

Архитектура системы «Фитокуб»

Техническое описание

Содержание

| | |
|---|---|
| 1. Введение | 2 |
| 2. Общая архитектура системы | 2 |
| 2.1. Основные элементы взаимодействия | 2 |
| 3. Компоненты системы | 3 |
| 3.1. Устройство (ESP32) | 3 |
| 3.2. MQTT Broker | 3 |
| 3.3. Backend (Java + Spring Boot) | 3 |
| 3.4. Frontend (React) | 3 |
| 3.5. Database (PostgreSQL) | 3 |

1. Введение

Система «Фитокуб» предназначена для интеллектуального ухода за комнатными растениями. Устройство на базе **ESP32** анализирует состояние окружающей среды при помощи датчиков и управляет поливом через помпу и реле. Пользователь наблюдает данные в веб-приложении, где отображаются актуальные параметры растения и история измерений. Основная идея проекта — создать «умную оболочку» для горшка с растением, которая поддерживает оптимальные условия роста, уведомляет пользователя при отклонениях и позволяет удалённо управлять оборудованием.

1. Общая архитектура системы

Высокоуровневая архитектура объединяет устройство, облачную инфраструктуру и клиентские приложения. Все взаимодействия происходят через MQTT-брокер и защищённые интерфейсы backend-сервиса.

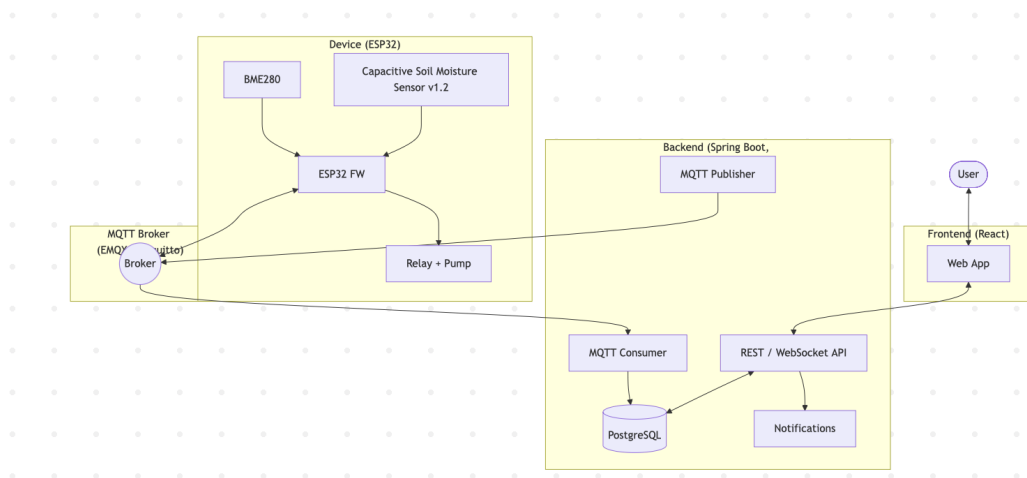


Figure 1: Рисунок 1 — Общая архитектура системы «Фитокуб».

1. Основные элементы взаимодействия

- Устройство ESP32 собирает телеметрию и реагирует на команды управления.
- MQTT Broker обеспечивает двусторонний обмен сообщениями между устройством и сервером.
- Backend-сервис выполняет обработку данных, хранение в БД и выдачу API.
- Веб-клиент отображает состояние растений, предоставляет историю показаний и отправляет команды.

1. Компоненты системы

1. Устройство (ESP32)

- Управляет реле и помпой полива.
- Считывает данные с датчиков **BME280** и датчика влажности почвы.
- Периодически публикует телеметрию в топик.
- Подписывается на команды из топика.
- Поддерживает локальные сценарии (например, экстренное отключение насоса при переполнении).

1. MQTT Broker

- Ведёт учёт подключений для мониторинга доступности устройств.

1. Backend (Java + Spring Boot)

- Получает телеметрию от брокера через MQTT-consumer и валидацию сообщений.
- Сохраняет данные в **PostgreSQL**.
- Отправляет команды устройствам через MQTT-publisher и подтверждает доставку.
- Обеспечивает REST и WebSocket API для пользовательских клиентов.
- Реализует аутентификацию и авторизацию пользователей.

1. Frontend (React)

- Показывает Dashboard состояния растений и графики истории измерений.
- Получает обновления по WebSocket / SSE.
- Позволяет управлять устройствами: запуск полива, настройка порогов и расписаний.
- Иницирует процессы привязки и открепления устройств.

1. Database (PostgreSQL)

- Хранит сущности телеметрии, данные пользователей и их устройства.