

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на создание ETL-пайплайна сбора метрик ERC-4626-волта и их загрузки в БД  
(проект «Vault Metrics ETL»)

## 1. Введение

**Полное наименование разработки:** «ETL-пайpline сбора и загрузки метрик ERC-4626-волта в БД».

**Краткое наименование:** «Vault Metrics ETL».

**Краткая характеристика области применения:**

Пайплайн предназначен для регулярного извлечения ончейн-данных из смарт-контрактов стандартного типа ERC-4626 (например, Kelp sbUSD Vault на сети Ethereum: [0x8ECC0B419dfe3AE197BC96f2a03636b5E1BE91db](#)), вычисления целевых метрик (TVL и цена доли/Share Price) и их сохранения в реляционной БД (PostgreSQL) для аналитики, отчетности и визуализации.

## 2. Основания для разработки

- Настоящее ТЗ составлено по ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению»; допускается уточнение содержания разделов и включение приложений.
- Учебно-практическое задание: «Создать ETL-пайплайн, реализующий сбор указанных метрик и их сохранение в БД за заданный период; приложить схему пайплайна».
- Тема разработки: «Создание ETL-пайплайна сбора и сохранения метрик ERC-4626-волта».
- Шифр темы: **Vault-ETL-4626**.

## 3. Назначение разработки

**Функциональное назначение:**

Обеспечить автоматизированный сбор ончейн-данных из адресов волтов ERC-4626 по расписанию и/или за заданный период (backfill), расчёт метрик и запись результатов в БД с гарантиями идемпотентности и возможностью расширения под новые волты/сети/метрики.

## **Эксплуатационное назначение:**

Пайплайн работает как периодическая фоновая задача (cron/systemd/Airflow) в среде Linux/macOS; оператор получает доступ к данным через SQL-клиенты (DBeaver/psql) и BI.

## **4. Требования к программе (программному изделию)**

### **4.1. Требования к функциональным характеристикам**

1. **Источники данных:** RPC-узел Ethereum (HTTPS), перечень адресов волтов ERC-4626 (конфигурация).
2. **Извлечение (Extract):** вызовы view-функций контракта ERC-4626:
  - asset() → адрес базового ERC-20;
  - totalAssets() → общий объём базового актива под управлением волта;
  - totalSupply() → суммарное число долей волта (share-токена).
3. **Трансформация (Transform):** расчёт метрик:
  - **TVL\_ASSET** = totalAssets() / 10<sup>decimals(asset)</sup>.
  - **SHARE\_PRICE** = totalAssets() / totalSupply() (при totalSupply()==0 → 0).
4. Примечание: для ERC-4626 взаимосвязь assets↔shares и функции конверсии определяют «цену доли» (exchange rate/price per share), что корректно отражает накопленную доходность; такие формулировки согласуются с практикой и разъясняющими материалами.
5. **Загрузка (Load):** запись в PostgreSQL: vault\_id, metric\_type\_id, value\_numeric, block\_number, block\_timestamp (если получен), collected\_at, source.
6. **Режимы работы:**
  - Инкрементальный сбор по расписанию (по умолчанию каждые 5 минут).
  - Backfill за период по времени/блокам с возможностью шага (по умолчанию 300 сек).
7. **Идемпотентность:** уникальность записи по (vault\_id, metric\_type\_id, block\_number); повторные запуски не создают дубликатов (UPSERT/ON CONFLICT DO NOTHING).
8. **Конфигурация:** .env: RPC\_URL, DB\_URL, VAULT\_ADDRESSES, POLL\_INTERVAL\_SEC, BACKFILL\_{START,END,STEP}, LOG\_LEVEL, SCRIPT\_VERSION.
9. **Логирование:** файл логов с отметками успешных итераций, временными метками, описаниями ошибок.

## **4.2. Требования к надёжности**

1. Ретрай RPC при таймаутах/ошибках (экспоненциальная задержка).
2. Проверки значений (неотрицательность, не-NaN/Inf); при нарушении — запись не производится, событие логируется.
3. Корректное завершение при недоступности БД; повтор через следующий цикл.

## **4.3. Условия эксплуатации**

1. ОС: Linux/macOS; стабильный интернет-доступ к RPC и доступ к PostgreSQL:5432.
2. Планировщик: cron/systemd; допускается Airflow (как вариант расширения).

## **4.4. Требования к составу и параметрам технических средств**

- Минимум: 1 vCPU, 1 ГБ RAM, 2 ГБ диска, стабильный канал сети (HTTPs к RPC).

## **4.5. Требования к информационной и программной совместимости**

- Python 3.10+; библиотеки: web3.py, SQLAlchemy, psycopg2/psycopg.
- PostgreSQL 13+; совместимость со средствами просмотра (psql/DBeaver).

## **4.6. Требования к защите информации (специальные требования)**

- Маскирование секретов в логах.

# **5. Требования к программной документации**

- Настоящее **ТЗ** по ГОСТ 19.201-78.
- **Руководство программиста/README** (сборка, конфигурация, запуск, backfill).
- **Схема БД (DDL)** и **диаграмма** (ERD/mermaid).
- **Схема пайплайна** (flowchart/UML, mermaid).
- **Журнал изменений** (CHANGELOG).

# **6. Технико-экономические показатели**

- Периодичность: каждые 5 минут (по умолчанию).

- Средняя длительность итерации на 1 волт: ≤ 2 сек (view-вызовы).

