

Астра ИС МД (Инфраструктурные сервисы)

Описание функциональных характеристик программного обеспечения

1 Введение

Документ содержит описание функциональных характеристик программного обеспечения (ПО) Астра ИС МД (Инфраструктурные сервисы), далее АИС МД.

2 Общие сведения о программном обеспечении

Платформа АИС МД является комплексным решением и состоит из АИС МД (средство запуска модулей) и модулей:

Модуль Astra API Gateway в гетерогенной службе каталогов позволяет создать центральную точку доступа для всех внешних и внутренних клиентов, обеспечивая унифицированный, безопасный и управляемый интерфейс для операций над доменными контроллерами Free IPA и SambaDC, что позволяет:

- управлять пользователями, группами, политиками, серверами FreeIPA;
- предоставляет возможность выполнения сложных сценариев (например, создание кросс-доменных связей);
- ограничивать частоту запросов (Rate Limiting, Throttling);
- транслировать вызовы в соответствующие команды API/CLI для FreeIPA и SambaDC;
- принимать и маршрутизировать входящие запросы от пользователей и сервисов;
- полноценно интегрировать FreeIPA и SambaDC.

Модуль Astra DNS Gateway отвечает за авторитативное разрешение и рекурсивное определение, балансировку, защиту и телеметрию службы доменных имен. Обеспечивает бесшовную работу систем с различными контроллерами доменами, что позволяет:

- балансировать нагрузку: запросы LDAP, Kerberos, DNS и др.
- равномерно распределяются между доступными;
- контроллерами/нодами по алгоритмам Round-Robin, Least-Connections, Hashing, для оптимального распределения запросов на основе задержек и загрузки;
- внедрить отказоустойчивость: при недоступности одного из узлов, DNS автоматически исключает его из ответов;
- внедрить защиту и фильтрацию: блокирует DDoS-атаку, DNStуннелирование, эксфильтрацию, псевдослучайные поддомены.

Модуль Astra Redis — это высокопроизводительная in-memory база данных типа ключ-значение, которая также может использоваться как кэш, брокер сообщений и хранилище структур данных. Модуль выполнять периодические снимки базы данных, логирование всех операций записи и комбинированный подход для максимальной надежности. Асинхронная репликация с master на множество реплик. Высокая доступность, масштабирование чтения, бекапирование.

Модуль Astra Scylla - распределенная Система Управления Базами Данных с Модульной Архитектурой «Astra Scylla» используется в банковской сфере для обеспечения работоспособности высоко нагруженных банковских сервисов и выполняет следующие функции: обеспечение сверхнизкого времени отклика (единицы миллисекунд) для транзакционных систем в реальном времени, горизонтальное масштабирование для поддержки пиковых нагрузок (например,

в период массовых списаний или платежных шлюзов), а также отказоустойчивость и гарантию целостности данных за счет репликации без единой точки отказа.

Модуль Astra ArangoDB

Astra ArangoDB — это мультимодельная NoSQL база данных с открытым исходным кодом, предоставляющая возможности хранения данных в документо-ориентированном, графовом и ключ-значение форматах. Она предлагает гибкость схемы и поддерживает язык запросов AQL (ArangoDB Query Language), который позволяет выполнять сложные запросы к данным.

Модуль Astra Apache Airflow

Программное обеспечение для создания, выполнения, мониторинга и оркестровки потоков операций «Astra Apache Airflow». содержит следующие модули:

- планировщик (Scheduler)
- executor (Исполнитель)
- база метаданных
- веб-сервер
- работники (Workers)
- триггер (Trigger)

