Программный комплекс «Центр управления пользователями (ЦУП) версия 2.0»

Описание

Оглавление

1. Назначение системы

ЦУП предназначен для организаций, которые в соответствии с требованиями законодательства в части импортозамещения или в связи с желанием снизить стоимость владения переходят на использование в ИТ-инфраструктуре операционных систем отечественной разработки или операционных систем с открытым кодом, построенных на базе Unix/Linux решений.

ЦУП применяется для централизованного управления учетными записями пользователей, рабочими станциями и предоставлением доступа учетным записям к рабочим станциям.

ЦУП обеспечивает возможность:

- безопасного хранения секретов;
- безопасной доставки секретов на ресурсы автоматизированных систем;
- безопасного создания секретов;
- безопасной передачи секретов по запросам автоматизированных систем;
- удаление и обновление секретов;
- конфиденциальности секретов при хранении и передаче с применением криптографических методов;
- удаления секретов без возможности восстановления;
- настройки гибкой ролевой модели;
- автоматизации управления пользователями;
- автоматизации управления рабочими станциями и предоставления доступа к ним;
- автоматизации управлениями доступом к периферийным устройствам;
- интеграции с субъектами аутентификации и с механизмами аутентификации.

В ЦУП отсутствуют противоречия между лицензиями на компоненты.

ЦУП состоит из следующих компонент:

- компонент централизованного управления пользователями;
- компонент службы каталогов;
- компонент управления рабочими станциями;
- компонент подготовки рабочих станций;
- компонент сервера печати.

Компонент централизованного управления пользователями обеспечивает организацию единого доступа к функциям остальных компонентов ЦУП в рамках принципа «одного окна».

Компонент централизованного управления пользователи обеспечивает интерфейс для выполнения следующих функций:

- использование гибкой ролевой модели, в части управления пользователями:
  - отображение списка учетных записей, зарегистрированных в компоненте службы каталогов;
  - создание новой учетной записи в компоненте службы каталогов;
  - редактирование атрибутов ранее созданных учетных записей в компоненте службы каталогов;
  - удаление учетной записи в компоненте службы каталогов;
  - предоставление доступа к рабочим станциям с указанием типа доступа: пользователь, администратор;
- управление группами учетных записей:
  - отображение групп учетных записей пользователей, зарегистрированных в компоненте службы каталогов;
  - создание новой группы учетных записей пользователей в компоненте службы каталогов;
  - редактирование состава группы учетных записей пользователей, зарегистрированной в компоненте службы каталогов;

    - удаление группы учетных записей пользователей в компоненте службы каталогов;
- управление рабочими станциями:
    - отображение списка рабочих станций, зарегистрированных в компоненте службы каталогов;
    - предоставление доступа учетным записям пользователей к рабочей станции;
    - назначение периферийного устройства на рабочую станцию;
- управление группами рабочих станций:
    - отображение списка групп рабочих станций, зарегистрированных в компоненте службы каталогов;
    - создание новой группы рабочих станций в компоненте службы каталогов;
    - управление составом групп рабочих станций, зарегистрированных в компоненте службы каталогов;
    - удаление группы рабочих станций, зарегистрированных в компоненте службы каталогов;
- управление приложениями:
    - отображение приложений из централизованного репозитория дистрибутивов приложений пакетного менеджера Aptitude;
    - отображение групп приложений;
    - определения состава приложений в группе приложений;
    - удаление групп приложений;
- управление периферийными устройствами, в части отображения списка принтеров, подключенных к компоненту сервера печати;
- отображение показателей функционирования ЦУП:
    - показатели системы: количество заведенных в компонент службы каталогов учетных записей пользователей и рабочих станций;

- показатели заданий настройки рабочих станций: обновлено, успешно, неуспешно;
- интерфейс взаимодействия представляет собой тонкий клиент, разработанный с применением веб-технологий.

Компонент службы каталогов обеспечивает иерархическое представление объектов управления: учетных записей, рабочих станций и обеспечивает централизованное управление правами доступами.

Компонент службы каталогов обеспечивает реализацию следующих функций:

- учет и хранение данных учетных записей и их атрибутов;
- учет и хранение рабочих станций;
- управление правилами доступа учетных записей к рабочим станциям;
- управление правами делегирования доступа к привилегированным ресурсам рабочей станции;
- управление рабочими станциями в части:
  - управления учетными записями рабочей станции;
  - управления правами доступа на рабочей станции;
  - настройки средств аутентификации;
  - управления делегированием доступа к привилегированным ресурсам рабочей станции;
- разрешение доменных имен;
- использование службы аутентификации LDAP, Kerberos;
- использование программного интерфейса доступа к своим функциям на базе REST или веб-сервисов.

Компонент управления рабочими станциями предназначен для удаленного управления конфигурацией рабочих станций

Компонент управления рабочими станциями обеспечивает выполнение следующих функций:

- удаленный доступ к рабочим станциям по протоколу ssh;
- удаленное выполнение заданий на подключение периферийных устройств и настройку служб печати;
- удаленная установка или удаление приложений из центрального репозитория с помощью пакетного менеджера;
- установку обновлений на рабочую станцию.

