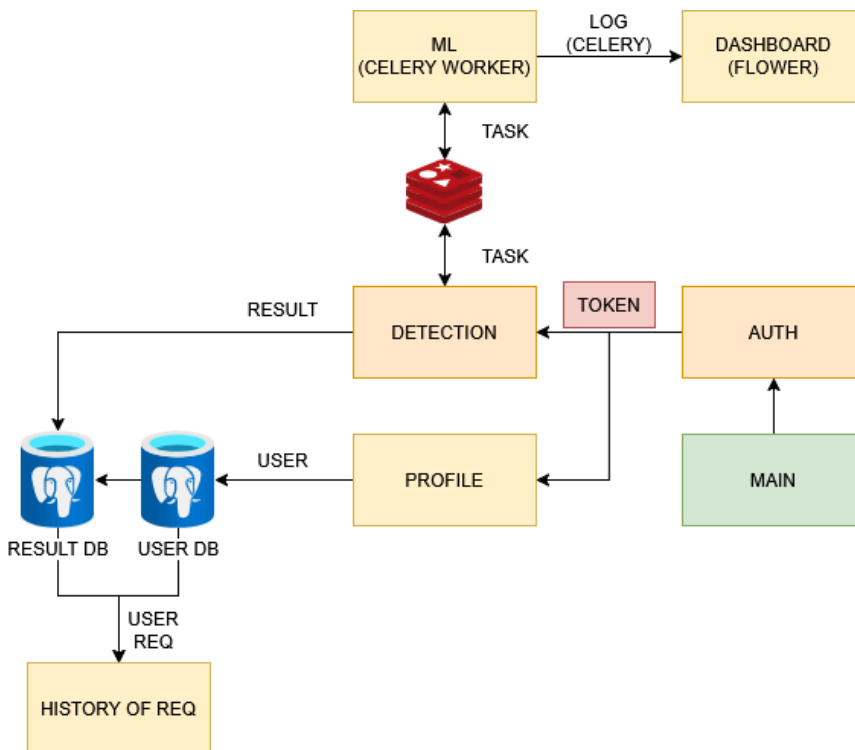


Backend schema



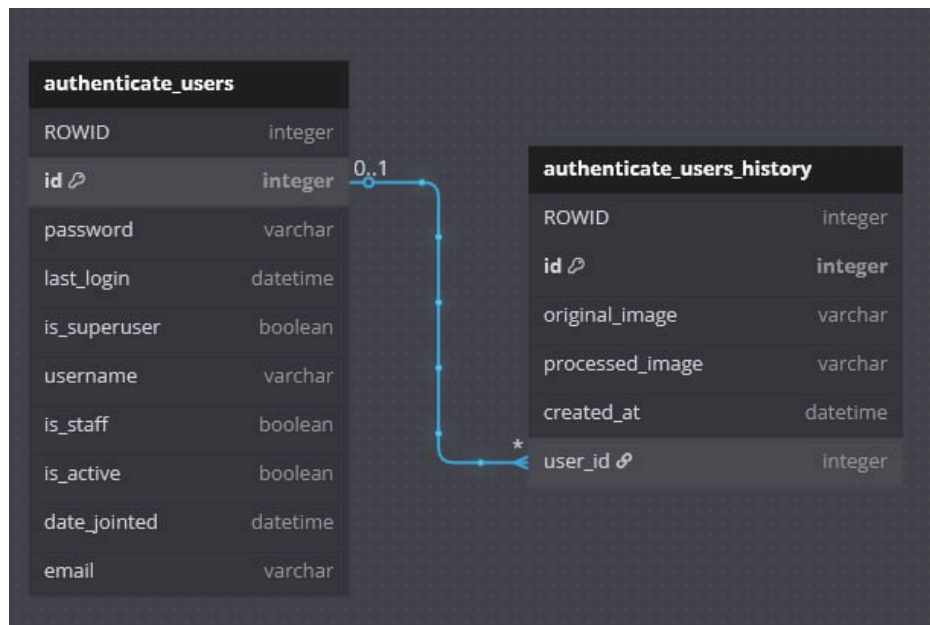
- Красные элементы - не начато (есть только на бумаге)
- Жёлтые - уже в разработке (есть код)
- Зелёные - готово (new feature ready)

Основные моменты

1. Передача токенов осуществляется по внутренке django
2. Требуется написать AUTH с подключением oauth
3. ML работает асинхронно на основе celery
4. celery собирает свои логи в борду flower
5. Результаты ML ещё НЕ связаны с DB
6. В MAIN просто view отображение
7. Detection service разбит на два модуля (DETECTION и ML) с целью дальнейшей масштабируемости решения (просто повышать кол-во worker серверов)
8. от Redis как message broker скорее всего откажемся в пользу rabbitmq (он быстрее в данной задаче и устойчивее с данными)

Текущая схема БД

User table и её отношение к Result table



Идея

Классическое приложение-арі. Пользователь регистрируется (получает токен вошедшего) и может получить доступ к инструментарию нейросетки. Нейронка крутится на отдельном worker сервере и работает асинхронно. Пользователь так же может

- получить арі ключ для доступа к модели (пока не может))
- посмотреть историю своих запросов в модель (т.е. посмотреть анализ)

Дальнейшие идеи

- Парсинг архивов (только через frontend)
- Если получился пункт выше, то указание координат (если есть в метаданных картинок) к каждому объекту
- немного переписать ML сервис, чтобы он асинхронно ещё и в бд кидал результаты