

СИСТЕМА УСЛОВНОГО ДОСТУПА «CAS RSCC»

Руководство по установке

2025

1. ОПИСАНИЕ КОМПОНЕНТОВ СИСТЕМЫ УСЛОВНОГО ДОСТУПА «CAS RSCC»

Настоящий документ устанавливает общие положения установки, применимые ко всем компонентам системы условного доступа «CAS RSCC».

Архитектура системы основана на микросервисном принципе построения.

В таблице №1 ниже указывается какие модули активируются в соответствующем компоненте в зависимости от технических условий эксплуатации и задач системы.

Реализация функциональных возможностей компонентов обеспечивается посредством взаимодействия включённых модулей в рамках заданной конфигурации.

Компонент	Назначение компонента	Описание компонента
Billing	Подсистема учета подписок и тарификации. Контроль и управление правами доступа абонентов к пакетам платных услуг, обработке сообщений от платежных систем, автоматического выставления счетов для оплаты.	Микросервис cas-mdb, включающий следующие модули: invoices - управление счетами; subscriptions - управление подписками; payment_orders - управление платежами; tariffs - управление тарифами; special_offers - управление специальными предложениями. Микросервис cas-crm - управление интеграциями (Bitrix24, ГПКС CRM).
CAS	Подсистема условного доступа. Компонент позволяет объединять все задействованные в работе компоненты, производить обработку биллинга, кодирования, настройку пользователей B2B и/или B2C. Компонент также обеспечивает взаимосвязь между серверной частью и устройствами управления потребителей.	Микросервис cas-mdb, включающий следующие модули: hardware - управление оборудованием; cam-modul - управление САМ-модулями; client-hardware - управление пользовательским оборудованием; client - управление клиентами (конечными пользователями); monitoring - управление мониторингом оборудования; procedures - управление автоматизацией; udp_streams - управление потоками; channels - управление каналами тарифов. Микросервис cas-ssm, включающий следующие модули: users - управление пользователями; roles - управление ролями пользователей; self_user - управление текущим пользователем; tokens - управление авторизацией и аутентификацией; user_notification - управление

		уведомлениями пользователей; integration_access_token - управление токенами интеграции. Микросервис cas-stream-analyzer, включающий следующие модули: stream-stats - мониторинг кодирующего оборудования, ошибок, битрейт; channels — мониторинг потока, управление кодированием. Микросервис cas-snmp-worker - мониторинг оборудования по протоколу SNMP.
EMM	Генерация управляющих сообщений (EMM).	Микросервис cas-mdb - модуль emm - формирование сообщений emm. Микросервис cas-emm - модуль emmg - получение сообщений emm от других модулей, формирование ts пакетов.
BD	База данных и механизм передачи сообщений между модулями.	Механизм передачи сообщений между модулями реализован через брокера сообщений Kafka, или REST API. Каждый указанный микросервис представлен собственным независимым набором баз данных: cas_mdb - Redis, Postgres, Mongo; cas_crm - Redis, Postgres; cas_ssm - Redis, Postgres; cas_stream_data_analyzer - Redis, Postgres, Clickhouse; cas_reports - Redis, Postgres, Mongo.
SMS	Компонент, отвечающий за управление информацией о потребителях, подписках по доступным услугам.	Микросервис cas-mdb, включающий следующие модули: subscriptions - управление подписками; emm — генерация emm сообщений.
Info	Модуль формирования отчетов и доставки информационных сообщений.	Микросервис Cas-reports - формирование отчетов. Микросервис cas-email - микросервис отправки уведомлений по протоколу SMTP. Микросервис cas-mdb - модуль notification - управление уведомлениями клиентов.
Encryptor	Кодирование транспортного потока вне зависимости от количества PID каналов внутри.	Encryptor – кодирование транспортного потока; ЕСМ – генерация сообщений для управления кодированием;

	Защита от подмены потока, а также его контроль на всех этапах передачи.	DRM – управление лицензиями и правами доступа к контенту.
Decryptor	Проверка целостности, защиты и соответствия с исходным кодированным потоком до его передачи на следующее устройство приема и/или передачи потока.	Decryptor – декодирование транспортного потока, с проверкой оригинальности кодирования, наличия данных от Encrptor.
Прошивка САМ-модуля	Абонентский модуль доступа.	Адаптированная версия компонента Decryptor для САМ-модуля. Устанавливается на совместимый САМ-модуль. Обеспечивает безопасное дескремблирование входящего потока после получения разрешающих сообщений от ядра системы.
PCI-e modulator	Аппаратный модуль вывода.	Преобразует декодированный видеопоток в формат, пригодный для финальной трансляции на телевизионные устройства.