Компонент подготовки рабочих станций предназначен для первичной настройки рабочей станции и регистрации ее в компоненте службы каталогов.

Компонент подготовки рабочих станций обеспечивает следующие функции:

- установка по сети операционной системы на рабочую станцию с помощью протокола PXE;
- автоматическая регистрация рабочей станции в компоненте службы каталогов;
- сетевой репозиторий с программными пакетами.

Компонент сервера печати предназначен для централизованного управления печатью с рабочих станций.

Компонент сервера печати обеспечивает следующие функции:

- регистрация сетевых принтеров;
- обеспечение функционирования очереди документов, отправляемых с рабочих станций на печать;
- взаимодействие с принтерами по сети для отправки документов на печать.

1.1.Цели и задачи системы

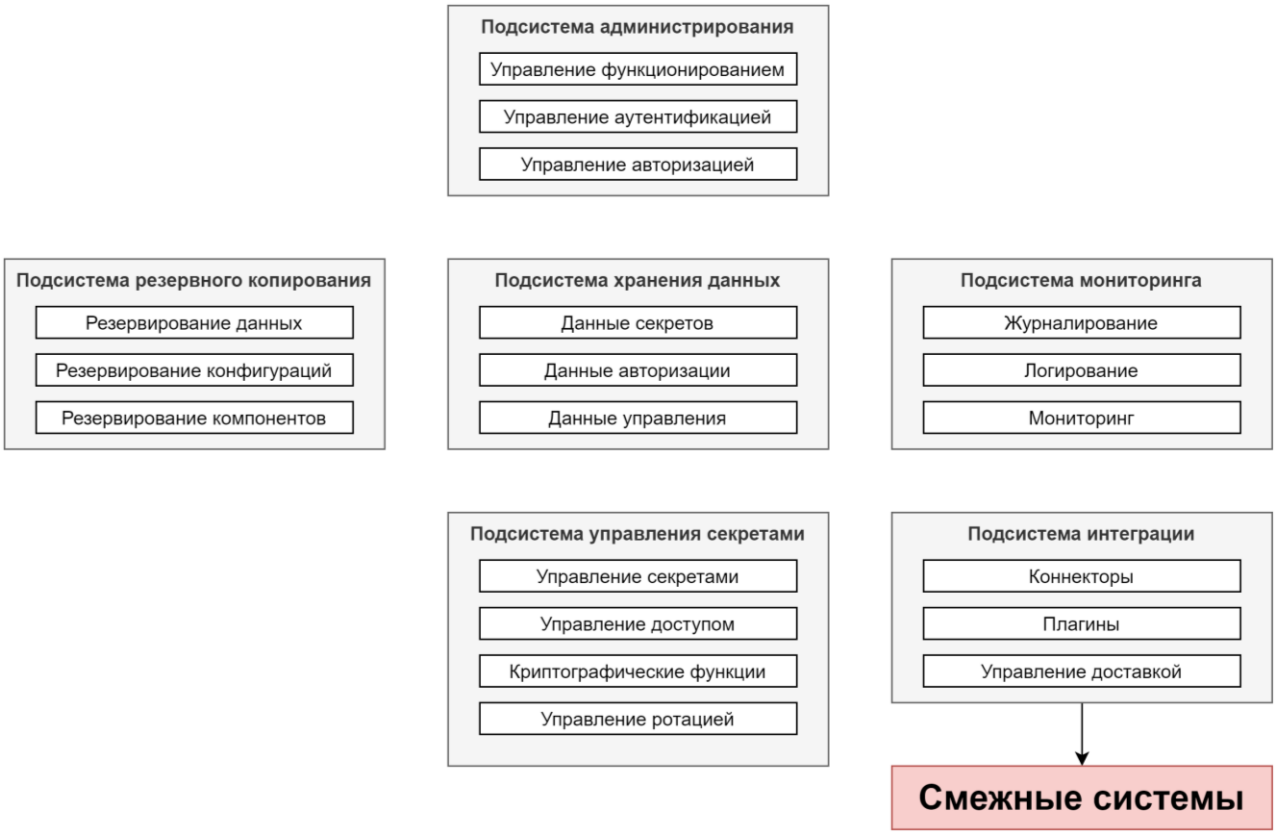
Основной целью системы является повышение уровня защищенности инфраструктуры, путем автоматизации и унификации процессов управления секретами.

Задачи:

- Безопасное хранение секретов.
- Безопасная доставка секретов на информационные ресурсы
- Безопасное создание секретов.

- Безопасная передача секретов по запросам.
- Отзыв (удаление) и ротация секретов.

1.2. Функциональная архитектура



Система состоит из набора функциональных блоков, которые отражают основные группы функциональных требований.

