Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MAUDEMetrics: An Open-Source Framework for Automated FDA MAUDE Data Extraction and Reproducible Device Safety Research

Mohamed Marouf

marouf@std.mans.edu.eg

Mansoura University

Yousef Hawas

Tanta University

Ibraheem M. Alkhawaldeh

Mutah University

Alina Ghazou

Jordan University of Science and Technology

Aya Awad

Mansoura University

Ahmed Negida

Virginia Commonwealth University

Article

Keywords: postmarket surveillance, medical-device safety, FDA MAUDE database, reproducible research, open-source software, health informatics

Posted Date: June 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9990811/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

The FDA Manufacturer and User Facility Device Experience (MAUDE) database is the cornerstone of U.S. medical-device post-market surveillance, yet it remains underutilized: existing access routes impose record limits, demand programming expertise, and produce poorly reproducible queries — only about a quarter of published MAUDE studies can be reproduced. We present MAUDEMetrics, an open-source, no-code framework distributed as one-click desktop installers for macOS and Windows (with optional Docker and Python deployment) that automates the openFDA cursor-based (search_after) pagination mechanism, persists results in a local SQLite store, and applies a fixed, documented standardization pipeline to produce analysis-ready outputs. Unlike existing MAUDE tools, MAUDEMetrics couples automated end-to-end extraction with explicit gold-standard recall validation against authoritative FDA quarterly raw device files (five product codes spanning three orders of magnitude in volume, two calendar years; n = 281–29,737), achieving complete recall across all ten code-year combinations. Across three independent operators it extracted 10,000 records in 70 ± 17 s (≈143 records/second), and a feasibility case study retrieved 38,437 raw Impella adverse-event submissions in under four minutes. By delivering recall-validated, reproducible, transparently provenanced extraction without programming, MAUDEMetrics expands practical access to post-market device-surveillance research for clinicians, regulatory professionals, and informatics teams.

1. Background

Medical-device regulation differs fundamentally from pharmaceutical regulation. Whereas drugs require Phase I–III clinical trials demonstrating safety and efficacy before approval, most medical devices in the United States enter the market through the 510(k) clearance pathway, which requires manufacturers to demonstrate “substantial equivalence” to a legally marketed predicate device, often without new clinical trials [1]. Consequently, the safety and performance profile of many devices is incompletely characterized at market entry, and post-market surveillance is the primary mechanism for detecting safety signals [2, 3].

The FDA’s Manufacturer and User Facility Device Experience (MAUDE) database is the cornerstone of U.S. medical-device post-market surveillance [4]. As of the 2025 master Medical Device Report (MDR) data release, the MAUDE master record file contains more than 23 million reports of device-associated adverse events submitted by manufacturers, importers, device user facilities, and voluntary reporters [5]. These reports include detailed information on device malfunctions, patient injuries, deaths, and contextual narratives describing clinical circumstances. Systematic utilization of MAUDE for rigorous safety research, however, is constrained by significant methodological barriers. Researchers currently access MAUDE data through three primary methods, each with substantial limitations.

(1) FDA Web Portal (manual extraction).

The official FDA MAUDE web interface allows browser-based searches but imposes a hard limit of 500 records per export [6]. For longitudinal studies or high-volume device analyses requiring tens of

thousands of records, this necessitates repeated manual searches with adjusted date ranges, individual download of dozens or hundreds of separate files, manual extraction of demographic data for each report, file consolidation, and extensive cleaning. The web-portal export also provides only a subset of available MAUDE fields and omits clinically useful variables, including detailed narrative text fields and selected patient demographics that were only partially included in public exports prior to October 2023 [6]. Compiling and standardizing 10,000 events via this method requires roughly 20 separate searches followed by approximately 28 hours of intensive manual curation under optimal conditions.

(2) openFDA API (programmatic access).

The openFDA API provides programmatic access to MAUDE data in JavaScript Object Notation (JSON) format [7]. Although the API includes complete data fields and supports complex queries, it imposes a strict 1,000-record limit per call [8] and an architectural ceiling under which the sum of skip and limit cannot exceed 26,000 records, so naive indexing approaches fail on high-volume queries. openFDA documents a search_after paging mechanism, but its implementation requires advanced programming proficiency — parsing Link response headers, managing cursor state across iterative API calls, handling server timeouts, and integrating retry logic [9]. Ad-hoc scripts also lack standardization: each research group develops independent solutions with variable quality, inconsistent field mappings, and minimal documentation, severely undermining reproducibility [10, 11].

(3) Raw FDA quarterly database downloads.

The FDA distributes complete quarterly database snapshots as compressed archives (MDRFOI.zip) [5]. Working at this level requires 80 + GB of local storage for the complete historical dataset, database infrastructure to load and query pipe-delimited text files, extensive data engineering to map cryptic column codes to clinical terminology, and periodic manual updates. Although suitable for large-scale data warehousing, this approach is impractical for typical research workflows focused on specific devices or time periods.

These barriers contribute to a reproducibility gap in device informatics. The scale of the problem was recently quantified by Li and colleagues, who screened 523 MAUDE-based studies and found that only 336 reported queries that could even be executed; of those executable queries, only 23.3% reproduced the originally reported record count within a pre-specified 0.95–1.05 reproducibility-coefficient threshold [12]. Standardized, transparent extraction tooling — with explicit query parameters, fixed field mappings, and shareable artifacts — directly addresses this gap. To our knowledge, no existing MAUDE tool meets these requirements: available solutions are either unmaintained classification or ZIP-wrangling scripts that require programming and perform no validated extraction, or web viewers constrained by the same API limits, and none reports recall validation against an authoritative FDA reference or returns standardized, analysis-ready output. MAUDEMetrics is, therefore, to our knowledge, the first openly released MAUDE framework to integrate automated full-dataset extraction with explicit gold-standard recall validation and a deterministic standardization pipeline — directly targeting reproducibility, the failure mode quantified by Li and colleagues, rather than convenience alone. It eliminates pagination

barriers through automated API orchestration capable of retrieving full datasets, validates retrieved records against FDA gold-standard sources for the tested scope, and transforms raw API responses into analysis-ready outputs. The framework is distributed as a downloadable desktop application that requires no programming skills at the point of use, while preserving Docker and Python source-code deployment as advanced-user options. This manuscript describes the technical architecture, validation methodology, and performance benchmarking of MAUDEMetrics. As the medical-device landscape grows in complexity — particularly with AI-enabled and software-as-medical-device products — accessible and validated tools for MAUDE data analysis are increasingly important for early detection of adverse events, identification and correction of device-related issues, and the generation of clinically actionable evidence.

2. Methods

MAUDEMetrics is designed as a modular application following separation-of-concerns principles to ensure maintainability, testability, and extensibility (Fig. 1). The system comprises four core components.

2.1 Orchestration Engine

The backend is implemented in Flask (Python 3.11+), chosen for its minimal dependencies and robust routing capabilities [13]. The orchestration engine manages API state — query parameters, pagination state, and error recovery — handles asynchronous request processing so that concurrent API calls do not block the user interface, and maintains session persistence by retaining query history and cached results across user sessions.

The primary function of this component is a fully automated implementation of the openFDA search_after paging mechanism [9], the API's own documented solution for traversing result sets exceeding the 26,000-record skip/limit ceiling. The engine executes an iterative loop that parses Link response headers to dynamically construct the next query URL using cursor-based offsets, allowing the framework to traverse an unbounded number of records without hitting the 26,000-record hard-stop, and aggregates all results into a unified dataset. The system also implements an exponential-backoff retry mechanism to handle HTTP 429 rate-limit responses and transient network interruptions. Additional details on openFDA query construction and pagination orchestration are provided in Supplementary File 1.

MAUDEMetrics interacts only with publicly released FDA resources via the official openFDA API and publicly available MAUDE data files. The framework respects published API access limits, supports an optional FDA-issued API key, and persists results locally to minimize repeated server requests. Use of openFDA is governed by the FDA's open-data terms; no proprietary or restricted content is accessed, and no authentication-protected endpoints are scraped.

All data retrieved through MAUDEMetrics are released by the FDA under the Creative Commons CC0 1.0 Universal dedication, which places the data in the public domain and permits unrestricted use, reproduction, and redistribution without attribution. The openFDA platform was established by the FDA specifically to enable programmatic access to its public datasets by researchers, developers, and the public (https://open.fda.gov/terms/). The cursor-based pagination technique (search_after) employed by MAUDEMetrics is an officially documented openFDA API feature, not a circumvention of access controls.

The core pagination logic is summarized below; the complete implementation, including exponential backoff, HTTP 429 handling, and API-key injection, is available in the public repository [14]:

next_url = build_initial_query(base_query, limit = 1000, sort="date_received:desc")

all_records = []

while next_url:

response = fetch_with_retry(next_url) # exponential backoff on failure

all_records.extend(response.json()["results"])

next_url = extract_link_header(response) # None when last page reached

return all_records

2.2 Data Persistence Layer

To minimize redundant load on FDA servers and enable offline analysis, retrieved data are persisted to a local SQLite database [15]. The schema includes normalized tables for events, devices, patients, and MDR text narratives, reflecting the nested structure of FDA JSON responses while enabling Structured Query Language (SQL) queries. Each new query initializes a clean database state to ensure result integrity and eliminate stale-data contamination between sessions: the system checks a preview record count against the live API before extraction and replaces any prior results with the current query output. SQLite was selected over document databases such as MongoDB or alternatives such as PostgreSQL for its zero-configuration deployment, single-file portability, and native SQL query capabilities; the entire database is contained within a single portable file (fda_data.db).