## 7. Стадии и этапы разработки

1. **Проектирование:** согласование требований; схема БД; схема пайплайна.
2. **Разработка:** модули Extract/Transform/Load, конфиг, логирование, идемпотентность.
3. **Тестирование:** юнит-тесты формул; интеграционные тесты записи в БД; тест backfill.
4. **Ввод в эксплуатацию:** деплой; настройка расписания; контроль первых 24 ч.
5. **Сопровождение:** добавление волтов/метрик; оптимизация; агрегации/материализованные представления; ретеншн.

## 8. Порядок контроля и приёмки

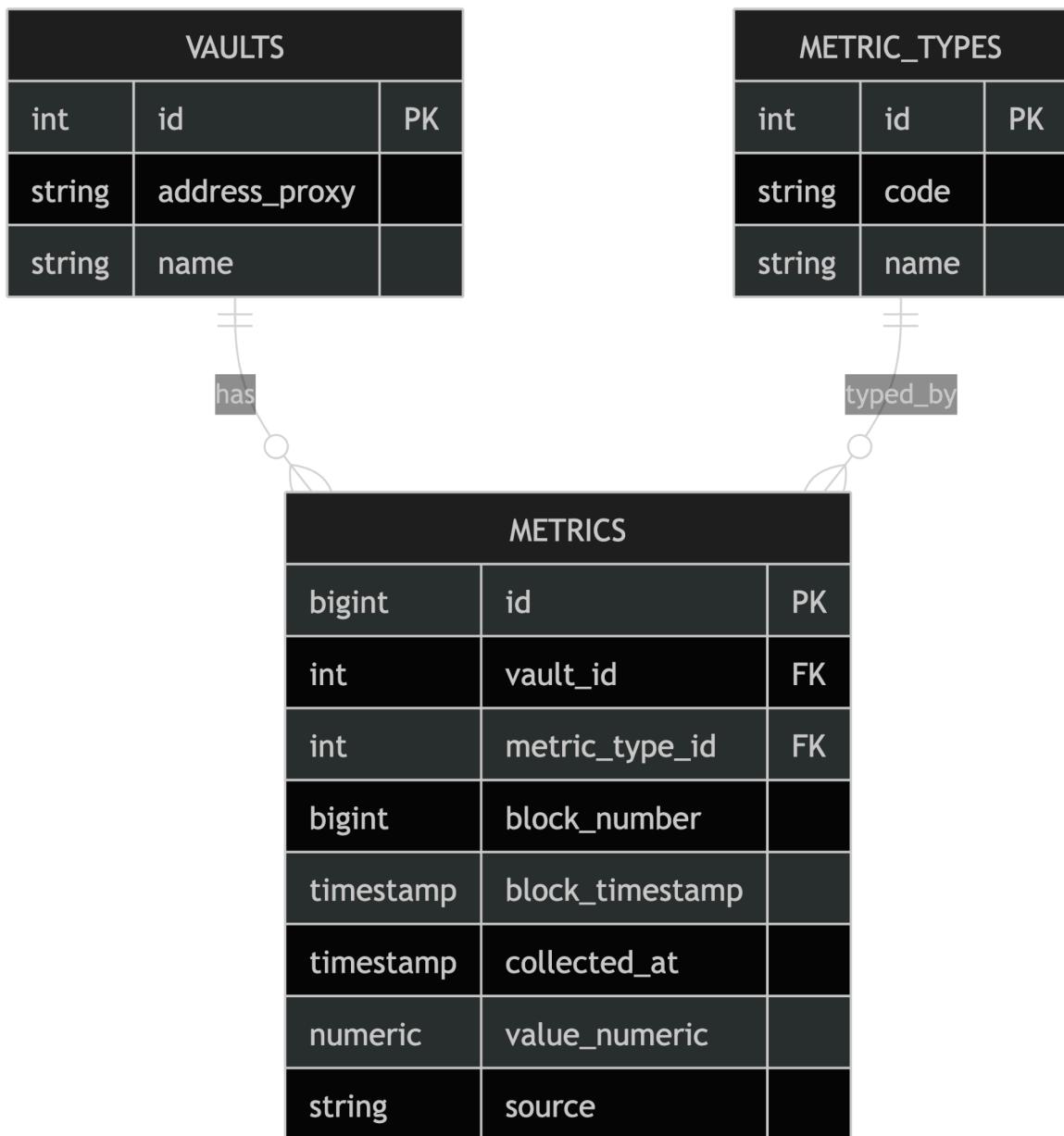
### Контроль качества:

- Проверка наличия и корректности данных каждые 5 минут (не менее 288 точек/метрику за сутки).
- Идемпотентность: отсутствие дублей по (vault\_id, metric\_type\_id, block\_number).
- Сопоставление значений с ручным вызовом totalAssets()/totalSupply() (Etherscan/скрипт) на тех же блоках.

### Приёмка:

- Успешный инкрементальный прогон ≥ 24 ч без критических ошибок.
- Успешный backfill за заданный период (например, 7 дней) с верификацией диапазона блоков и полноты данных.
- Наличие всей документации из раздела 5.

## Приложение А: ER-диаграмма БД



Данное Тз подготовлено с использованием Референса :

<https://pro-prof.com/forums/topic/%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE-%D0%B7%D0%B0%D0%B4%D0%BD%D0%BD%D0%B8%D1%8F-%D0%BF%D0%BE-%D0%B3%D0%BE%D1%81%D1%82-19-201-78>