Решение спроектировано таким образом, что возможно использование широкого ряда систем хранения данных. Это позволяет отделить архитектуру подсистемы хранения данных от архитектуры управления секретами и считать их относительно автономными.

Поскольку синхронизация данных осуществляется на уровне подсистемы хранения данных, архитектура подсистемы управления секретами может быть спроектирована без учета требования географической распределенности.

Напротив – обеспечение мультитенантности при сохранении единого пространства данных, должно быть учтено при проектировании подсистемы управления секретами. Мультитенантность, в числе прочих механизмов, является также инструментом распределения нагрузки.

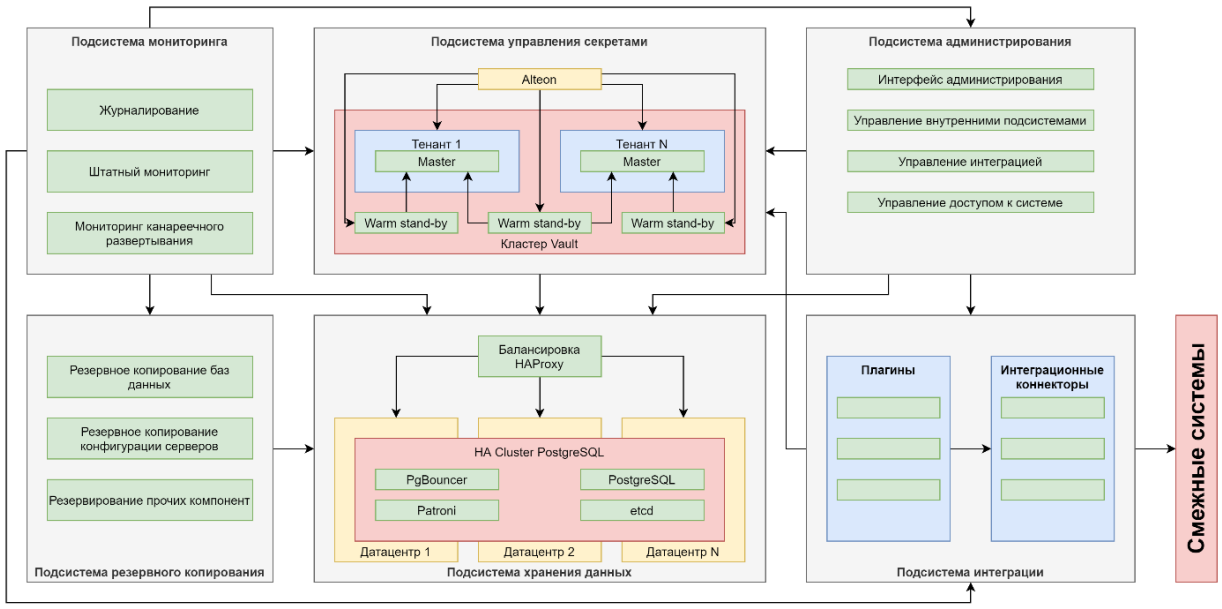
Подсистемы мониторинга и резервного копирования используют готовые компоненты. Список компонентов определяется стеком технологий Заказчика.

В качестве системы хранения выбрана СУБД PostgreSQL. Наличие большого количества дополнительных инструментов позволяет строить на ее основе кластеры

высокой надежности с резервированием компонентов и автоматизацией аварийного переключения.

1.3. Общая схема

Исходя из функциональных требований по обеспечению высокой доступности, географической распределенности и масштабирования предлагается следующая общая архитектура системы:



1.4. Подсистема управления секретами

Подсистема управления секретами реализует основные функции управления секретами, осуществляет контроль доступа к секретам, реализует плагинную систему расширения функциональности.

