DataSynth

Enterprise Synthetic Data Platform

High-Performance Generation for Accounting, Audit & ML

Rust core • Python wrapper • REST/gRPC API • Desktop UI

Executive Overview

Version 0.2.2

Deterministic, privacy-preserving synthetic data generation
for enterprise finance, compliance testing, and machine learning.

Built with Rust & Python • 15 modular crates • 100 K+ entries/sec

Open-source (Apache-2.0) • Commercial license available

Contact: michael.ivertowski@ch.ey.com

Contents

1 Executive Summary

DataSynth is a high-performance synthetic data platform purpose-built for enterprise accounting, audit analytics, and machine learning. Written in Rust for maximum throughput and memory safety — with a full Python wrapper for data-science workflows — it generates realistic, internally coherent financial data that satisfies the statistical and structural properties of real-world enterprise resource planning (ERP) systems.

Why DataSynth?

- **No real data required** — eliminates privacy, regulatory, and procurement barriers to analytics development.
- **Full auditability** — deterministic, seeded generation means every dataset is perfectly reproducible.
- **ML-ready from day one** — ground-truth labels for fraud, anomalies, and data quality ship alongside every record.
- **Domain depth** — covers the full accounting lifecycle: journal entries, document flows, subledgers, FX, intercompany, period close, and banking/AML.
- **Empirically grounded** — statistical distributions for journal entry line items, amount patterns, and temporal volumes are calibrated against empirical research conducted on real-world enterprise datasets, ensuring synthetic output mirrors the structural properties of production ERP data.

1.1 At a Glance

100 K+ entries/sec (single thread)	**20+** labeled fraud typologies	**10** industry presets
15 modular Rust crates	**14** phases fully completed	**4** privacy levels

2 Platform Capabilities

2.1 Enterprise Accounting Simulation

DataSynth generates a complete, internally consistent accounting universe:

Domain	Capabilities
Journal Entries	Balanced debits/credits, Benford-compliant amounts, configurable line-item distributions, SAP ACDOCA format export.
Master Data	Vendors, customers, materials, fixed assets, employees with hierarchies, payment terms, credit ratings, and intercompany flags.
Document Flows	Full Procure-to-Pay (PO → GR → Invoice → Payment) and Order-to-Cash (SO → Delivery → Invoice → Receipt) with three-way match validation.
Intercompany	Matched IC journal entry pairs, transfer pricing (Cost-Plus, Resale-Minus, CUP), and consolidation elimination entries.
Subledgers	AR/AP open items and aging, fixed asset register with depreciation schedules, inventory positions and movements, GL-to-subledger reconciliation.
FX & Translation	Ornstein–Uhlenbeck exchange rate process, multi-currency trial balance translation, currency translation adjustment entries.
Period Close	Month-end accruals, depreciation runs, year-end closing entries, fiscal period status tracking.

2.2 Banking, KYC & AML

A dedicated banking module generates realistic transaction data for anti-money-laundering testing:

- **Customer personas**: Retail, Business, Trust profiles with full KYC envelopes (declared turnover, source of funds, geographic exposure, cash intensity).
- **AML typologies**: Structuring, funnel accounts, layering schemes, money mule networks, round-tripping, and adversarial spoofing for robustness testing.
- **Ground-truth labels**: Entity-level risk classifications, transaction-level labels, and investigation narratives.

2.3 Audit Simulation

Generates ISA-compliant audit artifacts:

- Engagement metadata with materiality thresholds (ISA 320).
- Workpapers per ISA 230, evidence per ISA 500.
- Risk assessments (ISA 315/330), findings (ISA 265), and professional judgment documentation (ISA 200).

2.4 Process Mining (OCEL 2.0)

Object-Centric Event Logs track many-to-many relationships between business objects (orders, invoices, payments) and activities — enabling conformance checking and process variant analysis.

