EMSIB – Analytics Component

Technical Report – 1st Integration

Authors: Oleksandra Turovets, Marta Penina

Course: Software Engineering
Academic Year: 2025/2026

1. Introduction

The purpose of this document is to describe the first integration of the **Analytics Component** developed as part of the **EMSIB (Energy Management System in Intelligent Buildings)** project.

This component is responsible for analyzing incoming energy-related data and detecting abnormal patterns that may indicate inefficiencies, faults, or unexpected behavior in the building's energy usage.

The Analytics Component operates as a backend service implemented in **Python**, designed to be integrated with other EMSIB modules through REST APIs and a shared database.

2. Component Responsibilities

The Analytics Component performs the following main responsibilities:

2.1 Energy Data Analysis

Processes energy consumption data stored in the database and evaluates it using analytical rules and algorithms.

2.2 Anomaly Detection

Identifies abnormal energy usage patterns such as sudden spikes or unexpected consumption behavior.

2.3 Anomaly Exposure via API

Provides REST endpoints that allow other EMSIB components to query detected anomalies.

3. Project Structure

The project follows a modular and extensible structure:

```
analytics-component/
│
├── app/
│   ├── main.py
│   ├── config.py
│   ├── database.py
│   ├── models/
│   ├── api/
│   │   └── anomalies.py
│   └── services/
│       └── energy_spike_detector.py
│
├── tests/
│   └── test_energy_spike_detector.py
│
├── sql/
│   ├── create_tables.sql
│   └── anomaly_queries.sql
│
├── requirements.txt
├── README.md
```

File responsibilities:

- **main.py** – Application entry point and FastAPI initialization
- **database.py** – Database connection and session management
- **anomalies.py** – REST API endpoints related to anomalies
- **energy_spike_detector.py** – Core logic for detecting energy spikes
- **tests/** – Unit tests for analytics logic
- **sql/** – SQL scripts for database schema and queries

4. Implemented API Endpoints

4.1 GET /anomalies/spikes

Description:
Returns detected energy spike anomalies based on stored consumption data.

Input:
No request body required.

Output example:


```
[
  {
    "device_id": "meter-01",
    "timestamp": "2024-11-12T09:00:00Z",
    "value": 150.4,
    "type": "energy_spike"
  }
]
```


Internal behavior:
The endpoint queries the database and applies the energy spike detection logic defined in the service layer.

5. Internal Module Descriptions

5.1 Energy Spike Detector

Implements the core analytics logic for identifying abnormal increases in energy consumption. Threshold-based rules are applied to historical data to detect anomalies.

5.2 API Layer

Defines REST endpoints using FastAPI and exposes analytics results to other EMSIB components.

5.3 Database Module

Handles communication with the relational database and executes predefined SQL queries.

6. How to Run the Component

Install dependencies

pip install -r requirements.txt

Run the server

uvicorn app.main:app --reload

Access API documentation

Once running, open:

http://127.0.0.1:8000/docs

FastAPI automatically generates interactive Swagger documentation.

7. Testing the Component

The Analytics Component includes unit tests located in the tests/ directory.

Tests focus on:

- correctness of anomaly detection logic
- handling of edge cases

Tests can be executed using:

pytest

8. Conclusion

The Analytics Component successfully implements the required functionality for detecting and exposing energy consumption anomalies within the EMSIB system.

The first integration demonstrates that the component can be seamlessly connected with other system modules through REST APIs and a shared database.
Its modular design allows future extension with more advanced analytics and machine learning techniques.

This component provides a solid foundation for intelligent energy monitoring and optimization within smart buildings.