

Программа для ЭВМ «Система анализа и мониторинга событий»

Архитектура системы

Оглавление

1. Общее описание архитектуры	3
2. Компоненты системы	3
2.1 news-boiler – сборщик новостей.....	3
2.2 news-fermenter – ML-обогащение	4
2.3 boiler-proxy – API доступа	4
2.4 news-viewer – графическое представление событий.....	4
2.5 news-editor – редакция контента	4
2.6 app-post – серверная часть редакции контента	4
2.7 Инфраструктурные сервисы	5
3. Поток данных	5
4. Технологический стек.....	6
5. Системные требования	6

1. Общее описание архитектуры

Система построена в соответствии с основами микросервисной архитектуры, используется взаимодействие слабо связанных, хорошо масштабирующихся и легко изменяемых модулей – микросервисов. Данная архитектура подразумевает использование широко известных стандартов взаимодействия между компонентами, что обеспечивает возможность реализовать требуемую логику работы внутри создаваемого модуля, а в качестве вспомогательных компонентов/сервисов использовать готовые решения.

Программа для ЭВМ «Система анализа и мониторинга событий» представляет собой набор приложений, написанных на Python 3.10/3.11. Основу решения составляют микросервисы сбора новостей (news-boiler), ML-обогащения (news-fermenter) и API-доступа (boiler-proxy). Графический интерфейс реализован в виде отдельного веб-приложения (news-viewer, news-editor) на базе React, обеспечивающего просмотр, фильтрацию и редактирование событий и публикаций. Также используется большой набор open-source библиотек: FastAPI, Celery, CatBoost, SentenceTransformer, natasha и другие.

Для хранения данных используются MongoDB 4.4 (основное хранилище новостей), PostgreSQL (обработанные события) и Elasticsearch 7.17.5 (полнотекстовый поиск и индексирование). Брокер очередей – RabbitMQ 3.9.

Развертывание системы ведется в среде Docker с использованием Docker Compose. Компонент boiler-proxy предоставляет REST API интерфейс на базе FastAPI для внешних потребителей.

Программа обладает графическим интерфейсом, включающим:

- news-viewer – пользовательский интерфейс просмотра и фильтрации событий и публикаций, реализованный на базе React;
- news-editor – пользовательский интерфейс создания и редактирования контента (события, публикации, справочники), реализованный на базе React;
- админ-панели управления источниками и аккаунтами (Flask);
- интерфейс мониторинга Flower.

2. Компоненты системы

2.1 news-boiler – сборщик новостей

Веб-краулер на базе Celery. Собирает новости из семи типов источников (RSS, Telegram, VK, Дзен.Новости, веб-страницы) и сохраняет их в MongoDB (boiler / коллекция news). Включает 9 Celery-воркеров с различным уровнем конкуренции, планировщик beat, мониторинг Flower, админ-панели управления аккаунтами и источниками.

Celery-воркеры:

Воркер	Очередь	Concurrency
rss_worker	rss	6
tg_worker	tg	2
vk_worker	vk	2
yandex_worker	yandex_news	1
web_worker	web	6
service_worker	service_queue	10
archive_elastic_worker	archive_queue	1
parent_service_worker	parent_queue, celery	1

2.2 news-fermenter – ML-обогащение

Сервис ML-обработки новостей. Запускается каждые 3 минуты, забирает необработанные посты из MongoDB (is_fermented=False) и выполняет: предобработку текстов, извлечение NER-тегов (natasha) и классификацию регионов (CatBoost).

2.3 boiler-proxy – API доступа

FastAPI-сервис (порт 8885), предоставляющий read-only доступ к новостям из MongoDB. Все эндпоинты требуют токен авторизации (Authorization header). Поддерживает пагинацию, фильтрацию по регионам, поиск по каналам и регулярным выражениям, доступ к Дзен-ТОПу.

2.4 news-viewer – графическое представление событий

React-приложение, предоставляющее пользовательский интерфейс для просмотра и фильтрации событий и публикаций. Отображает сводку событий с возможностью фильтрации по регионам, источникам и временному диапазону. Доступ к приложению требует авторизации. Взаимодействует с boiler-proxy через REST API.

2.5 news-editor – редакция контента

React-приложение, обеспечивающее графический интерфейс для создания, редактирования и управления контентом системы. Позволяет работать с событиями, публикациями и справочниками. Включает административную панель управления источниками и аккаунтами. Доступ требует авторизации. Взаимодействует с app-post через REST API.