1. Планировщик (Scheduler) — читает расписание каждого DAG и определяет, какие задачи должны быть запущены и когда. Например, планировщик может определить, что ежедневная задача обработки данных должна быть запущена в 11:00 каждый день.
2. Executor (Исполнитель) — компонент, который определяет, как именно задачи будут выполняться. Airflow предоставляет несколько типов исполнителей, которые могут работать в различных средах и конфигурациях.
3. База метаданных — хранит описательную информацию, а не фактическую. Данный тип баз используют для управления, организации и поиска данных. База метаданных в AirflowAA хранит информацию о задачах, их статусе, зависимостях и истории выполнения.
4. Веб-сервер — предоставляет пользовательский интерфейс для мониторинга, управления и запуска задач. Через веб-интерфейс пользователи могут просматривать список задач, проверять их статус и управлять расписанием выполнения.
5. Работники (Workers) — берут на себя выполнение заданий, распределяемых исполнителем.
6. Триггер (Trigger) — механизм, который позволяет запустить процесс при наступлении определённого события или условия, не обязательно связанного с расписанием.

Модуль Astra ClickHouse

Astra ClickHouse — это популярная OLAP СУБД с открытым исходным кодом, предназначенная для высокопроизводительной аналитики наборов данных петабайтного масштаба с интенсивными вставками данных. Ее система хранения комбинирует формат данных на основе

традиционных LSM-деревьев (log-structured merge trees) с современными техниками с постоянной трансформацией (т.е. агрегацией, архивированием) исторических данных в фоновом режиме. Запросы пишутся на удобном диалекте языка SQL и обрабатываются векторным ядром выполнения запросом с опциональной компиляцией кода. Astra ClickHouse выполняет агрессивные техники исключения обработки ненужных для результата данных. Остальные СУБД могут быть интегрированы на уровне табличных функций, табличных или уровня базы ядер обработки (table или database engine). Бенчмарки на реальных данных демонстрируют, что Astra ClickHouse является одной из самых быстрых аналитических СУБД на рынке.

Astra ClickHouse предназначена для решения пяти ключевых вызовов при обработке современных аналитических данных:

- Огромные наборы данных с высокой скоростью вставки данных
- Много параллельных запросом с ожиданием низких задержек от них
- Разнообразный ландшафт хранилищ данных, их расположение и форматы
- Удобный язык запросов с возможностью анализа их производительности
- Надёжность и разнообразное развёртывание

Модуль Astra Секреты

Модуль Astra Секреты является комплексным решением для централизованного управления секретами и состоит из следующих основных компонентов:

Ядро системы — сервер, обеспечивающий хранение, шифрование и распределение секретов с использованием алгоритма AES-256-GCM. Ядро реализует механизмы Seal/Unseal для защиты данных при перезагрузке, консенсусный протокол Raft для обеспечения высокой доступности и горизонтального масштабирования кластера.

Подсистема аутентификации — обеспечивает проверку подлинности пользователей и приложений посредством поддержки 10 методов аутентификации: Token, LDAP, OIDC/JWT, Kubernetes, AppRole, Cert (сертификаты X.509), Userpass, Kerberos, RADIUS и многофакторная аутентификация (MFA).

Подсистема секретов (Secrets Engines) — предоставляет механизмы генерации, хранения и управления секретами различных типов: статические секреты (KV v1/v2), динамические учётные данные баз данных, сертификаты PKI, шифрование как услуга (Transit), SSH-ключи, TOTP-коды, а также интеграции с Kubernetes, OpenLDAP и RabbitMQ.

Подсистема политик доступа — реализует разграничение прав доступа на основе ACL-политик (формат HCL/JSON) и CEL Policy Engine (Custom Expression Language), аналогичного HashiCorp Sentinel. Поддерживается ролевая модель управления доступом (RBAC) и механизм квот для ограничения частоты запросов.

Подсистема пространств имён (Namespaces) — обеспечивает логическую изоляцию среды между командами, проектами и подразделениями организации. Каждый namespace может иметь собственные политики, методы аутентификации и секреты.

