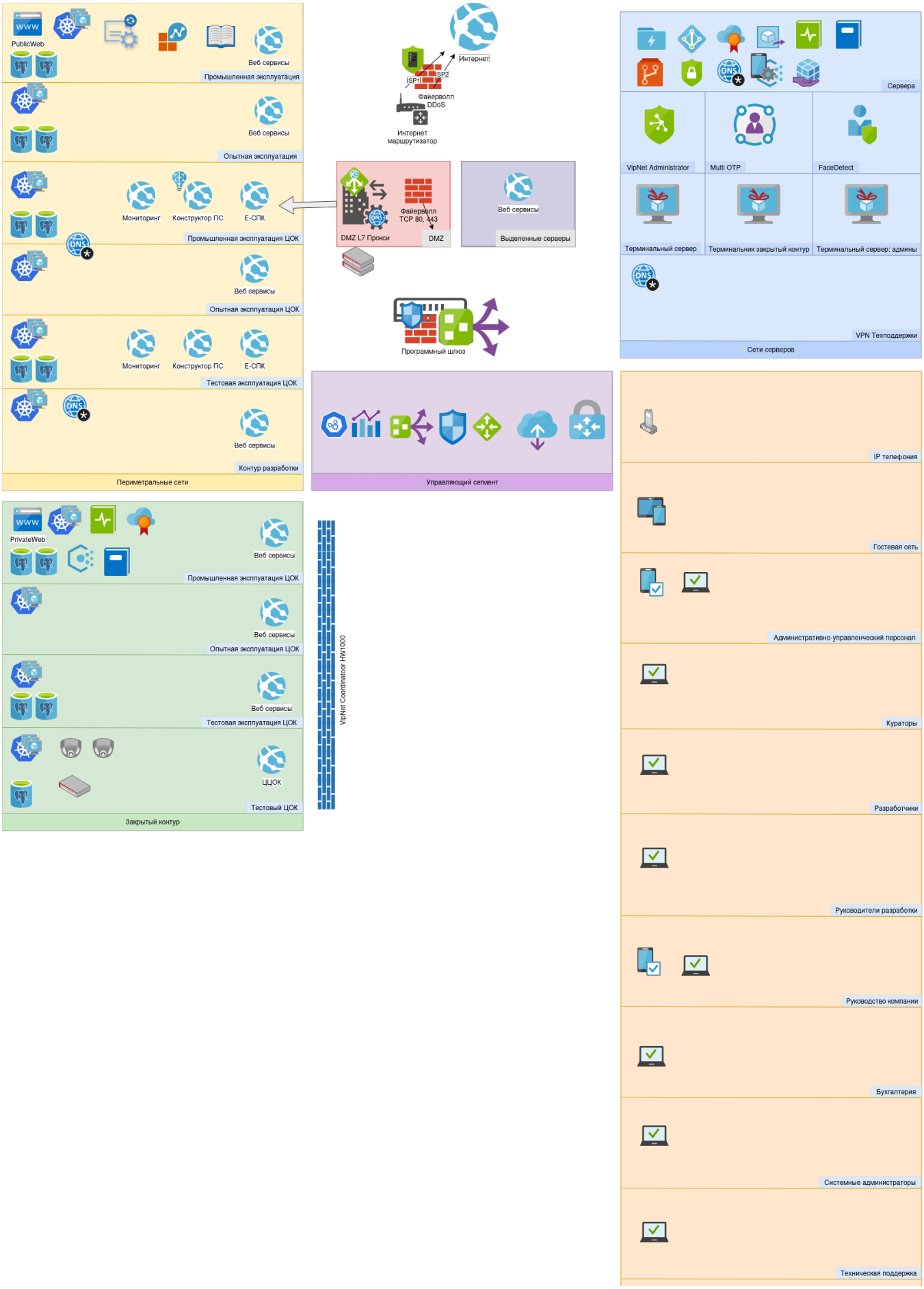
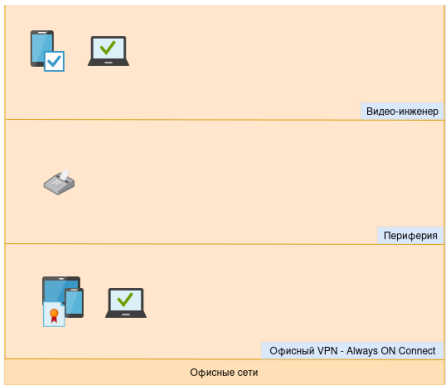
Новая система НСК