Таблица №1

2. ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Требования к аппаратному обеспечению

Минимальные технические требования к серверному оборудованию для развертывания системы в кластере Kubernetes v1.32 и для периферийных компонентов.

Компонент	Кол-во	vCPU	RAM (GB)	Disk (GB)	Дополнительно
K8s worker	3	12	16	200	SSD
K8s master	2	4	4	32	-
Encryptor/Decryptor	-	16	16	200	Edge-серверы CPU E5-2630 v2, 16 GB RAM, 2x HDD 500 Gb.

Требования к программному обеспечению

- Операционная система: Linux (рекомендуется Ubuntu или Astra Linux).
- Контейнеризация: Docker.
- Оркестрация: Kubernetes v1.32.
- Базы данных и брокеры сообщений: PostgreSQL, MongoDB, ClickHouse, Redis, Kafka.
- Docker Registry: для хранения и загрузки образов CAS.tar.
- Убедитесь в наличии YAML-файлов для каждого микросервиса с настройками apiVersion: apps/v1, kind: Deployment, selectors и labels.

Подготовка к Установке

- Установка ОС и Docker: установите Linux и Docker на все серверы кластера. Обеспечьте root-доступ (sudo -s).
- Настройка Kubernetes: Разверните кластер Kubernetes (3 worker, 2 master).
- Загрузка образов: загрузите архив CAS.tar в локальный Docker Registry, чтобы сделать образы доступными для кластера.
- Проверка YAML: для каждого сервиса выполните dry-run: kubectl apply -f <yaml-file> --dry-run=client для валидации перед применением.

3. УСТАНОВКА КОМПОНЕНТОВ СИСТЕМЫ УСЛОВНОГО ДОСТУПА «CAS RSCC»

Настоящий раздел описывает установку каждого компонента системы.

1. Компонент "CAS" (Подсистема условного доступа)

Назначение: Обработка информации о доступе к услугам и обмен ключами.

Включает микросервисы и модули:

- cas-mdb: модули hardware, cam-modul, client-hardware, client, monitoring, procedures, udp_streams, channels.
- cas-ssm: модули users, roles, self_user, tokens, user_notification, integration_access_token.
- cas-stream-analyzer: модули stream-stats, channels.
- cas-snmp-worker.

Порядок установки:

Разверните микросервисы в кластере Kubernetes1:

- `kubectl apply -f cas-mdb.yml` (YAML: apiVersion apps/v1, kind Deployment, selectors, image: cas-mdb:tag; dry-run: deployment.apps/cas-mdb configured, service/cas-mdb configured).
- `kubectl apply -f cas-ssm.yml` (replicas: 2, selectors, image из Docker Registry; dry-run: deployment.apps/cas-ssm configured).
- `kubectl apply -f cas-stream-analyzer.yml`.
- `kubectl apply -f cas-snmp-worker.yml` (зависимости: MongoDB, Redis для SNMP v3 мониторинга).

2. Компонент "ЕММ" (Модуль генерации ЕММ)

Назначение: Генерация управляющих сообщений (ЕММ) для абонентов.

Включает микросервисы и модули:

- cas-mdb: модуль emm (формирование сообщений).
- cas-emm: модуль emmg (агрегация и формирование TS-пакетов).

Порядок установки:

- Модуль emm является частью микросервиса cas-mdb.

Разверните микросервис cas-emm:

- `kubectl apply -f cas-emm.yml` (зависимости: MongoDB, Kafka, Encryptor; dry-run: deployment.apps/cas-emm configured)

Включенный в компонент ЕММ модуль DRM

- **cas-mdb:** модуль licenses (управление и выдача лицензий).
- **cas-ssm:** модуль licenses (проверка и верификация доступа по лицензии).

3. Компонент "Billing"

Назначение: Учет подписок, тарификация, управление платежами.

Включает микросервисы и модули:

- cas-mdb: модули invoices, subscriptions, payment_orders, tariffs, special_offers.
- cas-crm: модуль для управления интеграциями (Bitrix24, CRM оператора с помощью API).

Порядок установки:

- Модули биллинга являются частью микросервиса cas-mdb.
- Разверните микросервис CRM: `kubectl apply -f cas-crm.yml` (YAML: apiVersion apps/v1, kind Deployment, selectors; dry-run для проверки)

4. Компонент "BD" (База данных)

Назначение: База данных и механизм передачи сообщений.