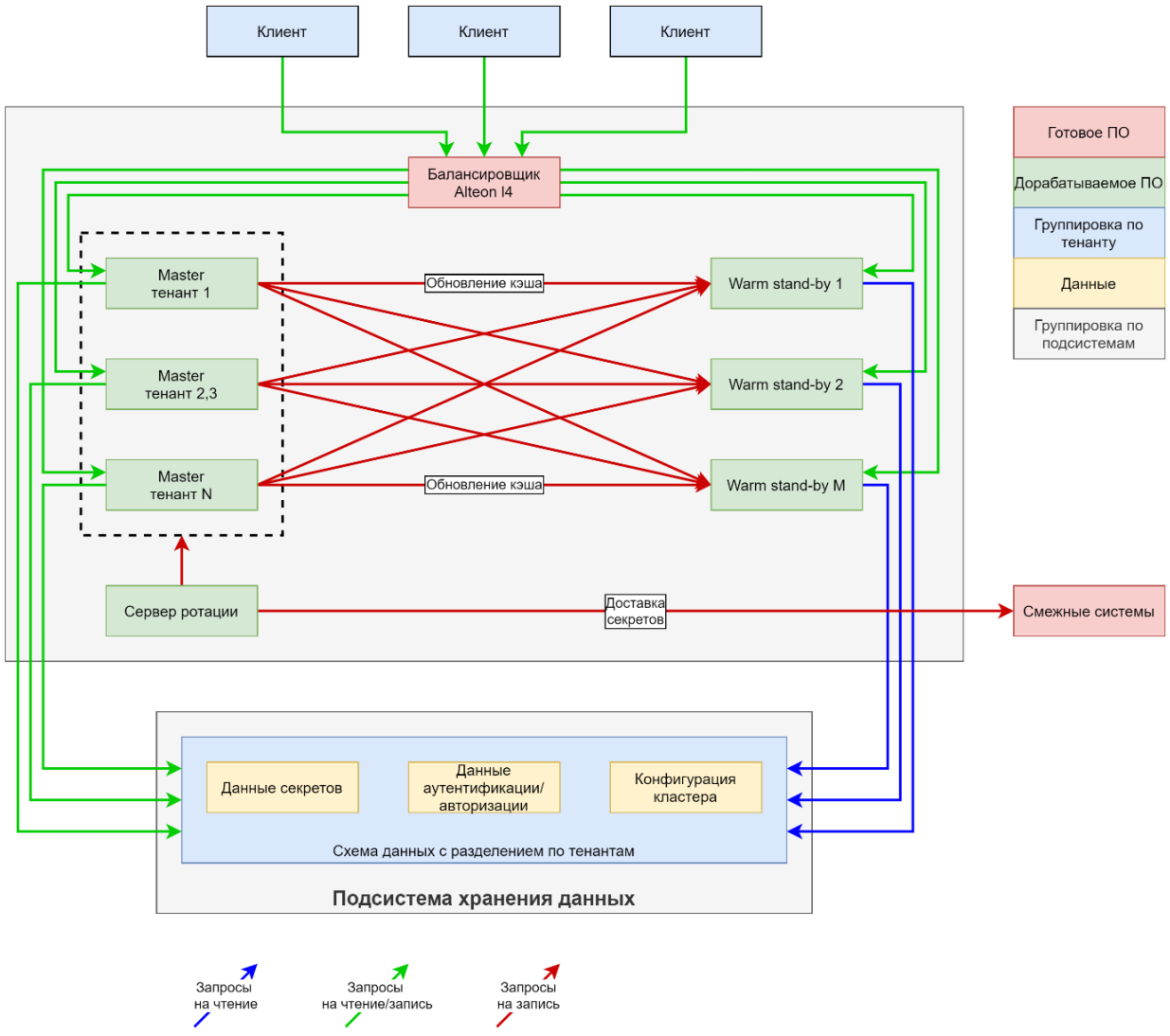
Каждый автономный кластер построен по стандартной схеме для обеспечения высокой доступности, которая обеспечивается резервированием серверов и автоматическим переключением на резервный сервер при наступлении аварии.

Отличия от стандартной схемы обусловлены доработкой с целью обеспечения возможности чтения со Stand-by узлов системы, внедрением распределенного кеширования и поддержкой мультитенантности:

1. Вместо одного Master-сервера, в системе могут быть (а могут не быть) отдельные серверы под каждый тенант.
2. Конфигурация HA-кластера расширяется информацией о тенантах в привязке к обслуживающим их серверам.
3. Балансировщик работает в режиме L4, обеспечивая тем самым обмен по TLS, с использованием аутентификации по сертификату.

4. Запросы распределяются по всем узлам системы вне зависимости, является это запросом на запись или на чтение.
5. Запросы на запись, пришедшие на соответствующий Master-сервер, обрабатываются сразу.
6. Запросы на запись, пришедшие на несоответствующий сервер, форвардятся на соответствующий Master-сервер с помощью уже существующих механизмов.
7. Запросы на запись приводят к инвалидации распределенного кэша.
8. Запросы на чтение обрабатываются всеми узлами.

1.5.Схема кластера



1.5.1. Внедрение распределенного кеширования

Для обеспечения согласованности данных в различных узлах, должна быть доработана подсистема кеширования с внедрением распределенного кэша.