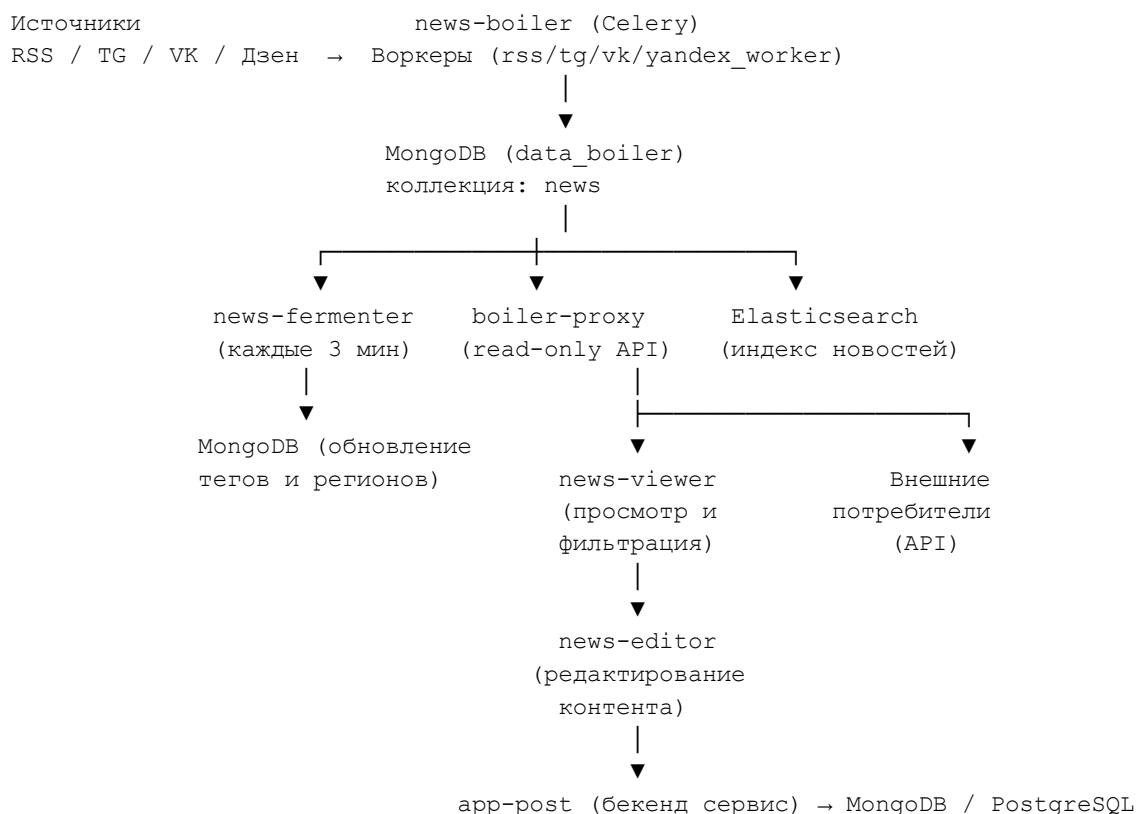
2.6 app-post – серверная часть редакции контента

Бэкенд-сервис, обеспечивающий серверную логику для приложения news-editor. Предоставляет REST API для создания, редактирования и удаления событий, публикаций и справочников, а также для управления источниками и аккаунтами пользователей.

2.7 Инфраструктурные сервисы

Сервис	Образ	Назначение
rabbit_boiler	RabbitMQ 3.9	Брокер задач Celery (15673/5673)
data_boiler	MongoDB 4.4	Основная БД новостей (57015)
archive_data_boiler	MongoDB 4.4	Архивная БД новостей (27037)
accounts_db_boiler	MongoDB 4.4	БД аккаунтов краулера (57016)
elastic	Elasticsearch 7.17.5	Поиск и индексирование (9200)
selenium-hub + chrome×5	Selenium Grid	JS-рендеринг страниц (4444)
prometheus_client	news-prometheus	Сбор метрик (8080)
prometheus_server	Prometheus	Хранение метрик (9090)
pg-db	PostgreSQL	Обработанные события и посты

3. Поток данных



Жизненный цикл статьи:

- Краулер собирает статью → сохраняет в MongoDB с is_fermented=False
- Ферментер забирает необработанные статьи → обогащает NER, регионами
- Ферментер обновляет MongoDB (теги, регионы)

- Параллельно boiler-proxy отдает статьи из MongoDB внешним потребителям по API
- news-viewer получает данные через boiler-proxy → отображает статьи и события пользователю в графическом интерфейсе с возможностью фильтрации
- Пользователь в news-editor создает или редактирует контент (события, публикации, справочники) → изменения сохраняются через app-post в MongoDB / PostgreSQL

4. Технологический стек

Категория	Технологии
Очереди	Celery + RabbitMQ
БД	MongoDB 4.4, PostgreSQL
Поиск	Elasticsearch 7.17.5
HTTP API	FastAPI (boiler-proxy), Flask (админ-панели)
ML	CatBoost
NLP	natasha (NER), newspaper3k
Браузер	Selenium Grid (Chrome × 5)
Мониторинг	Prometheus, Flower
Соцсети API	Telethon (TG), vk-api (VK)

5. Системные требования

Программные требования:

- ОС Linux Debian/Ubuntu/RHEL или другая Linux-система, поддерживаемая Docker
- Docker не менее 20.10
- Docker Compose не менее 3.5

Аппаратные требования:

- Вычислительный узел: 16 ядер ЦПУ, 32 ГБ ОЗУ, 100 ГБ SSD
- Система хранения: 8 ядер, 16 ГБ ОЗУ, 1 ТБ SSD