Подсистема аудита — регистрирует все операции с секретами и записывает их в журналы аудита. Поддерживаются типы: File (файловый журнал), Syslog, Socket (TCP/UDP) и HTTP

(отправка на внешний сервер). Потоки аудита могут направляться в SIEM-системы для мониторинга.

Подсистема автоматического распечатывания (Auto-Unseal) — обеспечивает автоматическое распечатывание сервера при перезапуске с использованием внешних модулей безопасности: PKCS#11 (HSM), KMIP, а также облачных KMS (AWS KMS, Azure Key Vault, GCP KMS).

3 Область применения

Программное обеспечение предназначено для централизованного управления IT-инфраструктурой на базе операционной системы Astra Linux, автоматизации рутинных задач системного администрирования, управления СУБД и обеспечения безопасности. Используется организациями любого масштаба для управления компьютерами, пользователями, групповыми политиками, установки операционных систем и программного обеспечения по сети, а также для удаленного доступа и мониторинга, управление базами данных.

Областью применения настоящего программного обеспечения являются любые сферы государственной или частной деятельности, автоматизирующие свою деятельность (использующие программное обеспечение при ведении деятельности).

4 Языки программирования

Исходный код АИС МД написан на следующих языках:

- Java;
- Javascript (ReactJS framework);
- C++;
- Python;
- Rust;
- TypeScript react;
- SQL.

5 Общее описание функциональных характеристик

Модуль Astra Балансировщик

- Корректная работа сервисов служб каталога (LDAP, KDC, NTP, DNS, RPC и др.)
- Развертывание и масштабирование комплексных информационных структур на основе продуктов ГК Astra и партнеров, демонстрационный вариант которых был представлен здесь. Развертывание и масштабирование возможно в облачных окружениях, частных центрах обработки данных и их сочетаниях (гибридных окружениях).
- Настройка сетевого оборудования, как зарубежного, так и отечественного.
- Автоматизация применения мер безопасности.

Модуль Astra DNS Gateway

- Позволяет масштабировать инфраструктуру под планируемые задачи.
- Средства внутренней самодиагностики позволяют оперативно определять и устранять проблемы в сопряженных сервисах банковской инфраструктуры.
- Аналитический модуль позволяет отслеживать состояние и параметры системы в режиме реального времени, строить отчёты в различных разрезах данных. Осуществляет мониторинг и управление базами данных с модульной архитектурой.

Модуль Astra Apache AirFlow

Программное обеспечение, модуль «Astra Apache Airflow» обладает следующими основными функциями:

- Оркестрация данных — система позволяет определить сложные рабочие процессы как направленные ациклические графы (DAG), где каждая задача имеет чётко определённые зависимости от других. Это обеспечивает правильный порядок выполнения операций и предотв

Модуль Astra ArangoDB

Astra ArangoDB – мультимодельная БД с возможностью хранения данных в форме графов, документов и ключ-значений. У Astra ArangoDB высокая скорость работы, производительность, низкие затраты памяти.

Astra ArangoDB поддерживает масштабирующиеся запросы к графовым данным. Все данные в Astra ArangoDB во внешнем представлении выглядят как JSON и аналогично структурированные документы, которые могут быть объединены в коллекции — подобие таблицы в реляционных базах данных.

Astra ArangoDB хранит объекты в json документах с поддержкой обращения по ключу, за которое отвечает специальное поле `_key`.

Кластер Astra ArangoDB состоит из нескольких экземпляров ArangoDB, которые взаимодействуют друг с другом по сети. Они играют разные роли

Базовые (основные) модули

Агенты

Один или несколько агентов формируют Агентство в Кластере ArangoDB. Агентство — это центральное место хранения конфигурации Кластера. Оно выполняет выборы лидера и предоставляет остальные сервисы синхронизации для всего Кластера. Остальные компоненты не могут функционировать без Агентства.

Для Агентства в промышленных системах обязательно должна обеспечиваться отказоустойчивость. Для обеспечения этого используется консенсусный алгоритм Raft. Этот алгоритм формально гарантирует отсутствие конфликтов при управлении конфигурацией Кластера ArangoDB.

