The Journal of Open Source Software

# MAUDEMetrics: Automated extraction and comprehensive analysis of FDA MAUDE device safety data

**Mohamed Marouf** [1]

**1** Faculty of Medicine, Mansoura University, Mansoura, Egypt

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:**

**Submitted:** 07 September 2025
**Published:** unpublished



## Summary

Monitoring medical device safety is critical for patient safety and regulatory compliance. The FDA's Manufacturer and User Facility Device Experience (MAUDE) database contains over 22 million adverse event reports and remains a primary source for postmarket surveillance (U.S. Food and Drug Administration, 2025).
The FDA receives several hundred thousand new medical device reports annually, which are accessible via the openFDA Device Event API (openFDA, 2025).

However, practical use of this database is constrained by the API's 1,000-record pagination limit, the FDA website's 500-record export restriction (with missing demographic fields), and the need for significant technical expertise to aggregate and analyze reports.
These barriers make comprehensive device safety analysis difficult, time-consuming, and error-prone.

**MAUDEMetrics** is a Python-based web application that democratizes access to MAUDE data by automating extraction, aggregation, and analysis of adverse event reports via the openFDA API.
It provides an intuitive web interface, requires no programming expertise, and supports complex queries across brand names, product codes, manufacturers, and date ranges.
The tool automatically handles API pagination, validates and standardizes fields, and produces professional multi-sheet Excel exports with raw data, cleaned datasets, and statistical summaries.

By reducing analysis time from hours to minutes, MAUDEMetrics enhances reproducibility, efficiency, and accessibility of MAUDE data for clinicians, researchers, quality assurance teams, and regulatory professionals.

## Statement of Need

Medical device safety analysis is essential for patient safety and regulatory compliance, yet current MAUDE access methods impose significant barriers:

- **API Limitations**: The openFDA API restricts results to 1,000 records per query, requiring complex iteration (openFDA, 2025).
- **Web Interface Limitations**: The MAUDE portal only allows 500-record exports and omits essential fields such as demographics (U.S. Food and Drug Administration, 2025).
- **Manual Burden**: Researchers must perform multiple searches, downloads, and merges to assemble datasets.

- **Reproducibility Challenges**: Manual extraction is difficult to replicate and error-prone.

**MAUDEMetrics** addresses these gaps by providing:

- An accessible, programming-free web interface
- Intelligent pagination to assemble complete datasets
- Multi-parameter search across brand names, product codes, manufacturers, product classes, and date ranges
- Automated cleaning and validation of >100 FDA data fields
- Multi-sheet professional Excel exports with summaries
- Built-in visualization dashboard for event types and device brands
- Docker containerization for reproducible workflows

The software benefits clinical researchers, hospital QA teams, regulatory professionals, and healthcare institutions conducting device safety audits.

## Implementation

MAUDEMetrics is implemented as a **Flask** web application (Flask Development Team, 2023) that integrates directly with the openFDA Device Event API.
The `fetch_all_API_data()` function automates API pagination and error handling, while **Pandas** (McKinney, 2010) and **NumPy** (Harris et al., 2020) perform data processing.
Data are cached locally in **SQLite** (SQLite Development Team, 2023) for performance and persistence.

The export engine provides two complementary options for researchers:

- **Raw Data Export**: reports are saved as CSV files that preserve the original JSON structure from the openFDA API, ensuring no loss of information.
- **Processed Excel Export**: using **OpenPyXL** (Clark, 2023), the system generates multi-sheet Excel workbooks with:
  - **Events**: cleaned and standardized event data with integrated narratives
  - **Summary**: aggregated statistics of patient demographics and event types

Deployment uses **Docker** (Docker Inc., 2023) for reproducibility, and the web interface is styled with Bootstrap 5.
Testing covers API integration, database operations, export generation, and web routes, with a 100% pass rate.
Performance benchmarks show ~5,000 records extracted in under 30 seconds.