2.3 Data Standardization Pipeline

Raw FDA API responses contain over 100 nested fields with cryptic naming conventions and inconsistent data types. The standardization pipeline, implemented with Pandas [16] and NumPy [17], comprises four sequential, deterministic steps:

(1) Field humanization. API field names are mapped to clinical terminology, nested JSON structures are flattened to tabular columns, and categorical values are standardized against a controlled vocabulary.

(2) Demographic aggregation. Nested patient arrays are parsed, per-report descriptive statistics are computed (median age, weight, sex, race, and ethnicity), and multi-valued fields are explicitly handled.

(3) Date normalization. YYYYMMDD string timestamps from the API are converted to ISO 8601 dates; missing or malformed dates are coerced to explicit nulls.

(4) Narrative integration. Multi-part MDR narrative fields are concatenated into unified narrative columns, and Unicode artifacts and stray special characters are sanitized.

Field mappings, normalization rules, and the controlled vocabulary are specified in Supplementary File 2.

2.4 Export Generation System

The system provides three complementary export modes generated via OpenPyXL [18]. The first is an optimized Excel workbook (.xlsx) containing cleaned, flattened event-level data with integrated narratives, formatted columns, and professional styling. The second is a standalone summary-statistics workbook providing rapid overview of patient demographics, event-type frequencies, and device-problem taxonomies. The third is a raw-data ZIP archive containing five comma-separated-values (CSV) files that preserve the original FDA JSON structure (core event metadata, device characteristics, de-identified patient data, unprocessed narratives, and complete API responses) for transparency and downstream analysis flexibility.

2.5 Validation Methodology

Three independent validation protocols were conducted to characterize MAUDEMetrics.

Protocol 1: Gold-standard data completeness.

Gold-standard comparison was performed across five product codes spanning three orders of magnitude in report volume and four device categories for two non-consecutive calendar years (2023 and 2025): KYF (Ex-Press Glaucoma Filtration Device, ophthalmic), IYO (Ultrasound Imaging System, diagnostic imaging), DQY (Catheter, Percutaneous, cardiovascular), MNT (Ventilator, Continuous, Facility Use, critical care), and OZO (Artificial Pancreas Device System, metabolic/endocrine). For each year, the corresponding FDA quarterly raw device file (DEVICE2023.txt and DEVICE2025.txt) was downloaded and filtered for each product code, and the count of unique MDR Report Keys was recorded as the reference standard. An identical product-code query covering the corresponding full calendar year was executed in MAUDEMetrics for each code-year combination, and recall was computed as the ratio of records retrieved by the framework to records present in the raw file. For the primary validation code (KYF) in 2023, individual MDR Report Keys were additionally matched between the raw file and the API response to confirm zero missing records at the key level.

Protocol 2: Performance benchmarking.

Extraction performance was benchmarked across dataset sizes of 1,000, 5,000, and 10,000 records. Three methods were compared: (i) manual extraction via the FDA web portal, scaled from per-batch operations under the 500-record export ceiling and including per-report manual demographic extraction; (ii) a custom Python web scraper performing iterative API calls with downstream parsing; and (iii) MAUDEMetrics, timed end-to-end from the user clicking “Extract MAUDE Data” to the availability of the optimized download. All validation and benchmarking reported here were performed with MAUDEMetrics version 3.1.1 (archived at https://doi.org/10.5281/zenodo.16691960), running on Python 3.11 with pinned dependencies (Flask 2.3.3, pandas 1.5.3, NumPy 1.23.5, requests 2.28.1, openpyxl 3.1.2; full lock file in the repository). The MAUDEMetrics tests were performed independently by three operators on standard consumer hardware (Apple M2, 8 GB RAM; Intel Core i7-7820HQ at 2.9 GHz, 16 GB RAM) using a free FDA API key (https://open.fda.gov/apis/authentication/), enabling the maximum batch size of 1,000 records per call.

Performance is reported primarily as throughput (records processed per second) for each method, because throughput is a property of the method itself and is independently verifiable, whereas a speedup ratio is a derived quantity whose precision cannot exceed the precision of its inputs. Mean and standard deviation across operators are reported for MAUDEMetrics; manual and web-scraper throughputs are derived from per-batch operation timings rather than replicated trials, because exhaustive triplicate measurement of a multi-hour manual workflow was not practically feasible. Formal inferential testing (e.g., one-way ANOVA or Kruskal–Wallis) was deliberately not performed: applying inferential statistics to a comparison between a triplicate-measured method and two single-operator scaled estimates would convey spurious precision. Where a between-method comparison is made, it is expressed as an order-of-magnitude statement rather than a precise fold-change.

Protocol 3: Applied case study.

An applied case study illustrating end-to-end utility on a large dataset was conducted on the Impella device. Adverse-event reports filed under the brand name “impella” were retrieved with an open date range up to 25 March 2026, capturing all event types (malfunction, injury, death). The Impella case is presented as a demonstration of feasibility on high-volume queries, not as an epidemiological estimate; raw MDR submissions are reported without deduplication so that follow-up and per-component reports remain visible.

3. Results

3.1 Protocol 1: Gold-Standard Data Completeness

Across all five validated product codes and both validated calendar years (2023 and 2025), MAUDEMetrics achieved 100% recall relative to the FDA quarterly raw device files (Table 1). In 2023, record volumes ranged from 314 (KYF) to 29,737 (OZO); in 2025, from 281 (KYF) to 3,652 (OZO), reflecting year-to-year variation in reporting volume across device types. For the primary validation code

(KYF) in 2023, key-level matching confirmed that all 314 MDR Report Keys present in the raw file were individually retrievable, with zero missing records. For all remaining code-year combinations, recall was computed from the records retrieved by MAUDEMetrics; in all cases MAUDEMetrics retrieved a count equal to or exceeding the raw-file count. In six of ten validated combinations the counts were identical; in four (OZO 2023, DQY 2025, MNT 2025, and OZO 2025) MAUDEMetrics retrieved between one and two additional records relative to the quarterly snapshot, consistent with routine post-snapshot indexing of recently processed reports. These results confirm that the cursor-based search_after pagination algorithm achieves complete data fidelity across a clinically diverse set of device types and two non-consecutive calendar years.

Table 1
Gold-standard recall validation across five product codes and two calendar years (DEVICE2023.txt and DEVICE2025.txt).

Year	Code	Device	Category	Raw file (n)	MAUDEMetrics (n)	Recall
2023	KYF	Ex-Press Glaucoma Filtration Device	Ophthalmic	314	314	100.0%
2023	IYO	Ultrasound Imaging System	Diagnostic Imaging	527	527	100.0%
2023	DQY	Catheter, Percutaneous	Cardiovascular	3,049	3,049	100.0%
2023	MNT	Ventilator, Continuous (Facility)	Critical Care	4,565	4,565	100.0%
2023	OZO	Artificial Pancreas Device System	Metabolic/Endocrine	29,737	29,739†	100.0%
2025	KYF	Ex-Press Glaucoma Filtration Device	Ophthalmic	281	281	100.0%
2025	IYO	Ultrasound Imaging System	Diagnostic Imaging	492	492	100.0%
2025	DQY	Catheter, Percutaneous	Cardiovascular	3,093	3,095†	100.0%
2025	MNT	Ventilator, Continuous (Facility)	Critical Care	3,585	3,587†	100.0%
2025	OZO	Artificial Pancreas Device System	Metabolic/Endocrine	3,652	3,653†	100.0%

† In these combinations MAUDEMetrics retrieved between one and two additional records relative to the quarterly raw file (OZO 2023: +2; DQY 2025: +2; MNT 2025: +2; OZO 2025: +1), consistent with routine post-

snapshot indexing of recently processed reports — indicating the tool captures more current data than the quarterly snapshot. KYF 2023 was additionally validated by key-level matching of all 314 individual MDR Report Keys, confirming zero missing records.

3.2 Protocol 2: Performance Benchmarking

Performance was evaluated across three dataset sizes representing typical research scenarios, namely 1,000, 5,000, and 10,000 records. Across three independent operators, MAUDEMetrics completed end-to-end extraction in 17 ± 2 s for 1,000 records, 44 ± 11 s for 5,000 records, and 70 ± 17 s for 10,000 records (mean ± SD), corresponding to throughputs of approximately 59, 114, and 143 records per second, respectively (Table 2, Figs. 2 and 3). Throughput rises with batch size because per-query setup costs (connection establishment, initial cursor negotiation) are amortized over a larger number of records. For context, applying a 10-second-per-report manual extraction rate — a direct time-and-motion observation drawn from the co-author team's hands-on experience conducting prior MAUDE-based studies [19, 20] — a manual workflow corresponds to a throughput of approximately 0.1 records per second, and a comparison custom Python web-scraper to approximately 0.4 records per second. We report throughput as the primary metric rather than a precise speedup ratio because throughput is a property of each method that can be verified independently, whereas a ratio multiplies the uncertainty of two estimates (see Methods). On an order-of-magnitude basis, MAUDEMetrics throughput exceeds the manual rate by approximately three orders of magnitude across all tested dataset sizes; because the manual rate is an internal observation rather than an externally replicated measurement, we present this difference as an order-of-magnitude statement rather than a precise fold-change.

Table 2
Throughput comparison across MAUDE access methods.

