлах и переноса ввода и хранения непосредственно в хранилище данных.

Ниже представлен детальный список основных поставщиков (источников) и потребителей данных, включая описание объектов, способа передачи и частотой обновления:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поставщик (Информацион-ная система – источник данных для потребителей) | Способ передачи данных от поставщика до *хранилища данных* | Частота передачи данных | Передавая сущность. Data Object. | Потребитель (Информационная система, которая запрашивает / потребляет данные) | Способ получения Потребителем данных из *хранилища данных* (ETL – сервис, наименование API, MQ, DB Link, XML, SOAP UI) | Приоритет для внедрения на 1 этапе. |
| Mine Sense (система управления парком техники) | DB Link | Каждые 10 минут | * **Смены**   + ID;   + Смена (ночная \ дневная);   + Время начала смены;   + Время окончания смены;   + Бригада. * **Самосвалы**   + ID;   + Модель;   + Номинальная грузоподъемность;   + Тип кузова. * **Экскаваторы**   + ID;   + Модель;   + Марка. * **Рейсы**   + ID;   + Самосвал;   + Экскаватор;   + Тоннаж;   + Порода (руда \ лед \ пустая порода и т.п.);   + Карьер;   + Отвал;   + Время приезда;   + Время ожидания;   + Время погрузки;   + Время в пути;   + Время разгрузки;   + Время дистанция;   + Время скорость. * **Состояние техники**   + ID техники;   + Статус (в работе\ поломка \ задержка и т.п.);   + Локация;   + Оператор техники;   + Дата начала события;   + Дата окончания события;   + Продолжительность события. * **Сигналы тревоги (Alerts)**   + Тип;   + ID техники;   + Локация;   + Оператор;   + Уровень тревоги (alert level);   + Код VIMS. | Отчетно-аналитическая система | DB Link или API | Да |
| Kobus (система управления буровыми машинами) | DB Link | Каждые 10 минут | * **Буровые станки**   + ID;   + Модель;   + Марка. * **Записи результатов бурения**   + ID техники;   + Скважина;   + метраж;   + время начала;   + время завершения;   + оператор;   + порода (руда \ лед \ пустая порода и т. п.). * **Состояние техники**   + ID техники,   + статус (в работе\ поломка \ задержка и т. п.),   + локация,   + оператор техники,   + дата начала события   + дата окончания события,   + продолжительность события | Отчетно-аналитическая система | DB Link или API | Да |
| Convoy | DB Link | 1 раз в день | * **Информация по конвою**   + Конвой № (первый \ второй \ промежуточный и т .д.)   + Описание груза   + Тип контейнера \ прицепа   + Комментарии   + Кол-во грузовиков | Отчетно-аналитическая система | DB Link или API | Нет |
| Blast | SharePoint List via HTTPS | 1 раз в день | * **Информация по взрывным работам**   + Дата   + Локация (рабочая зона, взрывной блок)   + Комментарии | Отчетно-аналитическая система | DB Link или API | Нет |
| MP5 | CSV file | 1 раз в день | * **Метеоданные**   + Дата и время   + Метеостанция   + Температура   + Атмосферное давление   + Влажность   + Осадки (мм) | Отчетно-аналитическая система | DB Link или API | Нет |
| Mill | Excel | 1 раз в день | * **Статистика по фабрике**   + Питание Шаровой Мельницы   + Тоннаж   + Содержание в питании   + Расчетное сквозное извлечение   + Расчетное извлеченное золото   + Выплавленное золото   + Общие конечные хвосты   + Оперативное время   + УМ Переработка   + УМ Извлеченное золото   + Измельчение   + Флотация * **УВР концентрата** | Отчетно-аналитическая система | DB Link или API | Да |
| Mine | DB Link | Каждые 10 минут | * **Производительность рудника**   + Производительность по карьеру итоговая   + Производительность по карьеру по рабочим зонам   + Перемещение внутри карьера, дороги & проекты * **Production KPIs**    + Производительность   + Скорость   + Производсственный цикл   + Качество данных   + Производительность   + Задержки   + Оператор   + Горячая смена и т.д. | Отчетно-аналитическая система | DB Link или API | Да |
| Maintenance | DB Link | Каждые 10 минут | * **Техобслуживание**   + Использование тяжелой техники   + Готовность тяжелой техники   + Производительность экскаваторов | Отчетно-аналитическая система | DB Link или API | Да |
| Манас ERP system | DB Link | По требованию | * **Промежуточная выгрузка для последующего анализа в Excel**   + Финансовые отчеты   + Отчеты по бюджету и планированию   + Складские отчеты   + Отчеты по заказам и поставщикам   + Отчеты по нарядам   + Отчеты по реагентам, топливу и другим расходным материалам | Отчетно-аналитическая система | DB Link или API | Нет |
| Манас ERP system | DB Link | 1 раз в день | * **Аналитические модели**   + Planning   + Inventory   + Cost Data Reports   + Fuel Consumption | Отчетно-аналитическая система | DB Link или API | Да |
| Safety department | Excel / PDF file | 1 раз в день | * **Статистика по безопасности**   + Выявленные источники опасности (Hazards)   + Несостоявшиеся проишествия (Near Misses)   + Материальный ущерб   + Первая помощь   + Медицинская помощь   + Потеря рабочего времени * **Daily Safety Flash** | Отчетно-аналитическая система | DB Link или API | Нет |