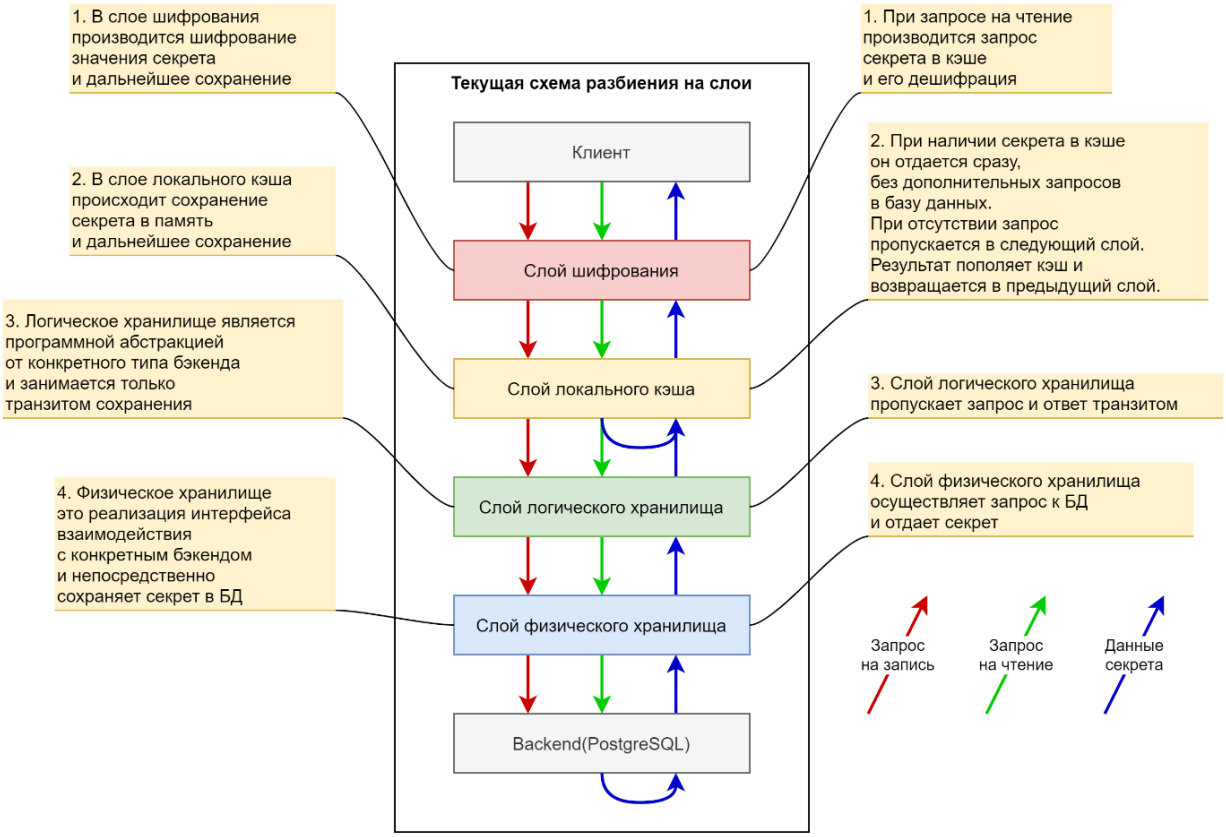
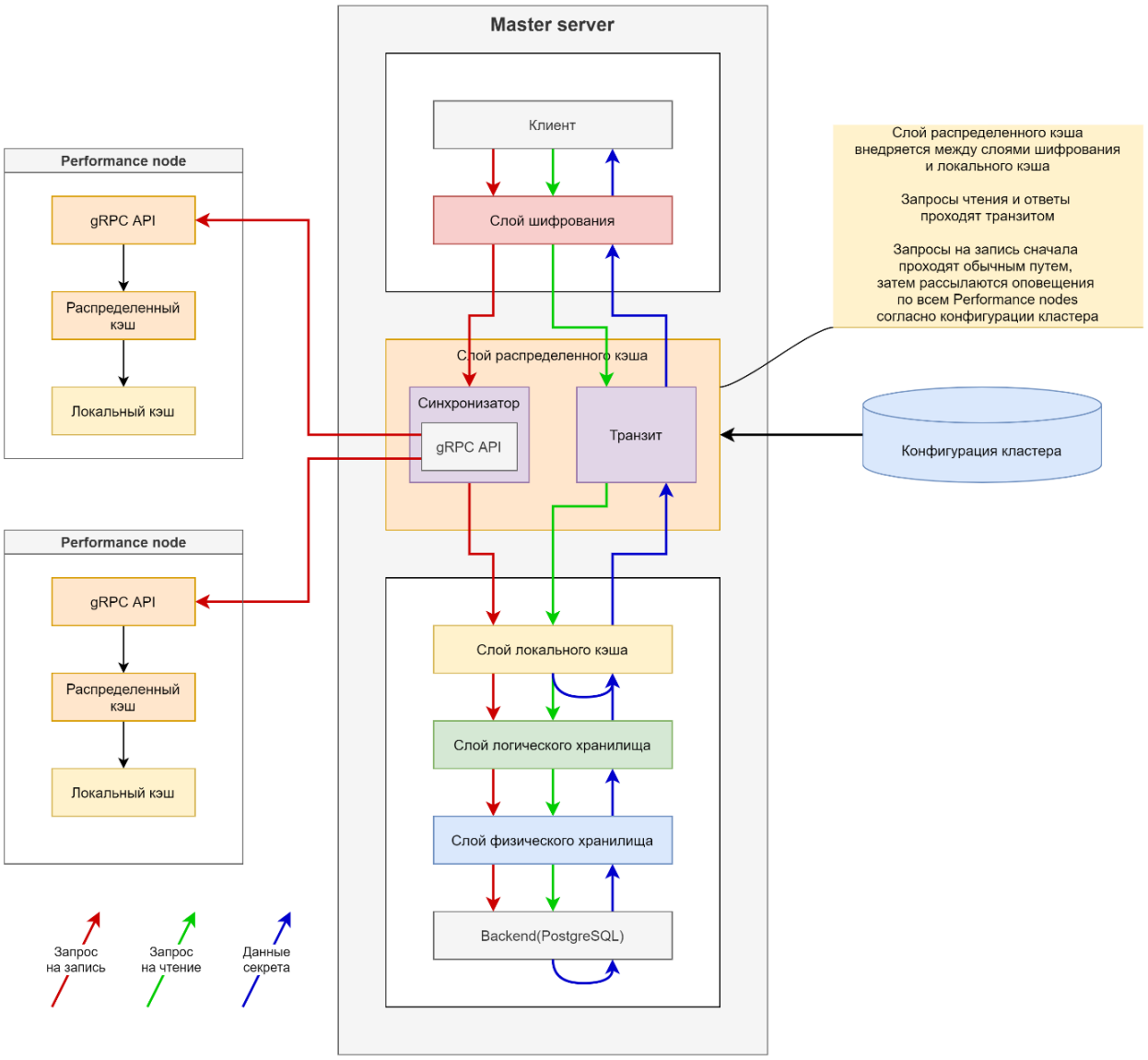