Dataset size	MAUDEMetrics time (mean ± SD)	MAUDEMetrics throughput	Web-scraper throughput‡	Manual throughput†
1,000 records	0:00:17 ± 0:00:02	≈ 59 records/sec	≈ 0.4 records/sec	≈ 0.1 records/sec
5,000 records	0:00:44 ± 0:00:11	≈ 114 records/sec	≈ 0.4 records/sec	≈ 0.1 records/sec
10,000 records	0:01:10 ± 0:00:17	≈ 143 records/sec	≈ 0.4 records/sec	≈ 0.1 records/sec

MAUDEMetrics times are mean ± SD (H:MM:SS) across three independent operators; throughput = records processed / wall-clock seconds. † Manual throughput is a scaled estimate derived from per-batch operations (search + export + save ≈ 11 s; between-batch setup ≈ 13 s; merge ≈ 27 s; manual demographic extraction ≈ 10 s per report). The 10-s/report figure is based on direct co-author experience accumulated during the conduct of multiple prior MAUDE-based studies [19, 20]; *independent replication by external investigators is identified as a primary benchmarking limitation. ‡ Web-scraper throughput is derived from execution and JSON parsing alone, excluding one-time development cost (typically 2–4 hours) and the upstream MAUDE web search needed to identify export URLs. Manual and web-scraper values are scaled*

single-operator estimates rather than replicated measurements; readers should compare them to MAUDEMetrics on an order-of-magnitude basis rather than as point estimates.

Several additional observations emerged. Repeat queries for identical parameters returned within one second from the local SQLite store, effectively eliminating redundant API load. End-to-end extraction-and-export for 10,000 records completed in 70 ± 17 s. No API rate limiting, throttling, or IP blocks were encountered across up to 50 consecutive queries, demonstrating robustness of the extraction pipeline under sustained use.

Table 3
Qualitative feature comparison across MAUDE data-access methods.

Feature	FDA Web Portal	Custom Python API script	MAUDEMetrics	Raw FDA quarterly files
Throughput (10K records)	≈ 0.1 records/sec (estimated)	≈ 0.4 records/sec (estimated)	≈ 143 records/sec (mean, n = 3 operators)	Internet- and disk-dependent
Pagination handling	500-record manual limit	Requires custom iteration	Automated cursor-based fetch	None (file-based)
Output data format	Partial CSV (missing fields)	Raw JSON / CSV	Curated Excel + summary-stats Excel + raw CSV	Raw pipe-delimited text
Patient demographics	Manual extraction needed	Requires custom parsing	Auto-aggregated	Available; requires parsing
Standardization	None	Variable by script	Fixed automated pipeline	Variable
Reproducibility	Highly operator-dependent	Variable by script	Containerized + pinned pipeline	Analysis-dependent
Required expertise	None	Advanced programming	None (GUI)	Advanced data handling
Installation	None (web-based)	Environment setup required	One-click .dmg / .exe; optional Docker	Web download + local unzip

Manual and web-scraper throughputs are scaled estimates with operator and network variability; only MAUDEMetrics throughput is a replicated measurement. See Table 2 *for replicated MAUDEMetrics timings and Methods for the rationale for reporting throughput rather than a precise speedup ratio.*

To visualize multidimensional performance differences, the methodological capabilities of the three primary access methods are mapped onto a radar chart (Fig. 4), with a qualitative feature comparison provided in Table 3. MAUDEMetrics consistently maximized extraction speed, data completeness,

standardization, reproducibility, and accessibility, whereas manual and scraper methods required substantial trade-offs.

3.3 Protocol 3: Applied Case Study — Impella Device

MAUDEMetrics was queried with brand name “impella” and an open date range to retrieve all available historical records up to 25 March 2026. A total of 38,437 raw MDR submissions were retrieved, with extraction completed in 2 min 36 s and export generation in 45 s (total end-to-end time, 3 min 21 s).

MAUDEMetrics completed query formulation in approximately 10 s and end-to-end extraction-and-export in 3 min 21 s, yielding a total time-to-insight of well under 4 min and a wall-clock throughput of approximately 191 records per second across the 38,437-submission dataset. For comparison, a scaled manual workflow combining portal extraction (estimated 1 h 5 min) and per-report demographic aggregation across 38,437 submissions at the 10-second-per-report rate — a direct time-and-motion observation drawn from the co-author team's hands-on experience conducting prior MAUDE-based studies [19, 20] — would require approximately 107.8 h, corresponding to a manual throughput of approximately 0.1 records per second. The two throughputs differ by approximately three orders of magnitude. We frame this comparison as an order-of-magnitude statement rather than a precise fold-change because the manual per-report rate is based on direct co-author experience consistently accumulated across [19, 20] and has not been independently replicated (see Limitations) (Fig. 5).

Table 4
Most frequently reported device problems for the Impella case study.

Device problem	Frequency	Percentage
Restricted flow rate	2,842	6.3%
Device sensing problem	2,554	5.6%
Patient–device incompatibility	2,196	4.8%
Pumping stopped	1,681	3.7%
Positioning failure	1,537	3.4%
Failure to advance	1,376	3.0%
Application-program problem	1,310	2.9%
Increase in pressure	1,243	2.7%
Malposition of device	939	2.1%
Fluid/blood leak	891	2.0%

Counts represent raw MDR submissions inclusive of follow-up reports and per-component submissions required by FDA reporting guidance; percentages are descriptive of the raw report stream and should not be interpreted as event incidence. No automated deduplication was applied; researchers performing epidemiological inference should apply study-specific deduplication logic.

Table 5
Most frequently reported patient outcomes for the Impella case study.

Patient problem	Frequency	Percentage
Hemorrhage / blood loss / bleeding	8,029	16.3%
Ischemia	3,728	7.6%
Hemolysis	3,291	6.7%
Hematoma	2,319	4.7%
Thrombosis / thrombus	2,203	4.5%
Tachycardia	1,403	2.9%
Arrhythmia	1,059	2.2%
Stroke / CVA	940	1.9%
Thrombocytopenia	776	1.6%
Ventricular fibrillation	676	1.4%

Counts and percentages are calculated against the raw submission stream (n = 38,437) inclusive of follow-up and per-component reports. Percentages should not be interpreted as event incidence. Researchers performing epidemiological inference should apply study-specific deduplication logic.

This case study is a feasibility demonstration (Tables 4 and 5, Fig. 6). The leading patient-level harms recovered — hemorrhage, ischemia, and hemolysis — and the leading device-level failure modes — restricted flow rate and sensing malfunction — are consistent with the established risk profile of percutaneous mechanical circulatory support, serving as a face-validity check that the extraction and standardization pipeline faithfully reproduces known signals. The contribution is operational: MAUDEMetrics rendered the complete 38,437-submission raw report stream extractable, standardized, and analysis-ready within minutes — enabling full-history, multi-code surveillance workflows (for example, feeding a clinical quality-assurance dashboard) that are impractical to assemble manually. Counts are raw MDR submissions and must not be interpreted as event incidence given the absence of deduplication. The complete extended device- and patient-problem taxonomies and the full optimized dataset and summary statistics are available in Supplementary File 3.

This case study demonstrates that MAUDEMetrics enables rapid, comprehensive device-safety assessments suitable for clinical quality-assurance dashboards, regulatory submissions requiring post-

market data, academic systematic reviews, and post-market device-safety surveillance programs.

4. Discussion

MAUDEMetrics directly addresses three fundamental methodological barriers that have constrained rigorous FDA medical-device post-market surveillance research: pagination, standardization, and accessibility.

Elimination of pagination constraints.

MAUDEMetrics automates the openFDA search_after paging mechanism [9] to bypass both the 1,000-record per-call limit and the 26,000-record absolute indexing ceiling encountered with conventional skip/limit approaches. The implementation — Link header parsing, cursor management, and retry orchestration — places the operational complexity inside the framework rather than on the user, enabling seamless extraction regardless of dataset volume (Fig. 7).

Validation scope and generalizability.

MAUDEMetrics is designed to operate on any MAUDE product code, brand name, or manufacturer search. Gold-standard recall validation was performed across five product codes and two non-consecutive calendar years (2023 and 2025), covering report volumes from 281 to 29,737 records and four device categories, with 100% recall achieved in all ten code-year combinations (Table 1). For Impella (a high-volume cardiovascular device, n = 38,437), the framework demonstrated end-to-end feasibility in minutes versus a scaled manual workflow estimated at ≈ 107.8 h, but this case study is presented as an applied feasibility demonstration rather than a gold-standard validation result.

Data standardization and reproducibility.

The lack of standardized MAUDE extraction methodology has historically contributed to inconsistent field definitions across studies. MAUDEMetrics implements a fixed, documented standardization protocol in which demographic fields are mapped to a controlled vocabulary, dates are normalized to ISO 8601, and missing data are explicitly represented as null. Containerized and signed-binary distribution ensures that the same standardization pipeline executes identically across institutions (Fig. 8); researchers running identical MAUDEMetrics queries will obtain identical datasets, which is not achievable with ad-hoc scripting.

Accessibility for non-programmers.

Health informatics workforce analyses document growing informatics staffing demands and identify programming as a specialist competency not universally held among clinical researchers [21]. MAUDEMetrics is now distributed as a downloadable desktop application for macOS (Apple Silicon and Intel) and Windows (10/11, 64-bit) via GitHub Releases, with built-in update checks. End users can install the application with a single click and execute extractions through a graphical interface without any command-line interaction, environment configuration, or programming knowledge. Docker and Python

source-code deployments remain available for advanced users who require containerized or scripted workflows. We therefore characterize MAUDEMetrics as no-code at the point of use for end users, while recognizing that advanced deployment routes still presume basic familiarity with Docker or Python.