3 Machine Learning & Analytics

3.1 Anomaly Injection Framework

DataSynth injects labeled anomalies across five categories, each with configurable rates and temporal patterns:

Category	Types	Examples
Fraud	20+	Fictitious transactions, revenue manipulation, ghost employees, kickback schemes
Error	7	Duplicate entries, reversed amounts, wrong period, misclassification
Process	5	Skipped approvals, threshold manipulation, out-of-sequence postings
Statistical	4	Unusual amounts, trend breaks, Benford violations, outlier values
Relational	3	Circular transactions, dormant account activity, unusual counterparties

Every injected anomaly carries a `LabeledAnomaly` record with full metadata, enabling supervised and semi-supervised learning pipelines without manual labeling effort.

3.2 Data Quality Variations

Realistic data imperfections for training data-quality ML models:

- **Missing values**: MCAR, MAR, MNAR, and systematic patterns.
- **Format variations**: Date, amount, and identifier format diversity across regional conventions.
- **Duplicates**: Exact, near, and fuzzy duplicates.
- **Typos**: Keyboard-aware substitution, OCR errors, homophones.
- **Encoding issues**: Mojibake, BOM artifacts, HTML entity corruption.

3.3 Graph & Network Export

Supported Graph Formats

PyTorch Geometric	`.pt` files with node features, edge index, edge attributes, labels, and train/val/test masks.
Neo4j	CSV node/edge files with Cypher import scripts.
DGL	Deep Graph Library format for GNN training.

Computed features include temporal signals (weekday, period-end flags), amount signals (log-amount, Benford probability, round-number flag), structural signals (line count, unique accounts), and one-hot categorical encodings.

4 Privacy & Fingerprinting

4.1 Privacy-Preserving Fingerprint Extraction

DataSynth can extract a statistical *fingerprint* from real data and synthesize new data that matches its properties — without ever copying individual records.