Схема разбиения на слои представляет собой write-through кэш, для которого основная последовательность инвалидации производится в момент записи нового значения секрета. Такая схема обеспечивает когерентность кэша и данных в БД в сочетании с высокой скоростью.

Но при этом текущая реализация не предполагает возможности чтения с нескольких узлов одновременно, поскольку согласование данных производится только в локальном кэше и для согласования нескольких узлов необходимо обеспечить синхронное уведомление всех узлов о необходимости инвалидации значения секрета в кэше при его изменении.

Для облегчения дальнейших обновлений программного кода из общедоступного репозитория, предлагается схема с минимальным изменением текущего кода и отделением новых функций в отдельный программный модуль:

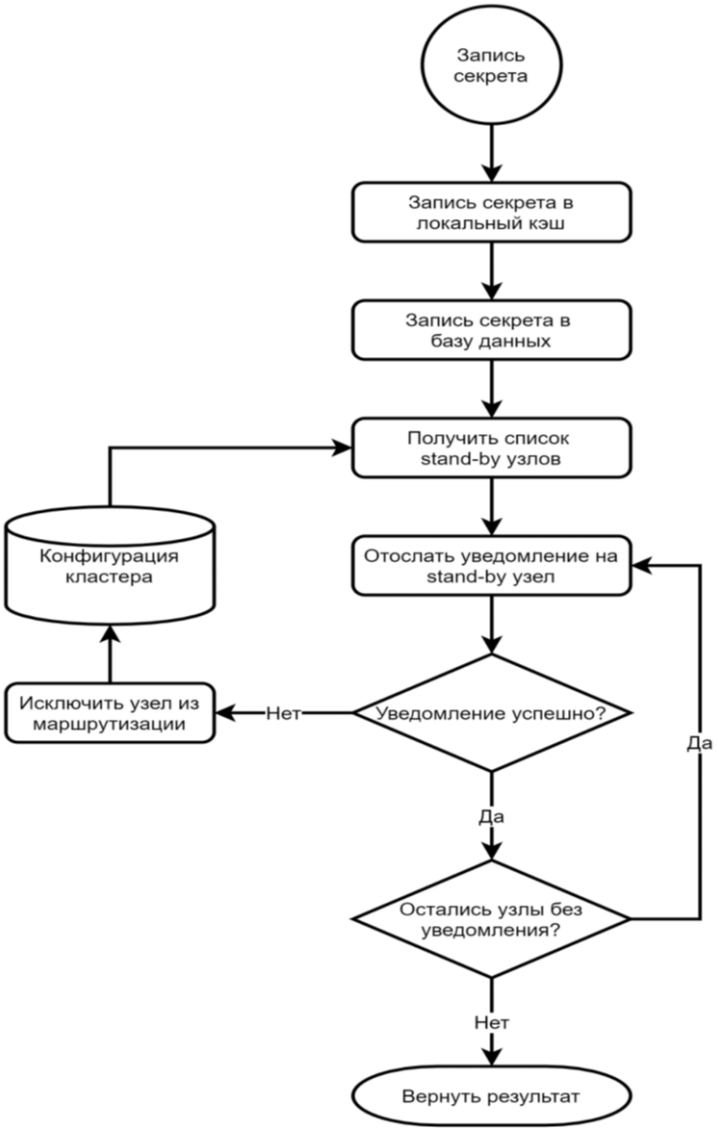


Новый слой, обеспечивающий взаимосвязь узлов между собой, наследуется от общего для всех слоев интерфейса и внедряется между слоем шифрования и слоем локального кэша. Таким образом необходимо небольшое вмешательство в код шифрующего слоя, остальные слои остаются без изменений.