Топология размещения и взаимодействия сервисов НСК




Видео-инженер
Периферия
Офисный VPN - Always ON Connect
Офисные сети

Механизмы CI/CD

Базовые механизмы конвеера постоянной интеграции и развертывания программных продуктов АО "Национальные квалификации" состоят ис следующих технологических этапов:

Тестовая эксплуатация

1. Публикация изменений исходного кода
2. Запуск по расписанию (2 часа ночи) или по триггеру процесса сборки ПО
3. Запуск набора автотестов (юнит-тестирование)
4. Упаковка установочного пакета в формате debian
5. Упаковка установочного пакета в формате tarball+setup.sh
6. Передача собранного deb пакета в репозиторий ПО "Тестовая эксплутаация"
7. Установка или обновление собранного пакета в среде тестовой эксплуатации по расписанию на 5 утра по Московскому часовому поясу или по триггеру.

После успешного завершения всех этапов программное обеспечение доступно в среде тестовой эксплуатации. Если хотя бы один этап завершается с ошибкой, владельну проекта отправляется отчет с уведомлением о возникшей ошибке.

Опытная и промышленная эксплуатация

1. После успешного тестирования ПО в тестовом контуре тестировщик подписывает Акт тестирования. В Акте фигурирует версия ПО, включающая номер коммита (зафиксированных изменений) и тег версии ПО.
2. После подписания акта тег версии присваивается номеру коммита.
3. Запускается процесс сборки ПО с версией согласно тегу.
4. Формируется журнал изменений из событий между тегами. в журнал попадают только события отмеченные символами +/-/* в начале каждой строки описания изменений.
5. Запускается набор автотестов (юнит-тестирование)
6. Упаковка установочного пакета в формате debian
7. Упаковка установочного пакета в формате tarball+setup.sh
8. Передача собранного deb пакета в репозиторий ПО "Опытная эксплутаация"
9. Отправка уведомлений кураторам и владельцам среды опытной эксплуатации об обновлении ПО включая список изменений.
10. Обновление ПО в среде опытной эксплуатации при помощи агента Puppet
11. После завершения тестирования ПО ведущий разработчик, технический директор или начальник отдела технической поддержки совместно с двумя кураторами регионов подписывают Акт тестирования. Всего устанавливается 3 подписи разными видами уполномоченных лиц.
12. После подписания акта тестирования ПО передается в репозиторий "Промышленная эксплуатация"
13. Отправка уведомлений кураторам и владельцам среды промышленной эксплуатации об обновлении ПО включая список изменений.

Документация

Документация должна автоматически собираться на дату сборки и формирования пакетов в формате PDF с использованием системы справки и поддержки АО НК. Файл документации должен автоматически формировать шапку, подвал, типографику, сноски, версию документа и иные параметры верстки и допечатной подготовки, согласно заданному шаблону.

Сборка программного обеспечения

Для каждого разрабатываемого продукта АО НК должен формироваться Debian пакет программного обеспечения для целей быстрого точеченого развертывания, контроля версионности, зависимостей и целостности распространяемого ПО.

