



cedrusdata

## Техническая архитектура CedrusData Catalog

ООО «Кверифай Лабс»

ОГРН 1217800163790

ИНН 7811766769

КПП 781101001

<b>Введение</b>	<b>3</b>
<b>1. Общая архитектура</b>	<b>3</b>
<b>2. Объекты каталога</b>	<b>4</b>
<b>3. Плагины</b>	<b>4</b>
<b>4. Технологический стек</b>	<b>5</b>

# Введение

CedrusData Catalog — это система управления метаданными для современных аналитических платформ.

Преимущества CedrusData Catalog: - Поддержка протокола Iceberg REST Catalog (<https://iceberg.apache.org/concepts/catalog/#decoupling-using-the-rest-catalog>) - Поддержка популярных аналитических систем обработки данных: CedrusData, Trino, Apache Spark, Apache Flink и др. - Поддержка файловых систем S3 и HDFS - Расширенные возможности мониторинга - Возможность приобретения коммерческой версии с технической поддержкой

Данный документ рассматривает ключевые аспекты архитектуры CedrusData Catalog.

## 1. Общая архитектура

Экземпляр CedrusData Catalog представляет собой отдельный процесс операционной системы.

Процесс запускает внутри себя сервер, который принимает пользовательские команды по протоколу HTTP или HTTPS.

Сервер реализует несколько точек доступа (endpoint), получающих и передающих данные в формате JSON:

1. Административные точки доступа — позволяют выполнять административные команды (например, создание принципала)
2. Точка доступа Apache Iceberg — позволяет работать с объектами Apache Iceberg по протоколу Apache Iceberg REST

При получении команды сервер производит аутентификацию и авторизацию действия пользователя. В случае успешной аутентификации и авторизации происходит передача команды на выполнение в ядро CedrusData Catalog. В противном случае сервер возвращает пользователю ошибку со статусом 401 или 403.

Ядро проверяет корректность содержимого команды. Если команда содержит некорректные параметры, ядро возвращает ошибку со статусом 400. Если команда корректна, ядро передает команду соответствующему обработчику.

В зависимости от природы команды, обработчик изменяет внутреннее состояние сервера. Выполнение команды может потребовать взаимодействия с озером данных через интерфейс файловой системы (S3 или HDFS), или персистентной базой данных (PostgreSQL или SQLite).

Если выполнение команды завершается успешно, сервер возвращает пользователю код 200. В противном случае сервер возвращает пользователю код 4xx или 5xx, где xx - число.

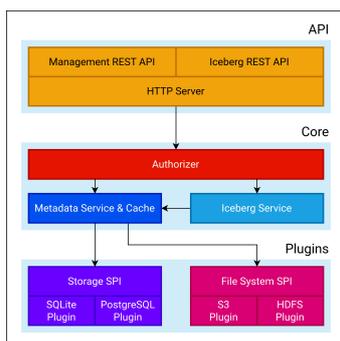


Рис. 1: Компоненты CedrusData Catalog

## 2. Объекты каталога

Команды к CedrusData Catalog приводят к чтению или изменению состояния тех или иных объектов CedrusData Catalog. Объекты CedrusData Catalog организованы в иерархию:

1. CedrusData Catalog — объект верхнего уровня, экземпляр CedrusData Catalog
2. Principal — принципал CedrusData Catalog. Представляет собой пользователя или роль. Выполнение каждой команды CedrusData Catalog происходит от имени определенного принципала
3. File System — подключение к распределенной файловой системе озера данных (S3 или HDFS)
4. Iceberg Catalog — логический каталог Apache Iceberg, который хранит схемы Apache Iceberg. Логический каталог имеет ассоциированную с ним файловую систему
5. Iceberg Namespace — логическая схема Apache Iceberg, которая хранит таблицы и виртуальные представления Apache Iceberg
6. Iceberg Table — таблица Apache Iceberg
7. Iceberg View — виртуальное представление Apache Iceberg