Comparison with existing tools.

Several groups have developed FDA MAUDE-related tools. HazyResearch ICIJ-MAUDE [10] is a weakly supervised classification pipeline that requires Python expertise, has no graphical interface, offers limited documentation, and is hosted in an archived repository that is no longer actively maintained. autonlab/fda_maude [11] provides scripts to download and wrangle quarterly ZIP files; it requires programming expertise, performs no API pagination, and leaves data cleaning to the user. Third-party openFDA web explorers (e.g., openFDA Explorer) support exploratory visualization but inherit the 1,000-record API ceiling and provide limited export functionality for downstream analysis. We additionally assessed the sheetalkalburgi/web-scraping repository and judged it not directly relevant: it scrapes the FDA Code of Federal Regulations from a separate FDA portal and does not interact with MAUDE adverse-event records. Against this landscape, MAUDEMetrics differentiates itself by combining a downloadable GUI desktop application, an automated cursor-based pagination engine that bypasses the 26,000-record skip/limit ceiling, a fixed standardization pipeline, three export modes (event-level Excel, summary-statistics Excel, and raw CSV), and gold-standard recall validation for the tested scope. A summary feature comparison is provided in Table 6.

Table 6
Feature comparison of FDA MAUDE-related tools.

Feature	HazyResearch ICIJ-MAUDE	autonlab/fda_maude	openFDA Explorer (3rd-party)	MAUDEMetrics
Primary function	Weakly supervised AE classification	ZIP download / wrangling scripts	Web-based AE visualization	End-to-end extraction + standardization + export
API pagination handling	None (no API use)	None (no API use)	Subject to 1,000-record limit	Automated cursor-based fetch
No-code GUI	No	No	Partial (web UI)	Yes (.dmg / .exe / GUI)
Automated data standardization	Partial (classification labels)	None (user-defined)	None	Fixed automated pipeline
Validated recall against raw FDA file	No	No	No	Yes (five product codes, two calendar years)
Actively maintained	No (archived)	Limited	Variable	Yes (semantic versioning)
Reproducible deployment	Environment setup required	Environment setup required	Web only	One-click installer + Docker

Duplicate-report handling.

The MAUDE database contains duplicate reports: a single adverse event may generate multiple MDR submissions through follow-up reports, mandatory per-component reporting, or manufacturer updates [6]. MAUDEMetrics deliberately preserves all retrieved submissions in its exports rather than imposing an automated deduplication rule, because the optimal deduplication logic is study-specific and frequently depends on fields (such as patient identifiers and event-date precision) that are partially redacted in the public MAUDE release. Researchers performing epidemiological inference should apply context-appropriate deduplication — for example, collapsing follow-ups onto initial reports or restricting to unique manufacturer report numbers within a defined window — in their downstream analysis. Accordingly, the Impella tables in this manuscript (Tables 4 and 5) are reported as raw MDR submission counts inclusive of follow-up and per-component duplicates, and percentages in those tables should not be interpreted as event incidence.

4.1 Limitations

MAUDEMetrics solves the data-access problem but cannot address the intrinsic quality limitations of passive surveillance systems [4]. Underreporting is a fundamental constraint: voluntary reporting

captures only a fraction of adverse events; significant transparency gaps in device safety reporting have been documented [22]; and reported event rates cannot be interpreted as true incidence. The absence of denominator data — implantation volumes and procedure counts — precludes rate calculations and requires that absolute event counts be interpreted alongside external epidemiological data. Variable report quality also limits inference: narratives range from highly detailed clinical descriptions to brief manufacturer submissions, and patient age is absent in 30–40% of reports, restricting subgroup analyses. Reporting bias is an additional limitation: high-profile devices receive disproportionate investigation through ascertainment bias and newly marketed devices have artificially low event counts owing to usage lag. MAUDEMetrics retrieves data from the openFDA API, which may exhibit indexing delays relative to raw FDA file releases; researchers conducting time-sensitive surveillance on very recent data should verify API currency against quarterly raw MDRFOI file releases when absolute completeness is required.

MAUDEMetrics facilitates access to MAUDE data but does not transform it into a gold-standard epidemiological resource.

The benchmarking assumption of approximately 10 s per report for manual demographic extraction is derived from time-and-motion observations accumulated by the co-author team across multiple collaborative MAUDE-based studies, with supporting references [19, 20]. This estimate reflects consistent hands-on experience across device categories and is therefore not solely dependent on a single publication. Nonetheless, it has not been independently verified by external investigators, and per-report processing time may vary across settings depending on report complexity, operator experience, and institutional workflow. Third-party replication of this estimate across diverse clinical research environments would further strengthen the benchmarking conclusions presented here.

Benchmarking variability is a further limitation. Replicated trials were performed only for MAUDEMetrics (three operators per dataset size); manual and web-scraper throughputs are scaled estimates derived from per-batch operation times rather than independent replicates, which is why we deliberately did not perform formal inferential statistical testing across methods and why we report throughput per method rather than a precise speedup ratio. Comparisons between methods should be interpreted as order-of-magnitude statements rather than statistical effect sizes. Independent multi-site benchmarking with replicated trials of all three methods would substantially strengthen these conclusions.

MAUDEMetrics does not deduplicate MAUDE submissions, by design. Frequencies and percentages reported in any case study — including the Impella analysis above — represent raw MDR submission counts and may include follow-up reports and per-component submissions. We did not perform a sensitivity analysis applying a particular deduplication rule (for example, same manufacturer report number, same event date, same device component) because reliable patient-level deduplication is constrained by FDA redactions in the public release. Investigators using MAUDEMetrics for epidemiological inference should apply context-appropriate deduplication and report the rule explicitly.

4.2 Future Directions

Several directions are planned or under consideration for future development of MAUDEMetrics. As the nearest-term deployment priority, a hosted web interface is planned to enable fully browser-based access without local installation, which would eliminate the remaining setup barrier for clinical researchers without institutional IT support and complement the existing downloadable desktop application. Beyond accessibility, three substantive research directions are being pursued. First, we plan to evaluate the integration of natural-language-processing approaches — including large-language-model-assisted entity extraction and semantic search — over the mdr_text narrative fields, which currently represent an underutilized source of structured clinical signal. Second, the inclusion of post-market signal-detection methods, such as proportional reporting ratios, time-series changepoint analysis, and comparative device hazard ratios, is being scoped to move MAUDEMetrics from data extraction toward active post-market device-safety signal-detection functionality. Third, we intend to explore record linkage with adjacent public resources, including the FDA 510(k) premarket clearance database, Centers for Medicare & Medicaid Services (CMS) claims data for denominator estimation, and PubMed-indexed literature, to enable richer contextualization of adverse-event signals. Each of these directions remains under feasibility evaluation rather than active deployment; preliminary work has not yet generated reportable results.

5. Conclusions

MAUDEMetrics is a downloadable, GUI-driven, open-source framework for FDA MAUDE post-market surveillance. By integrating cursor-based API pagination, a fixed standardization pipeline, multi-format export, and one-click desktop installation, it lowers the technical barrier to large-scale, transparent device-safety analyses. Across ten code-year combinations spanning five product codes and two calendar years (2023 and 2025), the framework achieved complete recall against authoritative FDA quarterly raw files, processed 10,000 records at approximately 143 records per second (mean across three independent operators) — exceeding an internal time-and-motion manual baseline by approximately three orders of magnitude — and supported a feasibility analysis of 38,437 raw Impella submissions in under four minutes. Its central contribution is recall-validated, reproducible extraction rather than speed alone. We invite the research community, regulatory professionals, and device manufacturers to use MAUDEMetrics for independent, evidence-based safety investigations and to contribute to its open-source development.

Abbreviations

Abbreviation	Full term
API	Application Programming Interface
CMS	Centers for Medicare & Medicaid Services
CSV	Comma-Separated Values
FDA	U.S. Food and Drug Administration
GUI	Graphical User Interface
JSON	JavaScript Object Notation
LLM	Large Language Model
MAUDE	Manufacturer and User Facility Device Experience
MDR	Medical Device Report
MDRFOI	Medical Device Reports Freedom of Information
NLP	Natural Language Processing
PRR	Proportional Reporting Ratio
SD	Standard Deviation
SQL	Structured Query Language

Declarations

Data Availability

The source code and binaries for MAUDEMetrics are publicly available at https://github.com/MohamedMaroufMD/MAUDEMetrics, with a persistent identifier assigned via DOI (https://doi.org/10.5281/zenodo.16691960). The software is released under the Apache License 2.0. Pre-built desktop installers are distributed via GitHub Releases (https://github.com/MohamedMaroufMD/MAUDEMetrics/releases/latest) for macOS (Apple Silicon and Intel .dmg) and Windows (NSIS .exe). Optional Docker and Python 3.11+ deployment routes are documented in the repository README. A complete step-by-step deployment tutorial with screenshots is provided in Supplementary File 4. Minimum system requirements: 2 GB RAM (8 GB recommended for >20,000-record queries) and an active internet connection.