Требования к формируемым пакетам:

1. Имя ПО является базовой частью имени пакета: <software>
2. Если пакет содержит компилируемую часть, то все компилируемые файлы должны входить в пакет <software>-bin
3. Если пакет содержит набор данных отдельно от программной части - изображения, шрифты, база

данных, то все компоненты должны входить в пакет <software>-data
4. Если пакет содержит набор типовых настроек, такие настройки включаются в базовый пакет <software>, если вараинтов настройки пакета несколько, создаются соответствующие конфигурационные пакеты вида: <software>-config-<kind>, где *kind* - разновидонсть предоставляемых настроек.
5. Документация на пакет ПО упаковывается в пакет <software>-doc
6. При наличии SDK, такие файлы должны помещаться в пакет <software>-dev
7. Если пакет имеет изменяемую структуру БД (миграции), то такие миграции должны быть включены в базовый пакет <software>
8. Версия пакета формируется автоматически из последнего тега + порядкового номера коммита относительно тега. ХЭШ коммита попадает в сведения о сборке пакета.
9. Список изменений проекта формируется автоматически из строк фиксации изменений кода в git, при этом учитываются только строки начинающиеся с символов +/-/*
10. Правила формирования пакетов описываются DevOps инженером совместно с руководителем проекта или тимлидом.

Формирование Docker контейнера

Docker файл формирует образ программного обеспечения как результат сборки debian пакета. Docker файл должен предоставлять simple контейнер, из одного или нескольких образов ФС:

- Базовая ОС
- Необходимые зависимости
- Образ с распакованынм ПО
- Сценарии при запуске контейнера (преинит)

Итоговый контейнер должен соответствовать следующим принципам:

1. Монолитность Отсуствие сборки ПО при установке контейнера
2. Стабильность Отсуствие скачивания дополнительных компонент из сети интернет
3. Повторяемость Возможность многократно развернуть контейнер в средах отличных от Docker
4. Атомарность Иметь отдельные образы контейнеров для нужд БД и рабочей нагрузки
5. Масштабируемость Иметь поддержку горизонтального масштабирования (NLB) с общей БД, разделяемой ФС и конфигурацией
6. Управляемость Контейнер должен содержать базовый набор Unix Shell утилит, на уровне не ниже Busybox 1.7

Репозиторий образов

Для целей поддержки инфраструктуры Kubernetes для программного обеспечения выделяется общий разделяемый репозиторий корпоративного ПО. Перед передачей ПО в ГосТех или иные сторонние площадки оркестрации и управления ПО **должно** пройти все необходимые процедуры тестирования и приемки в среде **Тестовой** и **Опытной** эксплуатации.

Аутентификация, Авторизация, Аудит

Общие положения

Основной задачи системы AAA/IAM является управление базой учетных записей пользователей, выдача разрешений и аудит доступа к ресурсам связанных с IAM подсистем.

Основные механизмы аутентификации базируются на принципах протокола Kerberos v5 с расширениями (Claims), применительно к системам веб аутентификации (jwt), включая возможности делегирования, пересылки токена, двусторонней аутентификации, запроса расширений авторизации по сервисному имени принципала (SPN).

Использование механизмов, завязанных на принципала, с использованием технологии электронных подписей совместимых со стандартом JWT позволяют проводить следующие способы аутентификации пользователя:

- Аутентификация по логину-паролю (ключевая пара)
- Сквозная аутентификация по сертификатам электронной подписи
- Аутентификация по одноразовому ключу (SMS, OTP)
- Аутентификация по оффлановому токену (QR-код) без доступа к IAM
- Аутентификация по сохраненному токену (Progressive Web Application, HTML5 Offline Application)
- Прокси-аутентификация со стороннего сервиса (OAuth2/SAML 2.0)

Общий формат записи токена аутентификации:

```
{
  "principal": "User Principal Name",
  "uid": "GUID",
  "displayName": "User Published Name",
  "groups": [...],
  "claims": [...],
  "principalSignature": "jwt digest/RSA|ECDSA 2048bit",
  "signature": "jwt digest/RSA|ECDSA 2048bit"
}
```

Поля *Groups*, *Claims* и *PrincipalSignature* могут отсутствовать. В качестве обязательного идентификатора пользователя выступают поля *Principal* и *UID*

Архитектура

Архитектура сервиса IAM предполагает наличие модульного механизма работы с поддержкой горизонтального масштабирования. Сервис IAM состоит из нескольких независимых максштабируемых сервисов:

- **AAA** — Сервис аутентификации, авторизации и аудита. Отвечает за выдачу токенов, производных токенов (для каждого SPN) и запись событий доступа.
- **Profile** — Сервис управления профилем пользователя, хранит сведения о профиле пользователя, позволяет хранить ряд дополнительных данных, таких как связаныне документы, изображения, дополнительные реквизиты и т.п.
- **Permission** — Сервис управления доступом. Управляет видами групп доступа для каждого сервиса и ролями доступными пользователю.