Координаторы

Координаторы должны быть доступны извне. Работа клиентов осуществляется через них

- **Высокая производительность и масштабируемость** — обусловлена форматом хранения данных в виде столбцов, асинхронной мульти-мастер репликацией и распределенной обработкой запросов. Эти особенности позволяют Astra ClickHouse обрабатывать большие объемы данных и выполнять несколько запросов одновременно, не снижая производительность.
- **Поддержка SQL** — поддержка расширенного SQL способствует легкому освоению ClickHouse новыми пользователями.
- **Поддержка сложных запросов** — используя язык SQL, СУБД предоставляет возможность пользователям писать сложные запросы для создания кастомных отчетов. Это включает в себя агрегацию данных, группировку, оконные функции, подзапросы и многие другие способы для проведения комплексного анализа данных. Поддержка вложенных структур данных — внутри одной ячейки можно хранить дополнительные таблицы.
- **Обработка данных в реальном времени**

мириться с некоторой степенью устаревания данных. Другими примерами специализированных Табличных Ядер являются Ядра обработки данных только в памяти, используемых для обработки временных данных, и Распределённые (Distributed) Табличные Ядра для горизонтального масштабирования.

- **Третья категория** — это виртуальные Табличные Ядра для двустороннего обмена данных с внешними системами, такими как реляционные СУБД (например, PostgreSQL, MySQL), системы подписок/публикаций (например, Kafka, RabbitMQ) или хранилища типа ключ/значение (например, Redis). Виртуальные Ядра могут также взаимодействовать с озёрами данных (data lakes, например, Iceberg, DeltaLake, Hudi) или файлами в объектном хранилище (например, AWS S3, Google GCP).

LDAP — аутентификация через корпоративный каталог LDAP/Active Directory. Поддерживается маппинг групп LDAP на политики Астра Секреты, что позволяет автоматически назначать права доступа на основе членства в группах.

OIDC/JWT — аутентификация через OpenID Connect провайдеров и JWT-токены. Поддерживает интеграцию с Keycloak, Astra Identity и другими OIDC-совместимыми IdP.

Kubernetes — аутентификация под (pods) Kubernetes через Service Account Token. Позволяет приложениям в кластере автоматически получать доступ к секретам без явной передачи учётных данных.

AppRole — механизм аутентификации машин и приложений через пару Role ID / Secret ID. Рекомендуемый метод для CI/CD систем и автоматизированных процессов.

PKI (Public Key Infrastructure) — управление сертификатами X.509. Полный цикл: создание корневых и промежуточных CA, выпуск сертификатов, проверка CRL/OCSP, настройка ролей для автоматической генерации сертификатов.

Transit — шифрование как услуга (Encryption-as-a-Service). Предоставляет API для шифрования и расшифрования данных без их хранения. Поддерживает алгоритмы AES-256-GCM, RSA, ChaCha20-Poly1305, Ed25519. Используется для шифрования данных приложений без предоставления доступа к ключам шифрования.

SSH — управление SSH-ключами и одноразовыми паролями (OTP). Поддерживает два режима: генерация кратковременных SSH-ключей и выдача одноразовых паролей для доступа к серверам. Обеспечивает аудит всех SSH-соединений.

RBAC (Role-Based Access Control) — ролевая модель управления доступом. Позволяет назначать политики ролям и привязывать роли к пользователям или группам.

Quotas (квоты) — ограничение частоты запросов (rate limiting) на уровне namespace, метода аутентификации или конкретного пути. Предотвращает злоупотребление ресурсами и защищает от DDoS-атак.

