FileBot

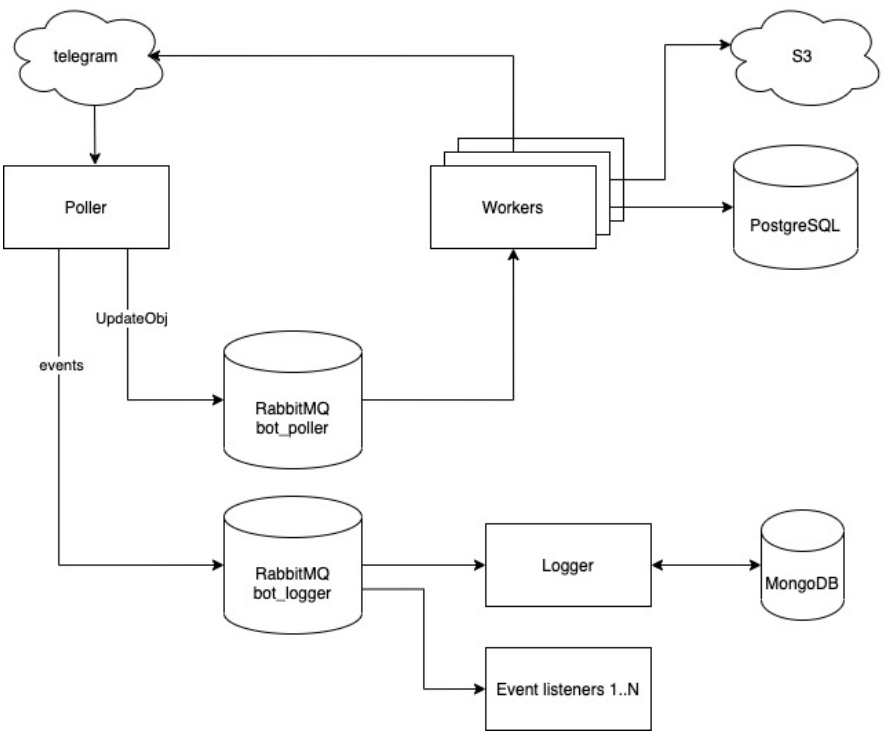
В прошлой главе мы разрабатывали FileBot.

Он принимал от пользователя файлы и сохранял их в файловое хранилище S3. В этом задании мы его улучшим. А именно: перепишем так, чтобы он поддерживал микросервисную архитектуру, взаимодействовал с брокером сообщений RabbitMQ, хранил состояние в Postgres и записывал события в MongoDB.

Основные требования

- сохранить функциональность предыдущего бота
- взаимодействовать между сущностями бота с помощью брокера сообщений RabbitMQ
- хранить состояние о пользователе в базе данных PostgreSQL
- сохранять новые сообщения из Telegram в базе данных MongoDB

Архитектура



Poller. Как и в предыдущем задании, получает уведомления и складывает их в очередь. Но теперь он складывает уведомления в RabbitMQ.

Worker. Как и в предыдущем задании, worker получает задачи из очереди, выполняет полезную работу и реализует бизнес-логику бота. Но теперь задачи он получает из RabbitMQ и хранит информацию о пользователях в PostgreSQL.

Logger. Новая сущность, позволяет подключаться к общей шине событий bot_logger, получать события из шины и записывать их в MongoDB.

В следующих карточках мы подробнее разберем, что нужно сделать в каждом компоненте.

Перед выполнениям задания нужно создать таблицу в Postgres. Зайдите в терминал в mercury и выполните команду:

```
psql $POSTGRES_DSN < bot/data/tg_users.sql
```

```
create table tg_users
(
        id serial
                constraint tg_users_pk
                        primary key,
        tg_id bigint not null,
        username varchar(256) not null
);
```