Общая схема работы OAuth2 сервиса:

Сервер поддерживает как обязательные (базовые) механизмы OAuth2, так и расширенные (опциональные) механизмы аутентификации и делегирования токена. В качестве расширений механизмов аутентификации вводятся дополнительные сущности - UserPrincipalName, OrganizationPrincipalName и ServicePrincipalName, где UPN и OPN относятся к категории субъекта аутентификации, а SPN - к категории конечной точки аутентификации. OAuth2 токен (Bearer) формируется по стандарту JWT и может использоваться в оффлайн операциях в течение всего срока (времени жизни) такого токена.

При прохождении процедуры аутентификации в системе цифрового паспорта пользователь получает базовый цифровой ключ доступа в виде JWT токена, кодированного в Base64 для передачи по техническим каналам связи или в открытом виде для формирования оффлайн QR кода. Допускается использовать формирование QR кода на основе Base64 кодированного JWT токена, для поддержки данного требовани рекомендуется использовать следующий алгоритм:

1. Попытка расшифровать JSON
2. Если расшифровать JSON не удается (ошибка в 1 символе) - производится декодирование Base64 в строку
3. Осуществляется повторная попытка декодирования JSON
4. Проверяется корректность подписи JWT
5. Проверяется срок действия JWT

После того как базовый токен аутентификации получен, он может использоваться для доступа к различным сервисам, с минимальным уровнем гарантированных привилегий. В случае если сервис поддерживает механизмы работы оффлайн, он может содержать реплику групп безопасности сервиса для аутентификации пользователя. Для получения полного доступа к ресурсам после процедуры аутентификации производится:

1. Выбор роли пользователя в системе - присвоение UID или OID и расширенных атрибутов (расширенный JWT токен) - утверждений
2. Расширенная аутентификация в сервисе - получение SPN для запрашиваемого ресурса
3. Получение специального токена для запрашиваемого ресурса по его SPN

Общий набор полей всех JWT токенов является одинаковым, но для каждого типа токенов характерны выделенные специальные поля. Состав и форматы полей JWT токенов описаны ниже.

Набор функций сервиса зависит от совпадения набора требований и утверждений сервиса и JWT токена для доступа к тому или иному функционалу приложения. Любой полученный токен имеет свой срок действия, может быть обновлен в преедах окна обновления и содержит валидную (проверяемую) цифровую подпись системы цифрового паспорта АО НК. Любой токен может быть использован оффлайн для аутентификации пользователя, например для каждого сервиса может быть сгенерирован отдельный временный оффлайн токен доступа в виде QR кода. Стандартный период действия токена составляет 2 часа, базового токена - 8 часов. При необходимости может быть запрошен токен расширенного срока действия, но не более 168 часов (7 дней) с момента получения такого токена.

При делегировании токена нижестоящей системе, промежуточный узел может как передать токен в режиме "как есть" (прокси-токен) или получить на его основании новый SPN токен для конечного сервиса. В этом случае для получения нового SPN сервис обращается к системе цифрового паспорта за делегированием токена передавая:

- Токен пользователя (SPN JWT)
- Адрес конечного сервиса

Срок действия делегированного токена сервиса не должен превышать срок использования оригинального токена пользователя.

Возможность продления истекшего JWT токена не допускается, из такого токена могут быть получены основные реквизиты для запроса подтверждения (аутентификации) со стороны пользователя с целью выпуска нового JWT токена с теми же условиями (Владелец, Организация, Сервис).