Функции слоя распределенного кэша сводятся к оповещению остальных узлов об необходимости инвалидации кэша. Уведомления должны рассылаться параллельно, таким образом максимальная задержка не будет зависеть от количества оповещаемых узлов, а только от времени ответа от самого медленного узла. Использование протокола gRPC обеспечит максимальную скорость взаимодействия.

Возможные ошибки взаимодействия с некоторыми узлами означают, что узел либо недоступен, либо некогерентен, поэтому такой узел должен быть выведен из состава кластера и таблицы маршрутизации для исключения возможности сбоя запросов или получения устаревших данных.

1.5.2. Алгоритм обработки секретов.



1.5.3. Обеспечение мультитенантности

Подсистема управления секретами логически делится на несколько тенантов. Для отделения тенантов, каждому из них присваивается уникальный человекочитаемый код(namespace). Тенант является практически автономной сущностью за исключением общесистемных функций API. Тенанты содержат внутри себя следующие сущности:

- Secret Engines
- Auth Methods
- Policies
- Identities (Entities, Groups)
- Tokens

Обращения к API, выполняемые в отдельном тенанте, могут быть выполнены путем конкатенации относительного пути запроса с путем к пространству имен, либо с помощью заголовка, либо сочетанием обоих методов. В любом случае система построит полный путь

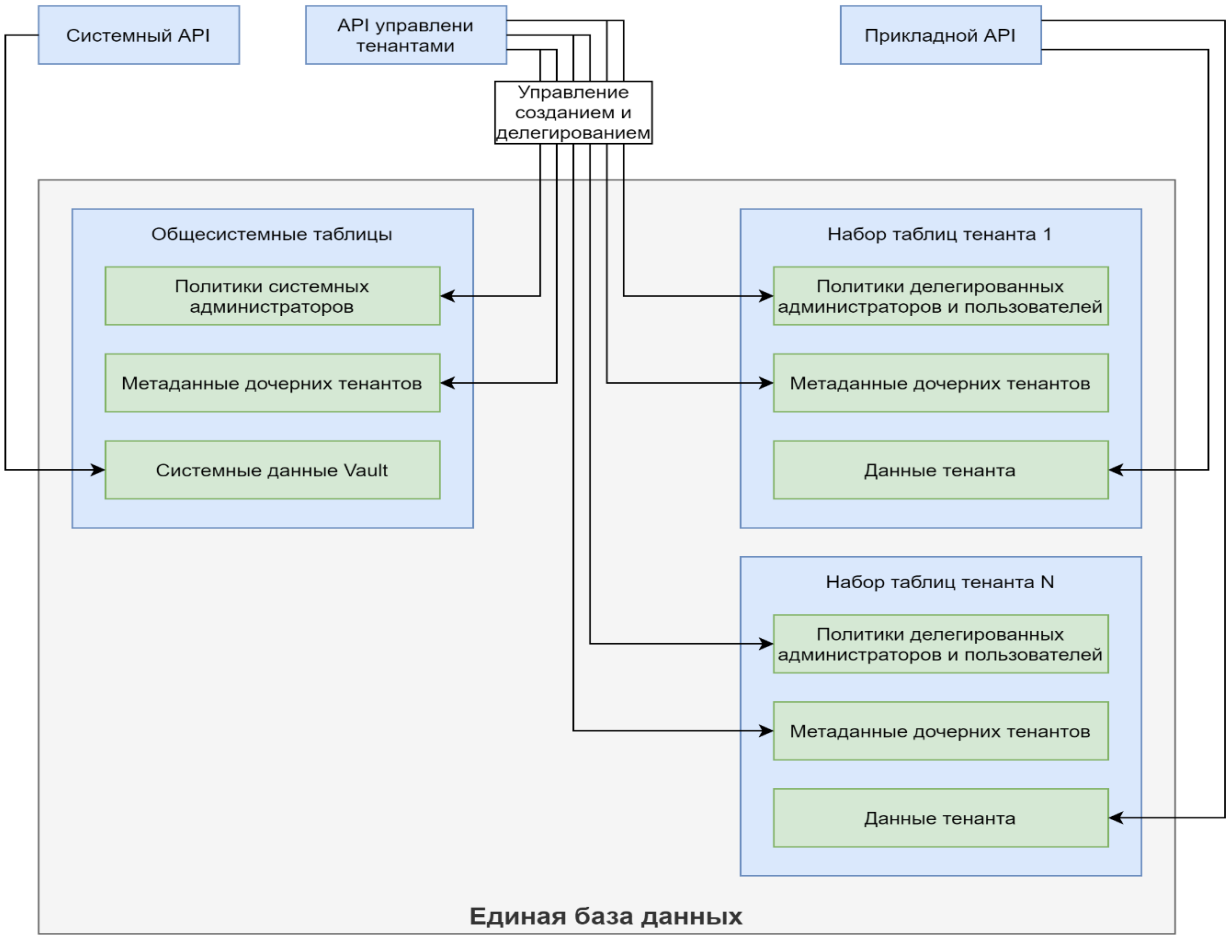
из этих двух источников, чтобы правильно направить запрос в соответствующее пространство имен.