Описание реализации: Компонент "BD" не является отдельно устанавливаемым модулем. Он представляет собой архитектурный слой, реализованный через:

- Брокер сообщений Kafka или REST API для обмена данными между микросервисами.
- Независимые наборы баз данных для каждого микросервиса (PostgreSQL, Redis, MongoDB, ClickHouse).

Развертывание этих зависимостей происходит в рамках установки соответствующих микросервисов (в первую очередь cas-mdb)

5. Компонент "SMS" (Управление информацией о потребителях)

Назначение: Управление информацией о потребителях и их подписках.

Включает микросервисы и модули:

- cas-mdb: модули subscriptions, emm.

Примечание: Функционал реализуется через микросервис cas-mdb, установка которого описана выше. Отдельная установка не требуется.

6. Компонент «Info»

Назначение: Формирование отчетов и доставка информационных сообщений.

Включает микросервисы и модули:

- cas-reports.
- cas-email.
- cas-mdb: модуль notification.

Порядок установки:

- `kubectl apply -f cas-email.yml` (SMTP relay; dry-run: deployment.apps/cas-email configured).
- `kubectl apply -f cas-telegram-bot.yml` (replicas: 2, selectors, metadata name: cas-telegram-bot)
- cas-report: Создайте файл cas-reports.yml с указанной конфигурацией, добавив путь до Docker-образа, учетные данные и адреса серверов. Разверните с помощью `kubectl apply -f cas-reports.yml`. Выполните dry-run для проверки: `kubectl apply -f cas-reports.yml --dry-run=client` (ожидаемый вывод: deployment.apps/cas-reports configured). Интеграция с PostgreSQL, Redis и MongoDB осуществляется через env variables.

7. Компонент Encryptor

Все операции по командам необходимо выполнять под суперпользователем (команда: `sudo -s`).

Порядок установки:

- Установите ПО на выделенный Edge-сервер.
- Скопируйте исполняемые файлы в /opt/encryptor.

Содержание архива Encryptor:

Конфигурационный файл – отвечает за конфигурацию компонента;

Скрипт для работы с логами компонента;

Файл для корректировки кодирования PID каналов;

Скрипт для запуска компонента в интерактивном режиме;

Скрипт для запуска компонента в docker;

Скрипт для остановки компонента в docker.

Скрипт для перезапуска компонента в docker

Включенный в компонент Encryptor модуль генерации ЕСМ

Назначение: Генерация сообщений управления шифрованием.

Порядок установки:

- Настройте конфигурационный файл, указав PID для аудио/видео потоков.
- Запустите Docker-контейнер encryptor.
- Интегрируйте с cas-cryptors-worker для UDP-обработки (YAML: cas-cryptors-worker.yml, selectors, dry-run: deployment.apps/cas-cryptors-worker configured)

8. Компонент Decryptor

Все операции по командам необходимо выполнять под суперпользователем (команда: `sudo -s`).

Порядок установки:

- Установите ПО на выделенный Edge-сервер.
- Скопируйте исполняемые файлы в /opt/decryptor.

Содержание архива Decryptor:

Конфигурационный файл – отвечает за конфигурацию компонента;

Скрипт для работы с логами компонента;

Скрипт для запуска компонента в интерактивном режиме;

Скрипт для запуска компонента в docker;

Скрипт для остановки компонента в docker.

Скрипт для перезапуска компонента в docker

9. Компонент Прошивка САМ-модуля

Назначение: Адаптированная версия компонента Decryptor для САМ-модуля.

Установка осуществляется путем записи в память ПЛИС (совместимого САМ-модуля) с помощью специализированного ПО или JTAG программатора. Команды и порядок прошивки САМ модуля зависят от модели и архитектуры модуля.

10. Компонент PCI-e modulator

Назначение: Позволяет использовать дескремблируемый поток для дальнейшей передачи на ТВ сети вещания.

Для корректной работы модулятора необходимо установить драйвера для соответствующей pci платы и настроить данный компонент для передачи транспортного потока.

Финальная проверка и Запуск

- Проверьте статус всех подов в кластере: `kubectl get pods -A`.
- Проверьте доступность сервисов через их внешние порты (NodePort).
- Запустите команду `VALIDATE_LEGAL` для автоматической проверки соответствия документации.
- После развертывания выполните `dry-run` для всех YAML для валидации конфигураций.