Попроектная кадрово-организационная структура

Технические требовния на систему управления задачами

Цели создания системы

Основной целью создания системы является систематизация управленческих и рабочих задач в рамках существующих процессов и реализующихся проектов компании. Сопутствующей целью создания системы - является прослеживаемость выполняемых задач сотрудниками компании для оптимизации трудовых ресурсов.

Вводимые термины и определения

В рамках настоящей технической спецификации вводятся основные термины и определения:

- **Процесс** — Решаемая глобальная задача имеющая одну или несколько целей, несколько исполнителей, характеризующаяся отсутствием конечного срока и набора подзадач. Подзадачи процесса формируются по мере поступления информации от участников такого процесса или явлсются следствием выполнения других (связанных) подзадач. Каждая подзадача процесса всегда направлена на достижение конкретной цели и имеет свой приоритет выполнения. Может затрагивать проекты и формировать дополнительные промежуточные цели проектов.
- **Проект** — Любое разрабатываемое решение, такое как: программное обеспечение, аппаратный комплекс, инженерное сооружение, законодательный акт или иной субьект права собственности, являющийся конечным продуктом проекта. Проект всегда завершается созданием артефакта в форме нематериального актива или основного средства. Проект может быть заранее подергнут процессу декомпозиции на промежуточные цели, задачи, подзадачи, проектные команды. В рамках проекта формируются и закреплаются артефакты, такие как техничекие спецификации, технические задания, архитектурные планы, дизайн-макеты и иные сопутствуюшие документы необходимые для целей реализации проекта.
- **Промежуточная цель** — Этап *Процесса* или разработки *Проекта* имеющий собственное описание, точные или оценочные сроки выполнения цели и набор вводных и результирующих документов (артефактов) цели.
- **Задача** — Сформулированное конечное требование для одного или нескольких исполнителей. Задача имеет следующий набор атрибутов:
  - Наименование задачи
  - Описание задачи
  - Вложенные документы
  - Дата постановки задачи
  - Дедлайн задачи
  - Состояние задачи
  - Приоритет задачи
  - Проект/Процесс задачи
  - Промежуточная цель задачи
  - Автор задачи
  - Ответственный по задаче
  - Исполнители задачи
  - Родительская задача

Задача может иметь декомпозицию в виде дерева подзадач, для каждой из которых могут быть назначены свои ответственные и исполнители. Список исполнителей и документов всех подзадач всегда отражается в родительской задаче.

- **Календарь** — Датированное представление задач в разрезе сохраненного фильтра, имеющий ссылку для доступа по протоколу CalDAV для чтения расписания и управления задачами. Цели обозначаются выделенной датой на календаре специальной рамкой и цветом.
- **KanBan Доска** — Постолбцовое представление задач в разрезе сохраненного фильтра. Столбец определяется условием группировки задач, может быть одним из атрибутов задачи (Состояние, Приоритет, Проект/Процесс, Цель, Автор, Ответственный, Дата постановки, Дедлайн), при этом для дат используется округление до недели, месяца или квартала по выбору пользователя. Задачи могут быть отсортированы по дате постановки, дедлайну, состоянию, приоритету, автору или ответственному. Цель выводится как одно из полей задачи.
- **Диаграмма Ганта** — Линейное представление задач в разрезе сохраненного фильтра. Данная диаграмма имеет вид хронологического горизонтального представления календаря, разделенного на Годы, Кварталы, Месяцы, Недели, Дни недели, где задачи представлены в виде плашек, распространяющихся от даты постановки задачи до её дедлайна или назначенной промежуточной цели. Порядок пересекающихся задач, приходящихся на один период времени может быть отсортирован по дате постановки, состоянию, приоритету, автору или ответственному. Цели отображаются в виде сплошных вертикальных плашек, затрагивающих дату завершения промежуточной цели.
- **Список задач** — Представление задач в виде таблицы с указанием наименования задачи, автора,