FAIR compliance. MAUDEMetrics adheres to the FAIR principles: it is Findable (a versioned, citable Zenodo DOI), Accessible (open Apache 2.0 source plus signed pre-built installers via GitHub Releases), Interoperable (outputs in open, non-proprietary formats — CSV, ISO 8601 dates, and .xlsx — with fields mapped to FDA-native terminology), and Reusable (a documented standardization protocol with field-mapping and controlled-vocabulary specifications provided in Supplementary File 2, pinned dependency

versions, a permissive Apache 2.0 license, and fully open source code enabling independent reproduction without development-team support). MAUDE Data Source: openFDA API (https://open.fda.gov/apis/); data licensed under Creative Commons CC0 1.0 Universal (public domain dedication); no proprietary or restricted data accessed; access governed by FDA openFDA Terms of Service (https://open.fda.gov/terms/).

Ethics Approval and Consent to Participate

Not applicable. This study utilized publicly available, de-identified secondary data released by the FDA under the Creative Commons CC0 1.0 Universal public-domain dedication; no human-subjects research requiring institutional review-board approval or informed consent was conducted, and no attempt was made to re-identify individuals. Users are advised that adverse-event narratives may contain residual indirect identifiers and should be handled in accordance with applicable institutional data-governance policies.

Consent for Publication

Not applicable.

Competing Interests

MM is the primary developer of MAUDEMetrics, the tool described in this manuscript. To ensure objectivity, data-completeness validation was assessed against an external, independent reference standard (FDA quarterly raw device files) rather than developer judgment, performance testing was independently conducted by co-authors (YH and IA), and the complete source code, controlled-vocabulary mappings, and case-study datasets are publicly released under an open license to enable fully independent third-party replication. MM holds no financial, commercial, or proprietary interest in MAUDEMetrics, which is distributed free of charge under the Apache License 2.0. All other authors declare no competing interests. This research received no commercial funding or sponsorship from medical-device manufacturers.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Clinical Trial Registration

Not applicable. This study did not involve a prospective clinical trial of a health intervention.

Authors' Contributions

MM: Conceptualization, Software Development, Methodology Design, Validation, Data Analysis, Manuscript Writing (Original Draft and Revisions), Visualization. YH: Methodology Design, Validation,

Performance Testing, Visualization, Manuscript Writing, Review, and Editing. IA: Methodology Design, Validation, Performance Testing, Review, and Editing. AG and AA: Introduction Writing, Review, and Editing. AN: Supervision, Review, and Editing. All authors read and approved the final manuscript.

Acknowledgments

The authors thank the openFDA team at the U.S. Food and Drug Administration for providing free programmatic access to MAUDE data, the open-source community contributors to Flask, Pandas, OpenPyXL, and SQLite, and the beta testers who assisted with usability evaluation and performance benchmarking.

References

1. Institute of Medicine (US) Committee on the Public Health Effectiveness of the FDA 510(k) Clearance Process. Medical Devices and the Public's Health: The FDA 510(k) Clearance Process at 35 Years. National Academies Press; 2011. doi:10.17226/13020.
2. Everhart AO, Karaca-Mandic P, Redberg RF, Ross JS, Dhruva SS. Late adverse event reporting from medical device manufacturers to the US Food and Drug Administration: cross sectional study. BMJ. 2025;388:e081518. doi:10.1136/bmj-2024-081518.
3. Zuckerman DM, Brown P, Nissen SE. Medical device recalls and the FDA approval process. Arch Intern Med. 2011;171(11):1006–1011. doi:10.1001/archinternmed.2011.30.
4. U.S. Food and Drug Administration. About Manufacturer and User Facility Device Experience (MAUDE) Database. Available at: https://www.fda.gov/medical-devices/mandatory-reporting-requirements-manufacturers-importers-and-device-user-facilities/about-manufacturer-and-user-facility-device-experience-maude-database
5. U.S. Food and Drug Administration. MDR Data Files. Available at: https://www.fda.gov/medical-devices/medical-device-reporting-mdr-how-report-medical-device-problems/mdr-data-files
6. Ensign LG, Cohen KB. A primer to the structure, content and linkage of the FDA's Manufacturer and User Facility Device Experience (MAUDE) files. eGEMs. 2017;5(1):12. doi:10.5334/egems.221.
7. openFDA. About the openFDA API. Available at: https://open.fda.gov/apis/
8. openFDA. Query parameters. Available at: https://open.fda.gov/apis/query-parameters/
9. openFDA. Paging. Available at: https://open.fda.gov/apis/paging/
10. HazyResearch. icij-maude: Weakly supervised classification of adverse-event reports from the FDA MAUDE database. GitHub. Available at: https://github.com/HazyResearch/icij-maude
11. autonlab. fda_maude: Scripts to download and wrangle publicly available data from FDA MAUDE. GitHub. Available at: https://github.com/autonlab/fda_maude
12. Li X, Feng Y, Gong Y, Chen Y. Assessing the reproducibility of research based on the Food and Drug Administration Manufacturer and User Facility Device Experience data. J Patient Saf.

2024;20(5):e45–e58. doi:10.1097/PTS.0000000000001220. PMID: 38470959. PMCID: PMC11636620.

13. Pallets Projects. Flask: a lightweight WSGI web application framework. Available at: https://flask.palletsprojects.com/
14. Marouf M. MAUDEMetrics: an open-source framework for automated extraction and analysis of FDA MAUDE device reports (version 3.1.1) [software]. Zenodo; 2026. doi:10.5281/zenodo.16691960.
15. SQLite Development Team. SQLite. Available at: https://www.sqlite.org/
16. McKinney W. Data structures for statistical computing in Python. Proc 9th Python Sci Conf. 2010:56–61. doi:10.25080/Majora-92bf1922-00a.
17. Harris CR, Millman KJ, van der Walt SJ, et al. Array programming with NumPy. Nature. 2020;585:357–362. doi:10.1038/s41586-020-2649-2.
18. openpyxl Development Team. openpyxl: a Python library to read/write Excel 2010 xlsx/xlsm files. Available at: https://openpyxl.readthedocs.io/
19. Marouf M, AbuYounis A, Ghazou A, Sbei O, Alqudah AA. Adverse events associated with the subconjunctival Ex-Press® Glaucoma Filtration Device: a ten-year review of the MAUDE database. Int Ophthalmol. 2025;46(1):44. doi:10.1007/s10792-025-03886-8.
20. Hawas Y, Ghazou A, Gerges MW, Marouf M, Alkhawaldeh IM, Alqudah AA. Adverse events of trabecular micro-bypass minimally invasive glaucoma surgery devices: a five-year analysis from the FDA MAUDE database. Int Ophthalmol. 2025;45(1):493. doi:10.1007/s10792-025-03866-y.
21. Hersh W. The health information technology workforce: estimations of demands and a framework for requirements. Appl Clin Inform. 2010;1(2):197–212. doi:10.4338/ACI-2009-11-R-0011.
22. Heneghan C, Thompson M, Billingsley M, Cohen D. Medical-device recalls in the UK and the device-regulation process: retrospective review of safety notices and alerts. BMJ Open. 2011;1(1):e000155. doi:10.1136/bmjopen-2011-000155.

Figures

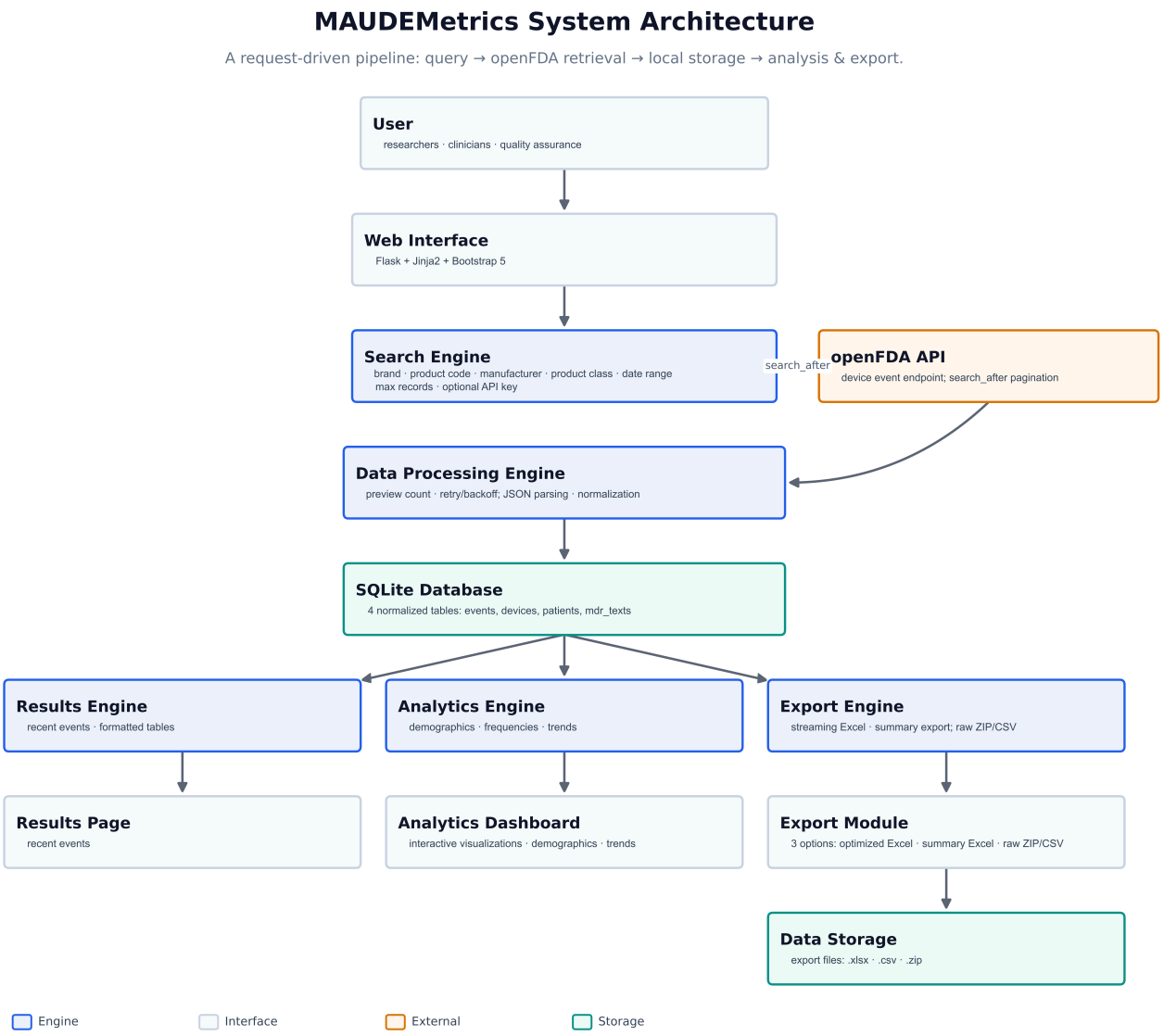


Figure 1

MAUDEMetrics architecture and workflow.