Узлы переправляют запросы в зависимости от кода тенанта. Каждый тенант является изолированным от других тенантов на уровне данных, при этом на один Master-сервер могут приходиться несколько тенантов.

Ролевая модель включает в себя три уровня.

Роль	Полномочия
Системный администратор	Полный спектр полномочий, в том числе доступ к общесистемным функциям и определению политик доступа. Управляет делегированием административных полномочий в тенантах.
Делегированный администратор	Имеет полномочия по управлению политиками внутри тенанта. Может делегировать полномочия на дочерний тенант.
Пользователь	Имеет полномочия в рамках политик, определенных делегированным администратором внутри тенанта.

Схема связей.



Таким образом под каждый тенант создается отдельный набор таблиц, хранящих данные изолированных в тенанте сущностей, политики, действующие внутри тенанта, а также метаданные дочерних тенантов.

Отличие общесистемного набора таблиц от таблиц тенантов состоит в наличии данных общесистемных API и в отсутствии прикладных данных.

1.6.Обработка ошибок

В ходе работы системы возможны аварийные ситуации, во время которых не будет доступа к данным или функциям системы. Отказоустойчивое исполнение предполагает, что недоступность данных или функций системы будет устранена за время, исчисляющееся секундами или десятками секунд. В указанных условиях обеспечение непрерывности работы клиентских систем предполагается реализовать с помощью методики повторения запросов («Retry»).

Авария системы может произойти на нескольких уровнях:

1. При недоступности БД.
2. При недоступности компонента системы.
3. При недоступности Системы в общем.

Логика обработки ошибок распределяется на несколько модулей системы.

1.6.1. Недоступна БД

В случае недоступности базы данных на уровне системы возвращается ошибка, на основании которой генерируется HTTP Response, содержащий заголовок Retry-After с некоторым значением (заданным в настройках), означающим время ожидания в секундах перед повтором запроса.

Получив ответ с указанным заголовком, программный клиент системы из состава SDK, осуществляет повторный запрос через указанное время.

Таким образом, программный комплекс, использующий программный клиент системы из состава SDK, получит доступ к нужной функции системы через некоторый промежуток времени, за который система восстановит свое функционирование.

Если программный комплекс использует REST API системы напрямую, без программного клиента системы, то

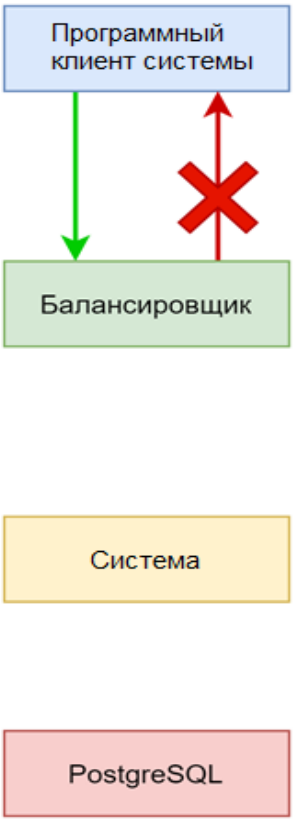
описанное поведение будет аналогичным, если REST клиент поддерживает обработку заголовка Retry-After.

1.6.2. Недоступен компонент системы

В случае недоступности системы, например, при аварийном переключении, возможны две ситуации:

1. Балансировщик передал запрос на Stand-by, но Master-server еще не выбран, Stand-by возвращает HTTP Response, аналогичный ответу из предыдущего пункта. Далее программный клиент системы или REST-клиент действуют аналогично предыдущему пункту.
2. Балансировщику не удалось доставить запрос вообще, в таком случае программный клиент системы действует аналогично предыдущему пункту, но с использованием заданного в настройках времени ожидания и количества попыток.

1.6.3. Недоступна система в общем



При недоступности системы вообще, программный клиент системы повторяет попытки через заданные в настройках промежутки времени и заданное количество раз.

1.7.Подсистема хранения данных

Подсистема хранения данных снаружи должна выглядеть как атомарная подсистема, с одним или несколькими балансировщиками, через которые происходит доступ к данным.

Внутри себя подсистема хранения данных должна быть территориально распределенной по нескольким датацентрам. Каждое развертывание в датацентрах является типовым и состоит из высокого доступного кластера, построенного на основе стека HAProxy, etcd, Patroni, PostgreSQL.

Резервирование данных достигается поточным реплицированием данных. Также это обеспечивает возможность масштабирования на чтение, которое можно осуществлять со Stand-by узлов. Stand-by узлы в схеме играют роль теплого резерва.

1.8. Базовая схема обеспечения высокой доступности