Пространства имён (Namespaces)

ПО Астра Секреты поддерживает логическую изоляцию среды через пространства имён. Каждый namespace представляет собой изолированную среду с собственными:

- Секретами (Secrets Engines)
- Методами аутентификации (Auth Methods)
- Политиками доступа (Policies)
- Пользователями и токенами
- Конфигурациями аудита

Namespaces поддерживают иерархическую структуру — пространства имён могут быть вложенными. Это позволяет моделировать организационную структуру предприятия: департамент → команда → проект.

Root namespace содержит глобальные конфигурации и имеет доступ ко всем дочерним пространствам. Администраторы namespace могут управлять только ресурсами в пределах своего пространства.

Кластер высокой доступности (HA Cluster)

ПО Астра Секреты поддерживает развёртывание в режиме кластера высокой доступности (HA) с использованием протокола консенсуса Raft.

- Протокол консенсуса Raft — обеспечивает согласованность данных между узлами кластера
- Автоматическое переключение (failover) — при отказе лидера кластера автоматически выбирается новый лидер
- Горизонтальное масштабирование — добавление новых узлов без остановки сервиса
- Репликация данных — все данные реплицируются между узлами кластера
- Рекомендуемая конфигурация — 3 или 5 узлов для обеспечения кворума
- Интегрированное хранилище (Integrated Storage) — встроенное хранилище на основе Raft, не требующее внешней СУБД

Кластер поддерживает режим Multi-Peer — подключение дополнительных узлов для распределения нагрузки и повышения отказоустойчивости.

Хранилище данных (Storage Backend)

ПО Астра Секреты поддерживает несколько типов хранилищ для персистентного хранения зашифрованных данных:

Raft (Integrated Storage) — встроенное хранилище, рекомендуемое для большинства развёртываний. Обеспечивает высокую доступность, репликацию и консенсус данных без необходимости во внешних компонентах.

PostgreSQL — хранение данных во внешней СУБД PostgreSQL. Используется в случаях, когда требуется интеграция с существующей инфраструктурой СУБД. Поддерживает транзакции и обеспечивает надёжность хранения.

Все данные в хранилище хранятся в зашифрованном виде. Выбор хранилища не влияет на безопасность данных, так как шифрование выполняется на уровне ядра системы.

Подсистема аудита

ПО Астра Секреты обеспечивает полный аудит всех операций с секретами. Каждое действие (чтение, запись, удаление, аутентификация) регистрируется и может быть отправлено во внешние системы мониторинга.

File — запись аудита в файл на диске сервера. Поддерживает ротацию файлов и настройку формата (JSON).

Syslog — отправка записей аудита в системный журнал syslog. Интеграция с стандартными средствами логирования ОС.

Socket — отправка аудита по сети через TCP или UDP. Позволяет направлять поток аудита на удалённый сервер сбора логов.

HTTP — отправка записей аудита по HTTP/HTTPS на внешний сервер. Используется для интеграции с SIEM-системами (Security Information and Event Management).

- Все записи аудита содержат: временную метку, тип запроса, путь, идентификатор токена, результат операции
- Чувствительные данные (значения секретов) в записях аудита хешируются с использованием HMAC-SHA256
- Поддержка нескольких одновременных устройств аудита

Интерфейсы управления

REST API — основной программный интерфейс управления. Доступен на порту 8200 (HTTP/HTTPS). Все операции с секретами, конфигурациями и политиками выполняются через REST API. Аутентификация через заголовок X-Vault-Token. Поддержка Swagger/OpenAPI документации.

CLI (Command Line Interface) — командная строка `bao` для управления секретами. Предоставляет полный набор команд для работы с секретами, политиками, методами аутентификации, кластером и аудитом. Поддерживает автодополнение команд.

Web UI — веб-интерфейс для визуального управления секретами. Позволяет просматривать и редактировать секреты, управлять политиками, методами аутентификации и мониторить состояние кластера.