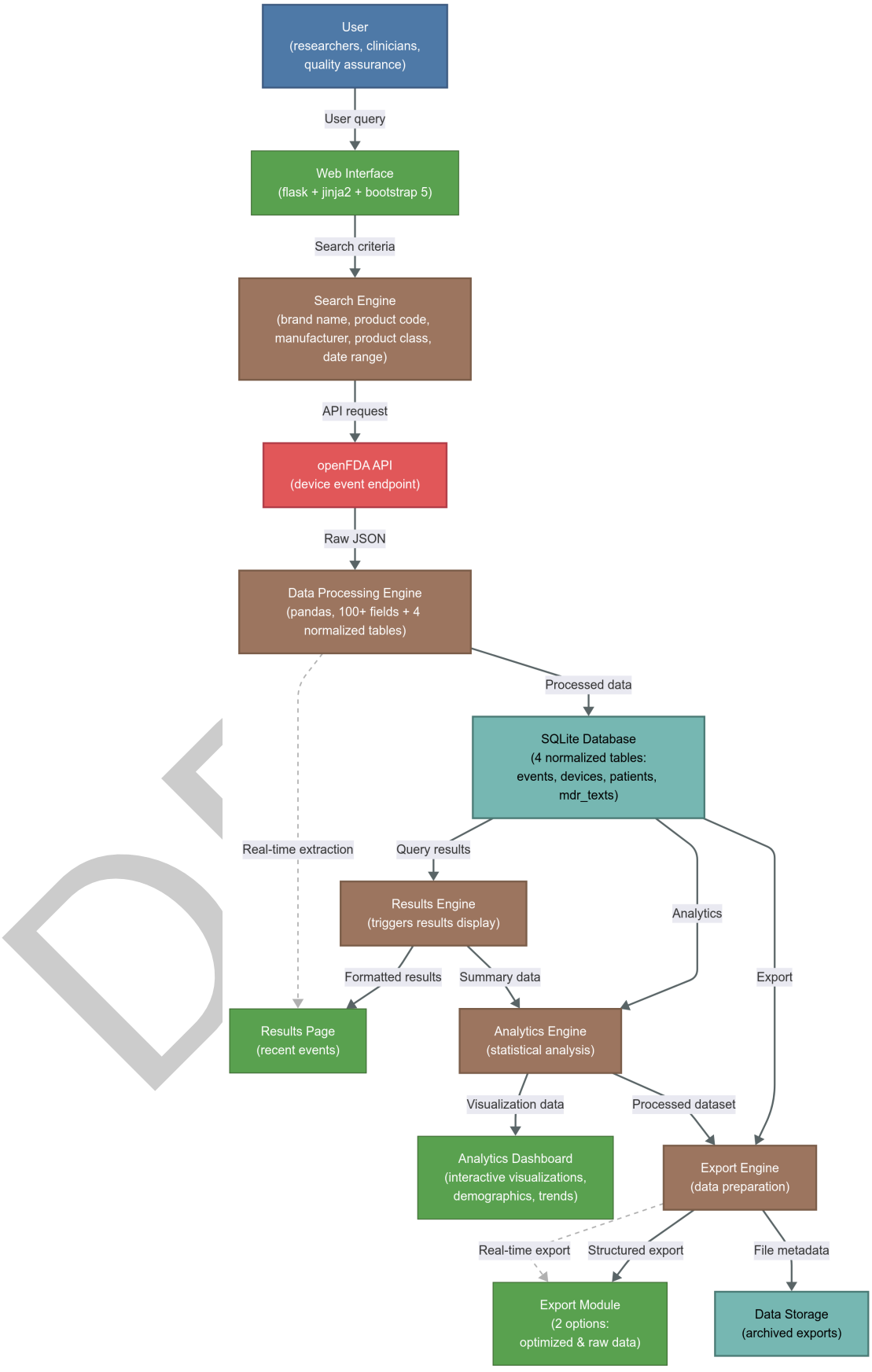


**Figure 1:** MAUDEMetrics Workflow

*Figure 1: MAUDEMetrics architecture and workflow. User input via the web interface triggers Flask endpoints that query the openFDA API (with pagination), cache results in SQLite, and process data with Pandas. The system supports two export modes: raw CSV files that preserve the original FDA JSON structure, and processed Excel workbooks (via OpenPyXL) with cleaned datasets and summary statistics. Docker ensures reproducible deployment.*

## Use Cases and Impact

**Research**:
- Aggregate device-specific adverse events for systematic reviews
- Compare event rates across manufacturers
- Conduct temporal trend analyses

**Clinical Quality Assurance**:
- Monitor devices used within an institution
- Detect emerging safety signals
- Generate reproducible safety reports

**Regulatory**:
- Support FDA submissions with structured evidence
- Standardize postmarket surveillance workflows

By automating extraction and standardizing outputs, MAUDEMetrics reduces manual effort, minimizes transcription errors, and enables reproducible device safety investigations.

## Software Availability

- **Name**: MAUDEMetrics
- **Repository**: https://github.com/MohamedMaroufMD/MAUDEMetrics
- **DOI**: https://doi.org/10.5281/zenodo.16691960
- **License**: Apache License 2.0
- **Version**: 2.1.0 (2025-09-08)

Detailed installation (Docker and manual), usage examples, tests, and sample exports are provided in the repository.

## References

Clark, C. (2023). *OpenPyXL: A python library for excel 2010 xlsx/xlsm files*. https://openpyxl.readthedocs.io/

Docker Inc. (2023). *Docker: Empowering app development for developers*. https://www.docker.com/

Flask Development Team. (2023). *Flask: A lightweight WSGI web application framework*. https://flask.palletsprojects.com/

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

McKinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 51–56. https://doi.org/10.25080/majora-92bf1922-00a

openFDA. (2025). *openFDA device event API*. https://open.fda.gov/apis/device/event/

SQLite Development Team. (2023). *SQLite: Self-contained, high-reliability SQL database engine*. https://www.sqlite.org/

U.S. Food and Drug Administration. (2025). *Manufacturer and user facility device experience (MAUDE)*. https://www.fda.gov/medical-devices/postmarket-requirements-devices/mandatory-reporting-requirements-manufacturers-importers-and-device-user-facilities