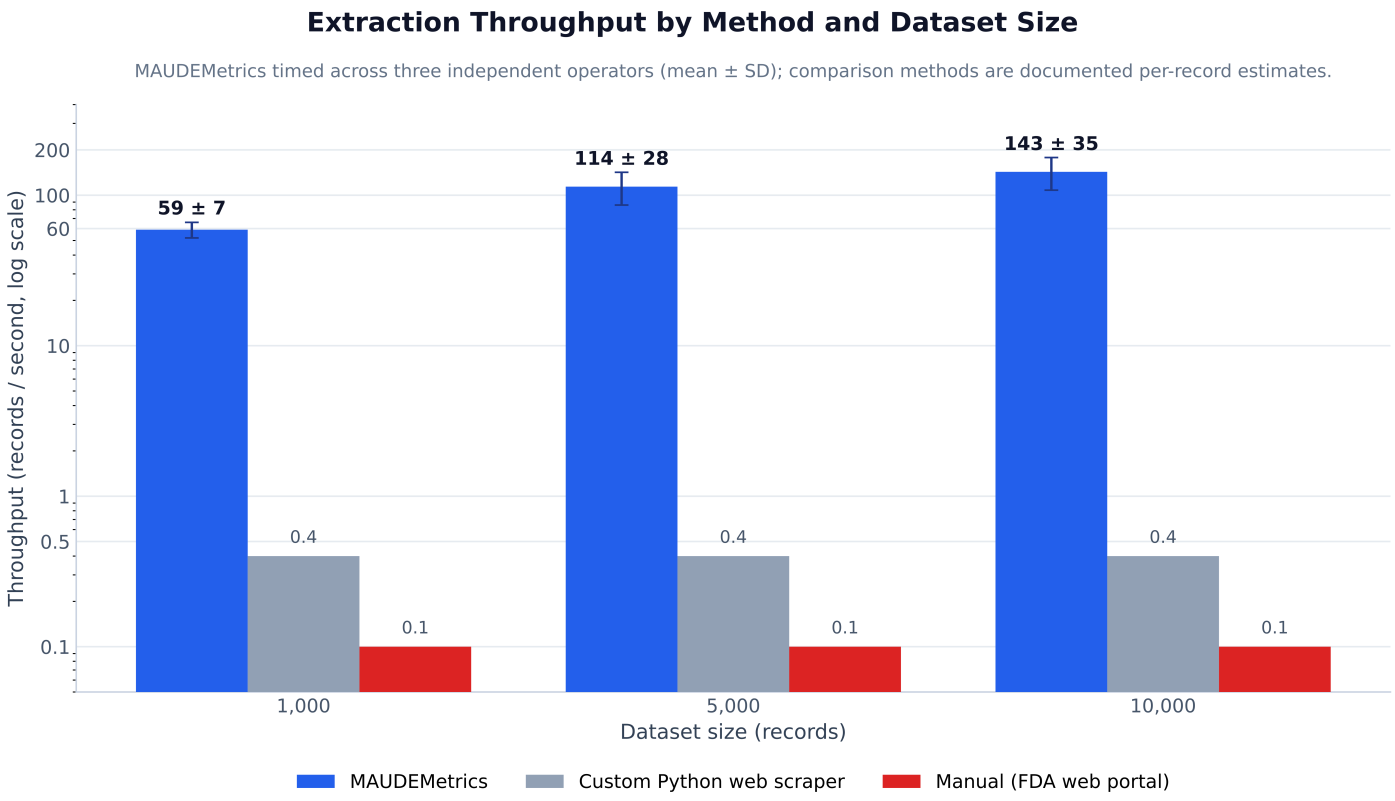


Figure 2

Data extraction performance by method. Log-scaled comparison of extraction throughput (records/second) across manual, scraping, and automated methods.

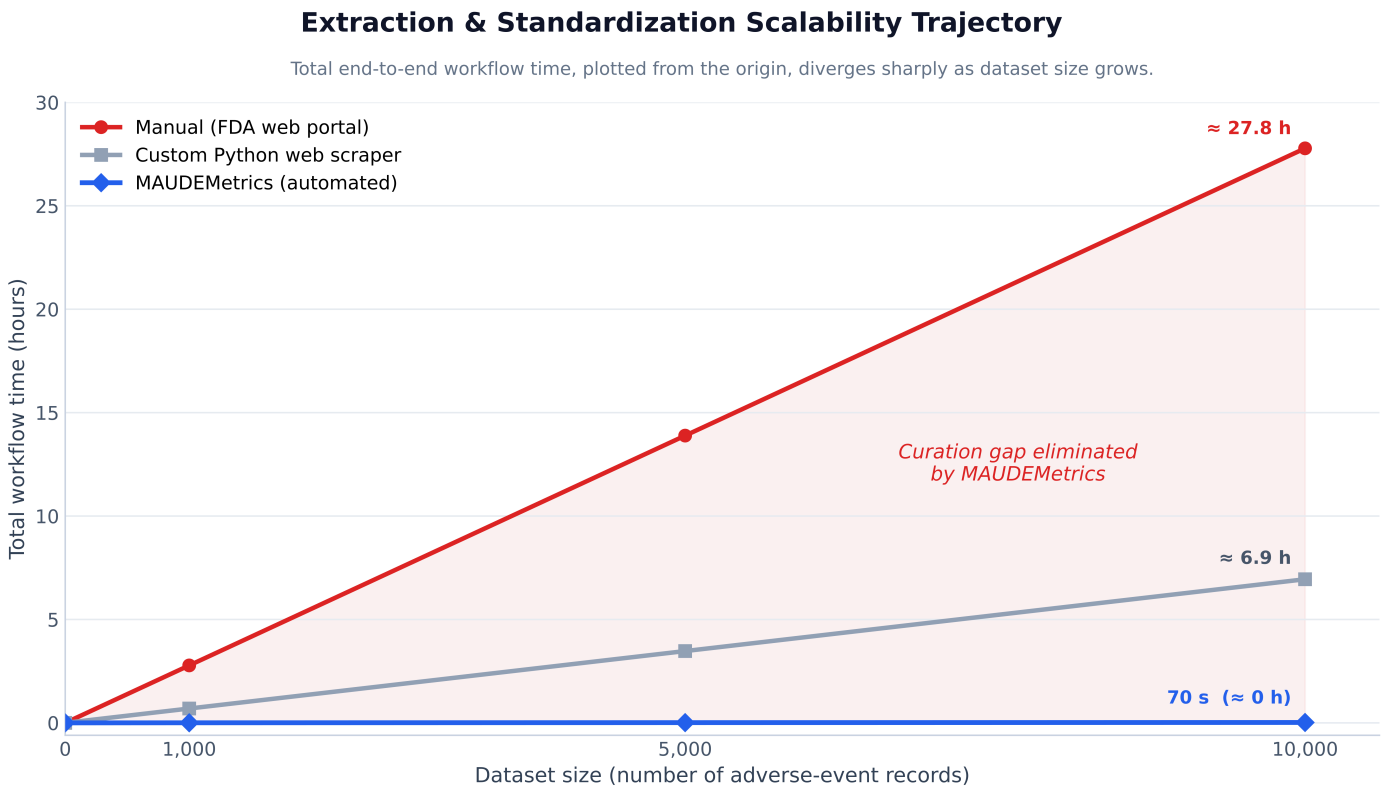


Figure 3

Extraction and standardization scalability trajectory. Divergence of manual vs. automated extraction cost across expanding dataset sizes.