Интеграции

Модуль Astra Секреты обеспечивает интеграцию с широким спектром систем и технологий:

- Astra Linux Special Edition — сертифицированная ОС для развёртывания
- FreeIPA / Active Directory — аутентификация через LDAP и Kerberos
- Keycloak / Astra Identity — аутентификация через OIDC/SAML
- Kubernetes / Red Hat OpenShift — аутентификация под и генерация токенов
- PostgreSQL / MySQL / Oracle / MongoDB — динамические учётные данные
- HSM (Hardware Security Module) — Auto-Unseal через PKCS#11 и KMIP
- SIEM-системы — интеграция через HTTP-аудит и syslog
- Docker — контейнерное развёртывание
- CI/CD системы (GitLab CI, Jenkins) — аутентификация через AppRole

6 6. Используемые технические средства

6.1 Аппаратные требования

6.1.1 Требования к серверу для Астра ИС МД

Процессор:

Минимальные: 2 ГГц, 2 ядра

Рекомендуемые: 4 ГГц, 4 ядра

Оперативная память:

Минимальные: 4 Гб

Рекомендуемые: 8 Гб

Дисковое пространство:

Минимальные: 10 Гб (предпочтительнее SSD)

Рекомендуемые: 50 Гб (предпочтительнее SSD)

6.1.2 Требования к АРМ:

Процессор:

Минимальные: 2,4 ГГц, 2 ядра

Рекомендуемые: 4 ГГц, 4 ядра

Оперативная память:

- Минимальные: 8 Гб
- Рекомендуемые: 16 Гб

Дисковое пространство:

Минимальные: 50 Гб (SSD)

Рекомендуемые: 100 Гб (SSD)

Монитор:

Диагональ: не менее 16"

Разрешение экрана: не менее 1024*768

Скорость подключения к сети: не менее 2 Мбит/с.

6.2 Программные требования

6.2.1 Операционные системы:

Операционная система Linux (Ubuntu 22.04 / Debian 12 / Astra Linux)

Python версии 3.10 или 3.11

Пакеты python3-venv, python3-dev, build-essential

Рекомендованная ОС:

Astra Linux Special Edition — перед установкой выполните инструкции из статьи Подготовка сервера с ОС Astra Linux.

Использование на других платформах может потребовать локальных изменений.

База данных:

Astra Apache Airflow требует базу данных для хранения своих данных. Рекомендуется использовать PostgreSQL.

АРМ:

Для работы с системой требуется наличие одного из браузеров:

Google Chrome, версия не ниже 109

Яндекс.Браузер, версия не ниже 23.7.2.767 (64-bit)

Mozilla FireFox, версия не ниже 102

Примечание: Astra Apache Airflow может также использовать дополнительные ресурсы для выполнения задач и оркестрации процессов. Ресурсы будут зависеть от объема задач и рабочих процессов, которые необходимо автоматизировать.

7 Входные и выходные данные

Модули Astra Балансировщик и Astra DNS Gateway

Входными данными являются параметры конфигурации серверов, на которых будет реализована автоматизация.

Выходными данными могут быть:

- отчеты о завершении исполнения заданий; – состояние системы.

Модуль Astra Redis

Входные данные

- Команда и её аргументы (например, GET user:1000)
- Все данные передаются в виде текстовых строк или бинарных данных

Выходные данные

Результат выполнения команды в одном из базовых форматов:

- ОК — команда выполнена успешно (например, после SET)
- Значение — строка, число или коллекция (например, после GET или INCR)
- nil — запрашиваемый ключ не найден
- Сообщение об ошибке — если команда не может быть выполнена

Модуль Astra Scylla

Входные данные: операторы INSERT / UPDATE на языке CQL, содержащие первичный ключ (обязательно) и значения столбцов (любые типы: строки, числа, коллекции и т.д.).

Выходные данные: набор строк (ResultSet) с запрошенными колонками, возвращаемый по SELECT с обязательным указанием ключа раздела в условиях WHERE.