ответственного, исполнителей, цели, приоритета и сроков выполнения задачи. Задачи выстраиваются в рамках иерархии подзадач. Сортировка может осуществляться по любому столбцу с учетом иерархии задач. Задачи могут быть сгруппированы аналогично KanBan доске.
- **Дашборд** — Специальное представление, задач, имеющее набор настроек внешнего вида: Гант, Доска, Календарь, Список; а так же набора фильтров по статусу, сотрудникам, датам, целям, проектам/процессам, приоритету, состоянию. Набор фильтров и внешний вид дашборда может быть сохранен как публичное или приватное представление, а так же иметь публичную ссылку в режиме "Только для чтения". Публичное представление может быть назначено на группу пользователей. При переключении настройк внешнего вида параметры фильтра и группировки сохраняются и применяются к выбранному внешнему виду.
- **Группа пользователей** — Выделенная группа пользователей, имеющих одинаковые настройки интерфейса, перечень доступных проектов и процессов, представлений и набор привилегий. Привилегии назначенные для разных групп пользователей суммируются для каждого конкретного пользователя, в случае их пересечения. Группа пользователей может иметь фильтр автодобавления, при котором все новые пользователи, соответствующие критериям фильтра автоматически будут добавлены в группу (автопополняемая группа).
- **Пользователь** — Сотрудник организации или внешний контрагент прошедший процедуру аутентификации и входящий в одну или несколько групп пользователей. Если пользователь не имеет ни одной назначенной группы, такой пользователь относится к служебной группе "Гости" и может просматривать только публичные представления.
- **Отчет** — Формируемый по расписанию срез задач, согласно заданному фильтру задач, содержит ссылку на интерактивный *дашборд* с применением указанного фильтра, в котором можно изменять внешний вид задач: Список, Календарь, Гант, Доска. Отчет может быть направлен по любому каналу уведомлений.
- **Уведомления** — Системное сообщение отправляемое по одному или нескольким каналам связи: Мгновенное сообщение (Telegram), Электронная почта, Push уведомление и т.п. Уведомления вызываются по триггеру события. Типы получаемых уведомлений настраиваются для каждого пользователя индивидуально. Уведомления по событиям могут аккумулироваться для пользователя, и отправляться одним сообщением, если для данного типа уведомлений указан период группировки сообщений и время отправки уведомления.
- **Журнал** — Системный журнал событий, включающий в себя действия пользователя, от момента аутентификации, до момента получения уведомления. Все операции изменения задачи пользователем попадают в журнал и могут быть отслежены как в отдельной задаче, так и формировать *уведомление* пользователя или группы по условию. По каждой задаче ведется контроль наличия изменений, относительно последней даты доступа пользователя к задаче и отображается пользователю путем цветового выделения задачи более ярким цветом или рамкой.
- **Чат** — Внутренняя система обмена мгновенными сообщениями. Чат является самостоятельным представлением, но может быть открыт в качестве виджета или отображаться в рамках задачи, к которой он относится. У каждой задачи может быть свой чат, который становится доступен всем участникам задачи (Автор, Ответственный, Исполнители) при отправке первого сообщения. Чаты группируются по проекту и задаче к которой они относятся. Для выполненных (завершенных) задач чат скрывается из общего списка чатов, список чатов включает в себя только активные/действующие задачи пользователя, связанного с такой задачей. Количество новых сообщений отображается в виде цифры в круглом блоке поверх соответствующщй задачи, в заголовке страницы и в окне чата для каждого чата сответственно.

Технические требования к реализации системы

Система управления задачами должна обладать набором слудующих обязательных характеристик:

- Аутентификация
  - Доменная
  - OAuth2
- Реактивность (Интерфейс зависит от поступаемых данных и отрисовывается на стороне клиента)
- Интерактивность (Все изменения задач в системе отражаются в виде событий и доставляются активным сессиям пользователей)
- Адаптивность (Возможность взаимодействия с основными функциями интерфейса как на мобильных устройствах, так и на ПК)
- Масштабируемость (Возможность запуска множества экземпляров Backend и Frontend серверов с общим центральным хранилищем, например с использованеим брокера RabbitMQ Stream)
- Поддерживаемость (Наличие сборочного сценария и документации на систему разработки и сборки проекта)
- Переносимость (Возможность развертывания в виде OnPermisse или облачного решения на стороне конечного заказчика)