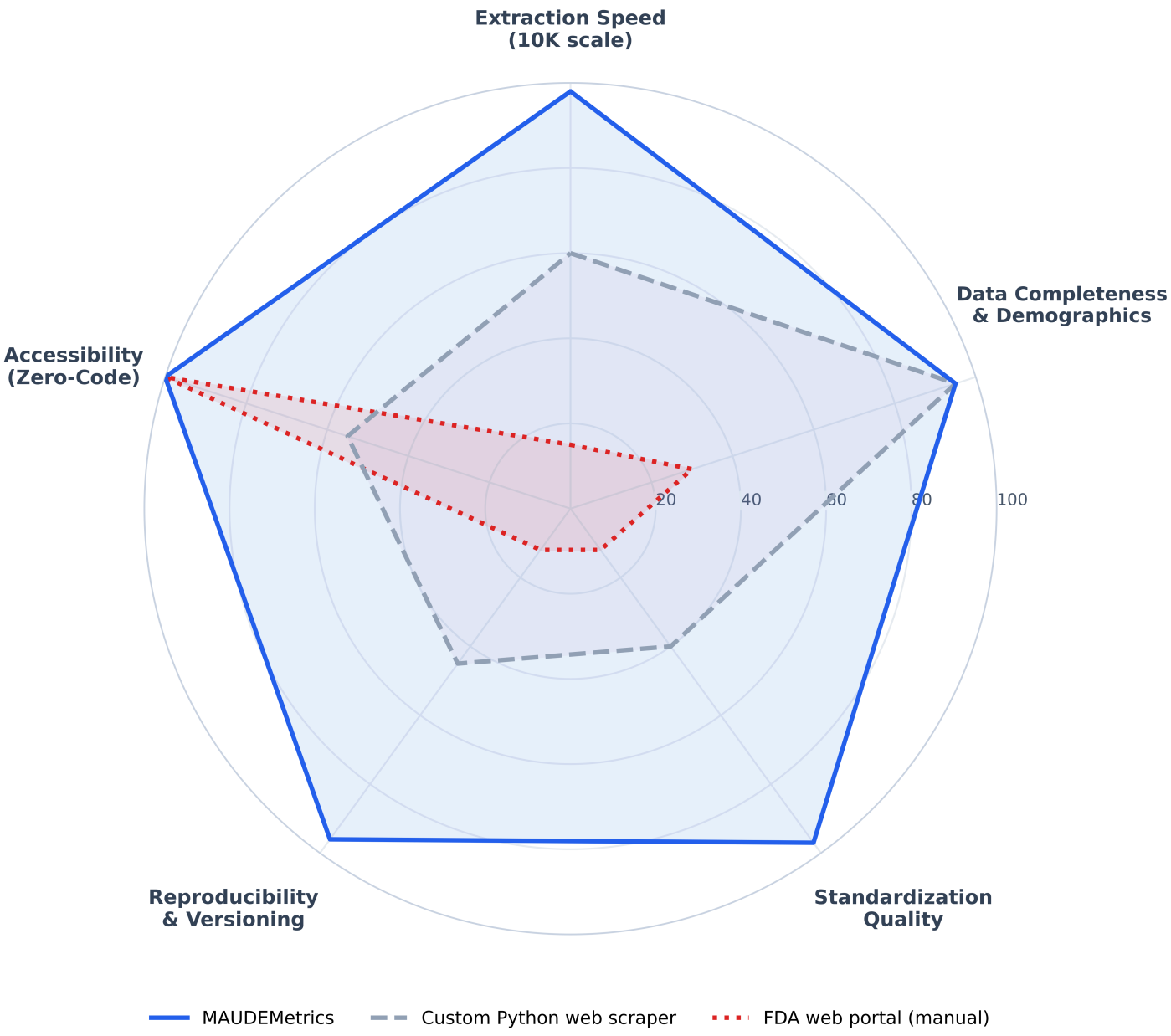


Figure 4

Performance fingerprint of MAUDE access methods. Multidimensional radar plot mapping the five core methodological capability dimensions across the FDA Web Portal, Custom Python Scraping, and MAUDEMetrics.

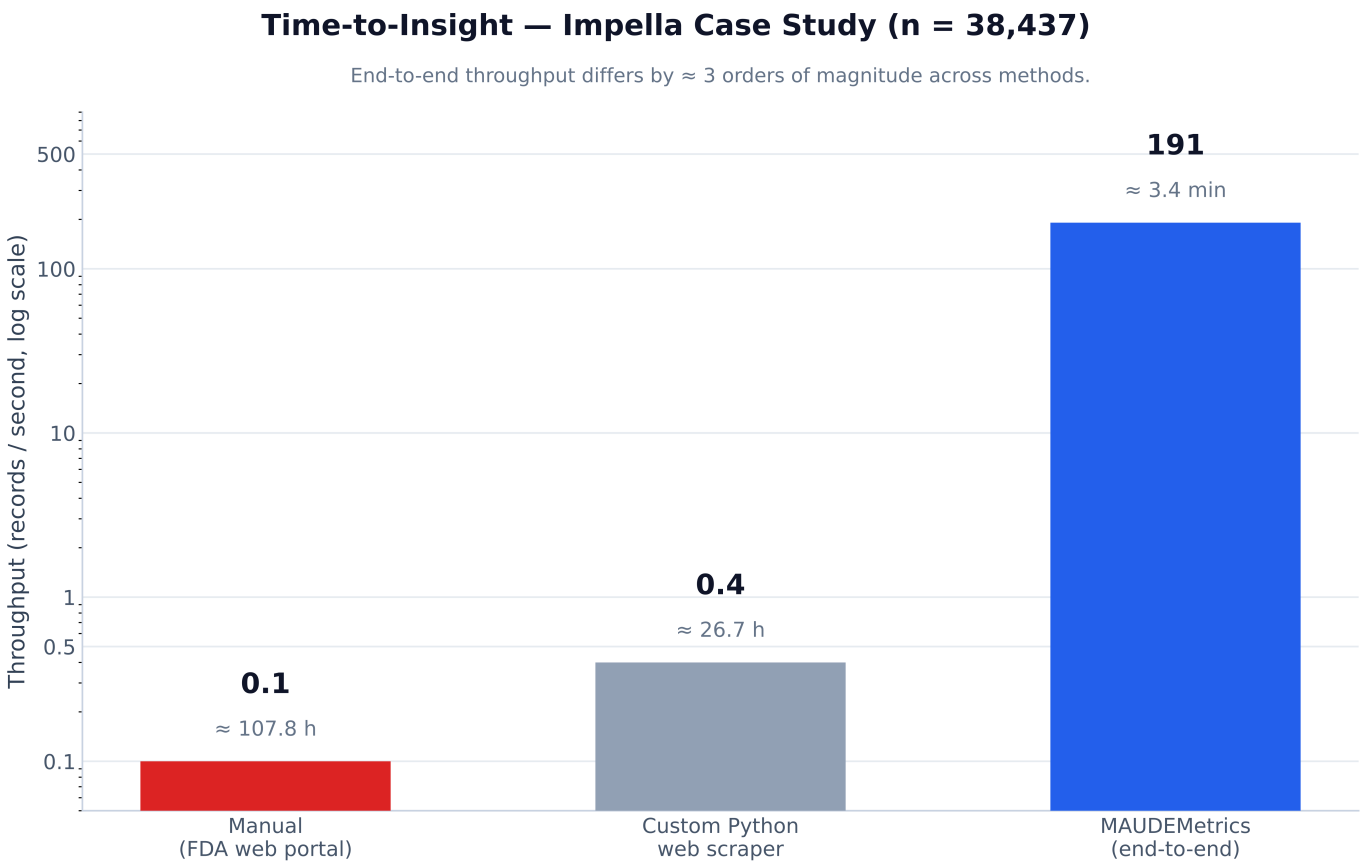


Figure 5

Time-to-insight comparison for the Impella case study (38,437 raw MDR submissions, inclusive of follow-ups and per-component duplicates). MAUDEMetrics throughput (≈ 191 records/sec end-to-end) is contrasted with an estimated manual throughput of ≈ 0.1 records/sec; the displayed difference is an order-of-magnitude comparison contingent on a per-report time-and-motion rate consistently observed by co-authors across prior MAUDE-based studies [19, 20].

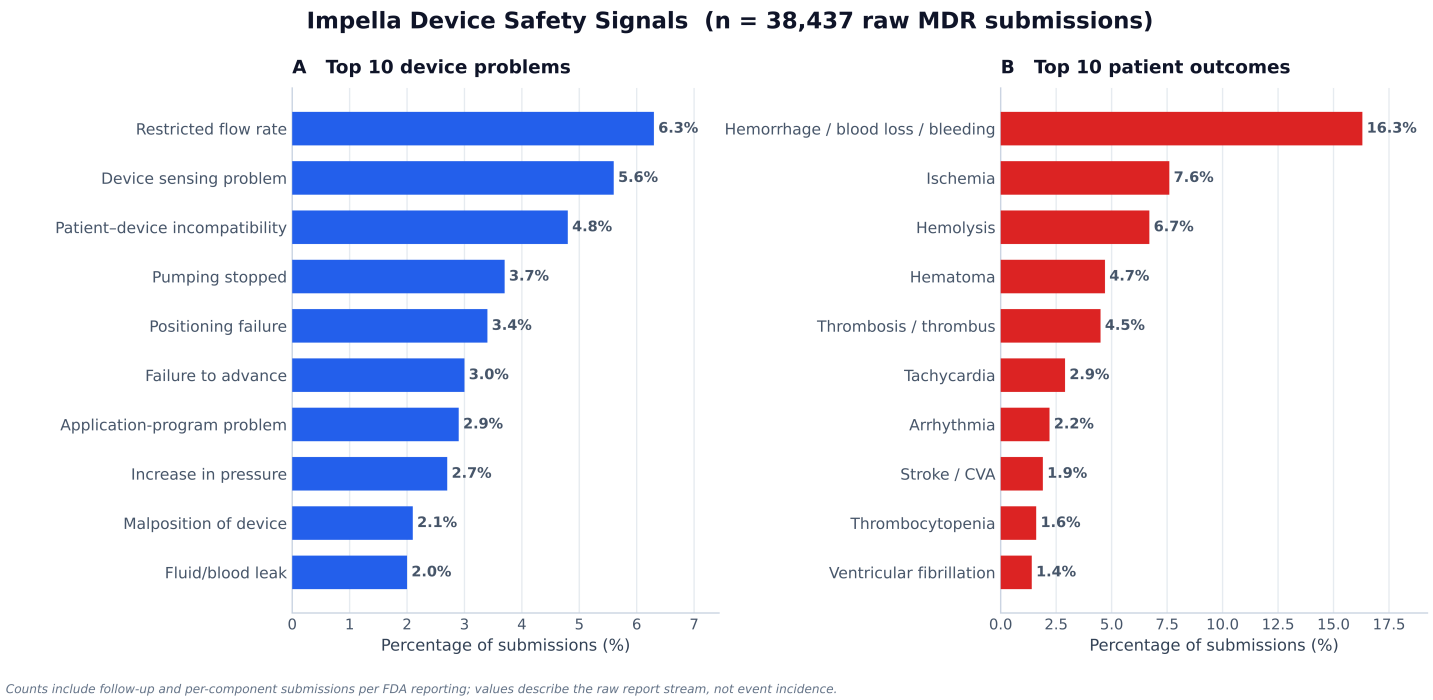


Figure 6

Impella device safety signals. Top 10 raw device problems and patient outcomes from the Impella case study (counts inclusive of follow-ups and per-component duplicates).