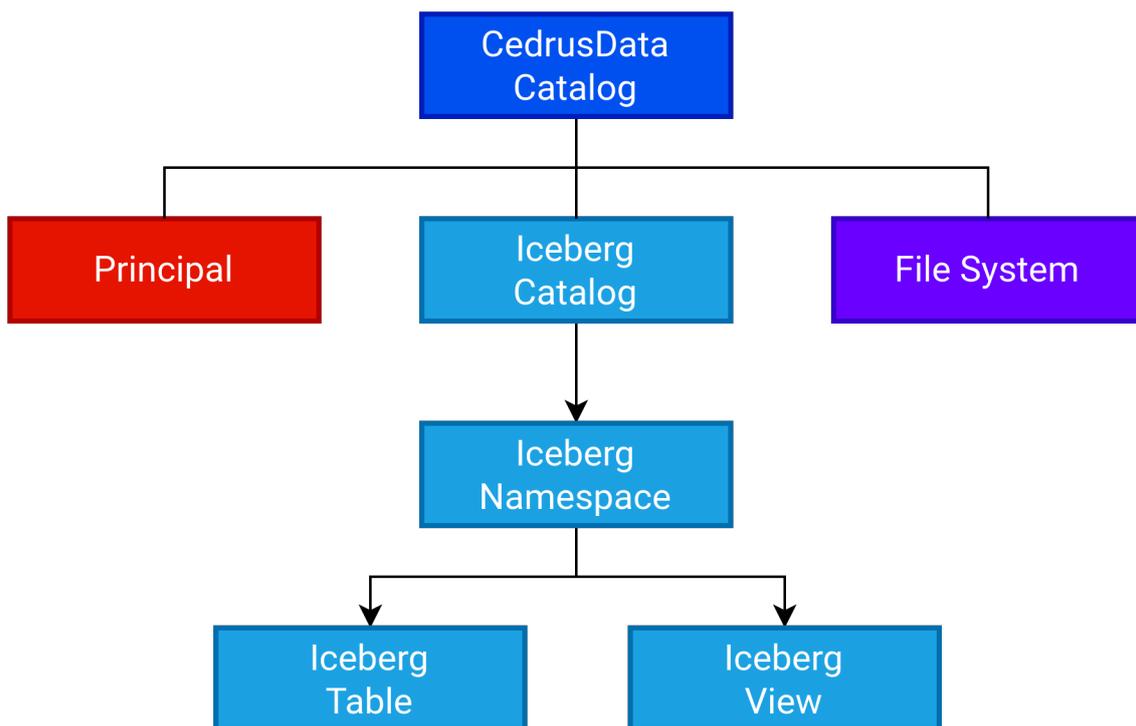


Рис. 2: Иерархия объектов CedrusData Catalog

## 3. Плагины

Взаимодействие CedrusData Catalog со внешними сервисами организовано с помощью плагинов.

Существует несколько типов плагинов. Каждый тип плагина описывает набор интерфейсов на языке Java, которые необходимо реализовать для получения работоспособного плагина. После реализации плагина он добавляется в поставку CedrusData Catalog.

CedrusData Catalog поддерживает два типа плагинов:

1. Плагин базы данных
2. Плагин файловой системы

### 3.1 Плагин базы данных

Плагин базы данных обеспечивает доступ к персистентной СУБД для сохранения состояния CedrusData Catalog между перезапусками.

Интерфейс плагина состоит из набора методов для чтения или изменения состояния персистентной СУБД. Например, метод `createPrincipalIfExists` реализует функцию создания нового принципа CedrusData Catalog, а метод `getPrincipalDetailsByName` возвращает информацию о принципе на основе его уникального имени.

Поставка CedrusData Catalog включает два плагина базы данных:

1. Плагин SQLite — позволяет сохранять состояние каталога в локальном экземпляре СУБД SQLite. Работает по протоколу JDBC. Подходит для изучения продукта и тестовых стендов.
2. Плагин PostgreSQL — позволяет сохранять состояние каталога в экземпляре СУБД PostgreSQL. Работает по протоколу JDBC. Подходит для промышленного развертывания.

### 3.1 Плагин файловой системы

Плагин файловой системы обеспечивает доступ к распределенной файловой системе озера данных для работы с файлами Apache Iceberg.

Плагин файловой системы задействован в операциях, связанных с изменением файлов Apache Iceberg. Например, такими операциями являются: создание таблицы Apache Iceberg, фиксация (commit) обновления таблицы Apache Iceberg, и т.д.

Интерфейс состоит из набора методов для работы с файловой системой: получение списка файлов, создание файла, удаление файла, чтение содержимого файла, и т.д.

Поставка CedrusData Catalog включает два плагина файловой системы:

1. Плагин S3 — позволяет работать с распределенными файловыми системами по протоколу S3. Например, MinIO, CEPH, Yandex Object Storage, и т.д.
2. Плагин HDFS — позволяет работать с распределенной файловой системой HDFS

## 4. Технологический стек

CedrusData Catalog реализован на языках программирования Java и Python.

Предварительными требованиями для запуска CedrusData Catalog являются:

- При запуске из архива - наличие на компьютере JDK 21 и Python версии 2.x или 3.x
- При запуске в Docker-контейнере - наличие на компьютере Docker.