Модуль Astra ArangoDB

Входные данные

Входными данными являются данные представлены в формате JSON-документов. В документе содержится ноль или более атрибутов, и каждому атрибуту присваивается значение. Значение может иметь атомарный тип (число, логическое значение и т. д.) или составной тип (встроенный документ, объект или массив).

Выходные данные

Выходными данными могут быть:

- Для графов — результаты запросов, которые могут включать списки всех вершин и всех рёбер между запрошенными вершинами.
- Для документов — результаты запросов, которые могут включать объект по ключу (например, RETURN DOCUMENT(_id)) или список объектов из коллекции по ключам (RETURN DOCUMENT(collection_name, [_key, ...])).

- Для результатов аналитики — например, результаты алгоритмов, которые агрегируют информацию о графе (connected components, label propagation и др.). В отличие от запросов к графу, аналитика включает весь граф сразу.

Также в результатах запросов к документам могут быть указаны версии документов (document revision) — токены MVCC, которые указывают на конкретную версию документа (identified by its `_id`). Это строковое значение, содержащее целое число, и уникально в списке версий для одного документа.

Модуль Astra Apache AirFlow Входными данными являются параметры конфигурации серверов, на которых будет реализована автоматизация.

Выходными данными могут быть:

- отчеты о завершении исполнения заданий;
- состояние системы.

Модуль Astra ClickHouse

Входными данными являются:

- Текстовые — CSV, TSV, JSON;
- двоичные колоночно-ориентированные — Native, Parquet, Arrow;
- двоичные строко-ориентированные — RowBinary, Protobuf, Avro.

Выходными данными могут быть:

- представления результатов запроса SELECT;
- выполнения операций INSERT в таблицы с файловой поддержкой.

Модуль Astra Секреты

REST API

Входными данными являются HTTP-запросы к REST API сервера Астра Секреты. Запросы передаются в формате JSON. Аутентификация выполняется через заголовок X-Vault-Token или через тело запроса (в зависимости от метода аутентификации).

Примеры входных данных REST API:

- Создание секрета: PUT `/v1/secret/data/{path}` с JSON-телом, содержащим пары ключ-значение
- Чтение секрета: GET `/v1/secret/data/{path}`
- Аутентификация: POST `/v1/auth/{method}/login` с учётными данными
- Создание политики: PUT `/v1/sys/policies/acl/{name}` с HCL-политикой

- Генерация динамических учётных данных: GET /v1/{engine}/creds/{role}

Выходными данными REST API являются HTTP-ответы в формате JSON. Каждый ответ содержит поля request_id, lease_id, renewable, lease_duration, data.

Примеры выходных данных REST API:

- Секрет: JSON-объект с полями data (содержимое секрета), metadata (версия, время создания)
- Токен аутентификации: JSON с полями client_token, accessor, policies, token_type, lease_duration
- Динамические учётные данные: JSON с полями username, password, TTL
- Статус кластера: JSON с полями sealed, standby, leader_address, raft_committed_index

HCL-политики

Входными данными являются файлы политик в формате HCL:

- path — путь к секрету или группе секретов (с поддержкой шаблонов)
- capabilities — список разрешённых операций (create, read, update, delete, list, sudo)
- parameters — ограничения на значения параметров (для CEL Policy Engine)

Выходными данными являются подтверждения применения политики и сообщения об ошибках валидации (синтаксические ошибки, нарушения ограничений).

Аудит

Выходными данными подсистемы аудита являются записи журнала аудита в формате JSON, содержащие:

- type — тип запроса (request/response)
- auth — информация об аутентификации (accessor, token_type, policies)
- request — детали запроса (operation, path, data, wrap_info)
- response — детали ответа (secret, data, wrap_info)
- error — информация об ошибке (если возникла)

8 Подготовка к работе

Состав работ по подготовке АИС МД к использованию изложен в Инструкции по установке и настройке, включая описание параметров конфигурационных файлов.