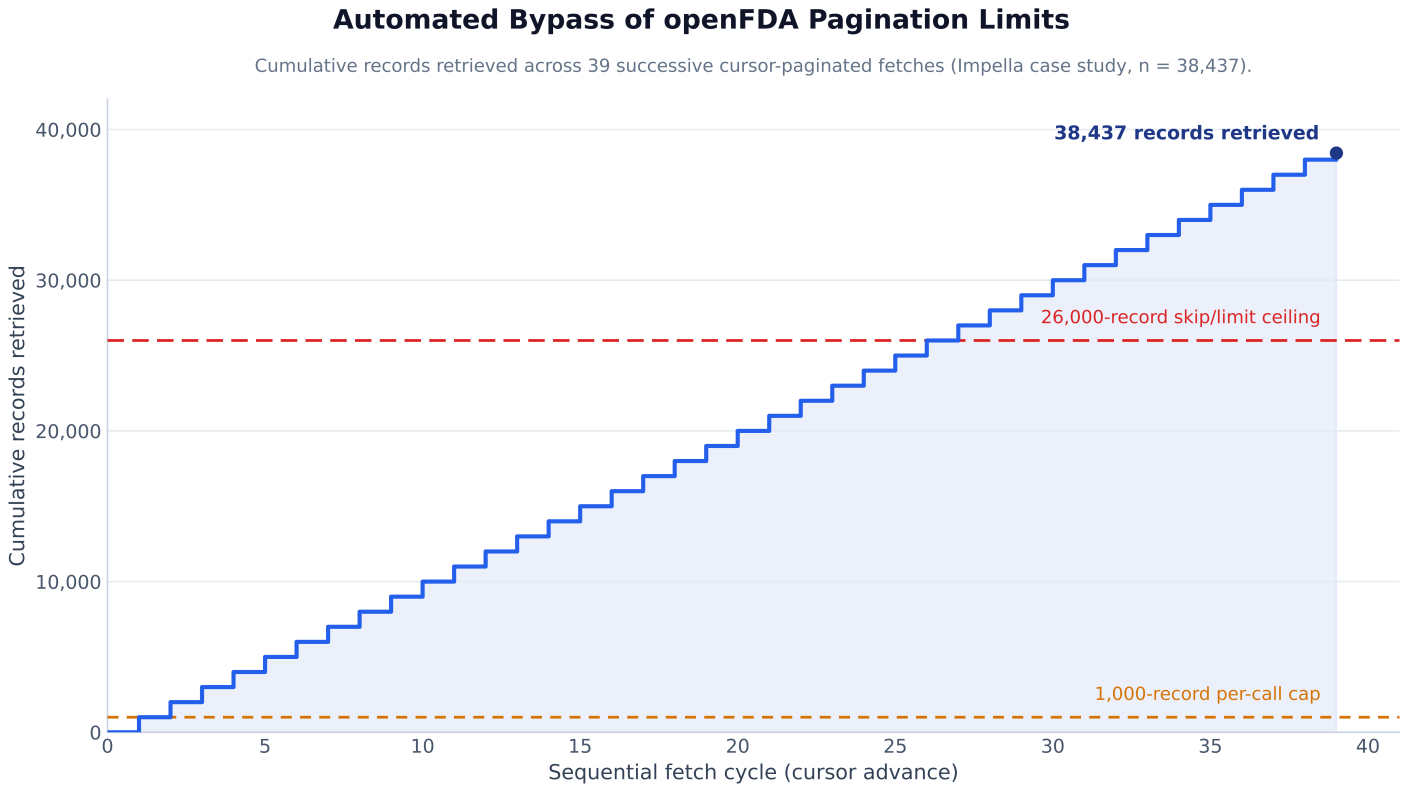


Figure 7

Automated overcoming of openFDA limits. Cumulative staircase plot demonstrating MAUDEMetrics' cursor-based pagination capacity to bypass the 1,000-record-per-call API limit, successfully retrieving all 38,437 Impella records across 39 successive fetch cycles.

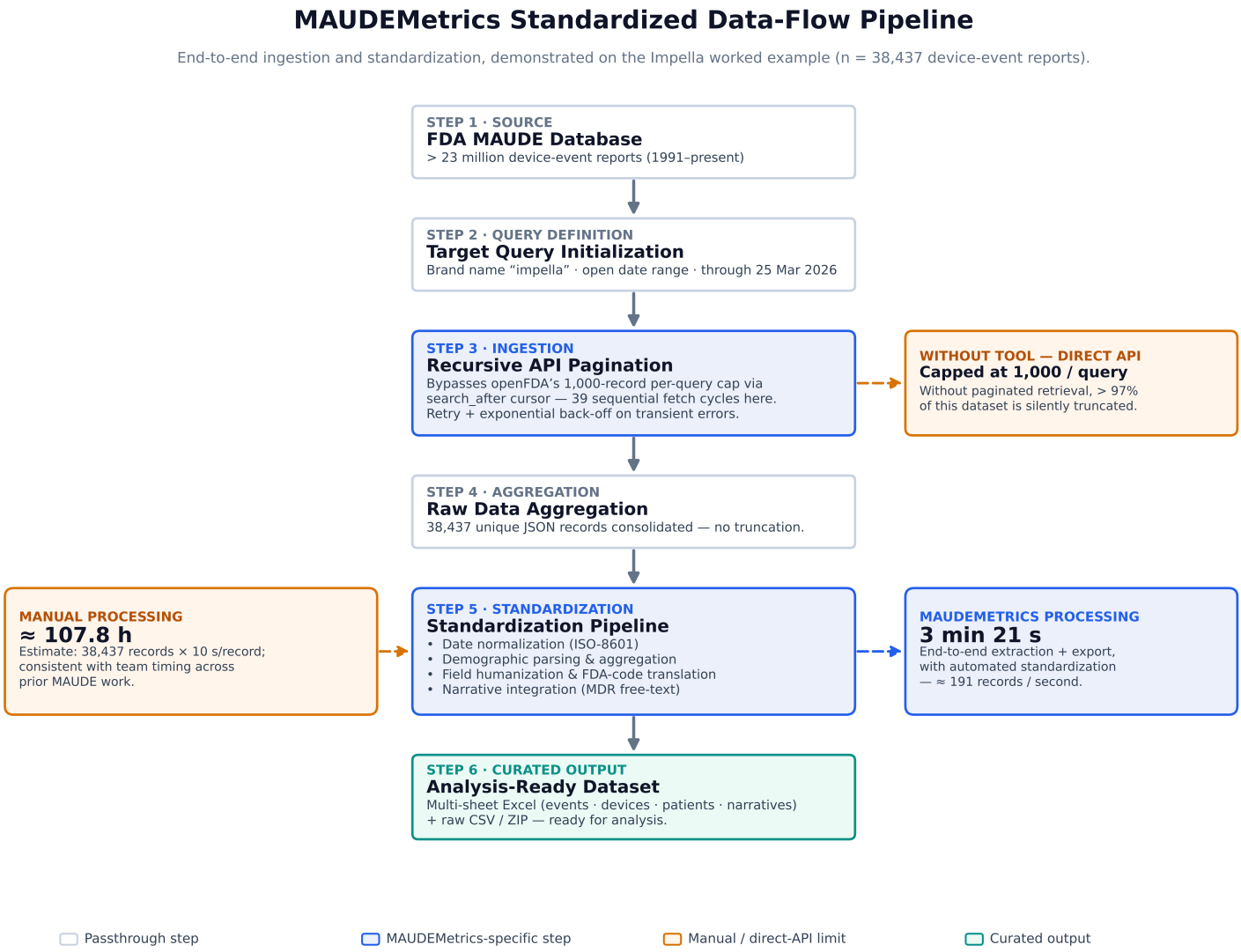


Figure 8

MAUDEMetrics standardized data-flow pipeline. Process flowchart detailing the automated stages of data extraction, pagination bypass, and demographic standardization, illustrated with the Impella case study.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryFile1.docx
- SupplementaryFile2.docx

- SupplementaryFile3aImpellaDataset.xlsx
- SupplementaryFile3bImpellaSummaryStatistics.xlsx
- SupplementaryFile4.docx