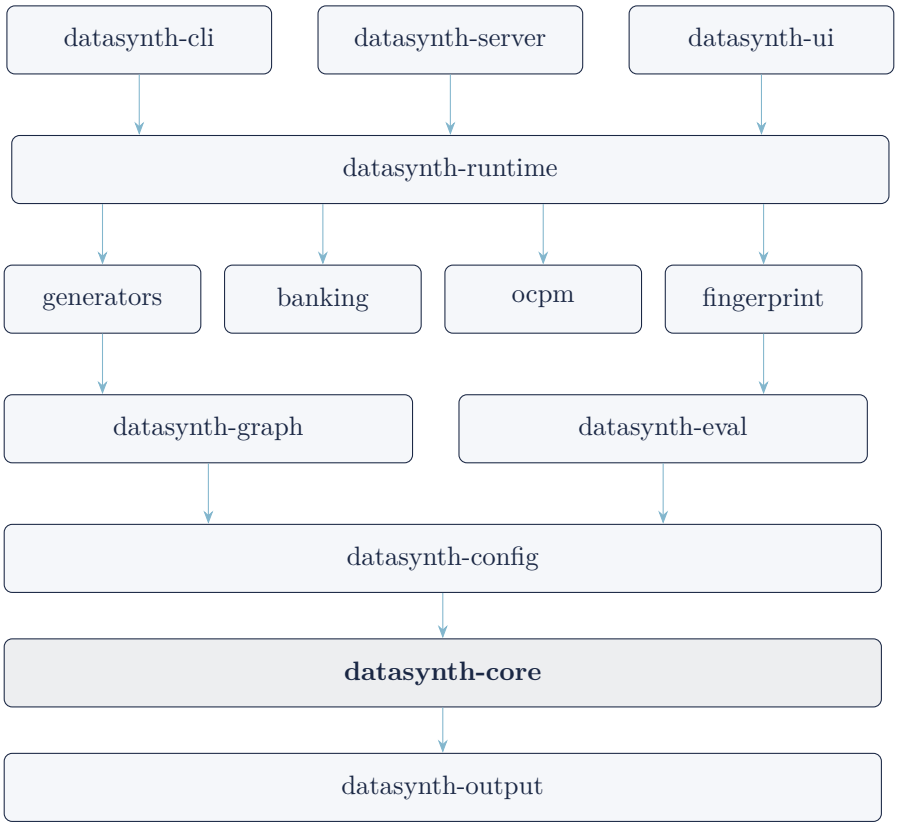
4.2 Privacy Levels

Level	Epsilon (ε)	k-Anon	Outlier %	Use Case
Minimal	5.0	3	99%	Low privacy, high utility
Standard	1.0	5	95%	Balanced (default)
High	0.5	10	90%	Sensitive environments
Maximum	0.1	20	85%	Maximum privacy guarantees

The privacy engine combines differential privacy (Laplace and Gaussian mechanisms), k-anonymity with rare-value suppression, and outlier winsorization. A full privacy audit trail records every decision and the cumulative epsilon budget spent.

5 Architecture

5.1 Layered Crate Architecture



5.2 Production-Grade Infrastructure

Capability	Details
Deterministic Output	ChaCha8 PRNG with configurable seed; identical seed ⇒ identical dataset.
Financial Precision	`rust_decimal` throughout; no IEEE 754 floating-point artifacts. Decimals serialized as strings.
Resource Guards	Unified CPU, memory, and disk monitoring with automatic throttling and graceful degradation (Normal → Reduced → Minimal → Emergency).
Collision-Free IDs	FNV-1a hash-based UUID factory with generator-type discriminators prevents document ID collisions across parallel generators.
API Layer	REST + gRPC + WebSocket with API-key authentication, sliding-window rate limiting, and configurable timeouts.
Desktop UI	Tauri + SvelteKit cross-platform application with 15+ configuration pages and real-time streaming viewer.
Python Wrapper	`datasynth-py` package with blueprints, pandas integration, and WebSocket streaming support.

6 Industry Presets

DataSynth ships with ten industry-specific configuration presets, each tuned for realistic business process weightings, chart-of-accounts structures, and regional multi-company setups.

Industry	Key Weight	Characteristics
Manufacturing	40% P2P	Heavy procurement, BOM-driven materials, multi-plant operations.
Retail	50% O2C	High-volume sales, inventory-intensive, multi-currency.
Financial Services	40% R2R	Report-heavy, regulatory compliance, intercompany structures.
Healthcare	15% H2R	Labour-intensive, complex billing, compliance-driven.
Technology	15% H2R	Knowledge workers, SaaS revenue recognition, R&D capitalization.
Professional Svcs.	—	Time-based billing, project accounting.
Energy	—	Capital-intensive, long-lived assets.
Transportation	—	Fleet management, route-based costing.
Real Estate	—	Property portfolios, lease accounting.
Telecommunications	—	Subscription revenue, network assets.

Each preset supports three complexity tiers: **Small** (∼100 GL accounts), **Medium** (∼400 accounts), and **Large** (∼2 500 accounts).

7 Use Cases

Primary Use Cases

Fraud Detection ML — Train and validate models on labeled fraud typologies with realistic base-rate imbalance.

Graph Neural Networks — Export transaction graphs in PyTorch Geometric, Neo4j, or DGL format with pre-computed features and train/val/test splits.

AML & KYC Testing — Generate banking transaction data with structuring, layering, and mule patterns for compliance system validation.

Audit Analytics — Produce ISA-compliant audit artifacts and anomaly-injected financial data for analytics tool development.

ERP Integration Testing — Generate SAP ACDOCA-format data with full document chains for system migration and integration testing.

Process Mining — OCEL 2.0 event logs for process discovery, conformance checking, and variant analysis.

SOX Compliance Testing — Internal control definitions, segregation-of-duties conflict detection, and approval threshold validation.

Data Quality ML — Labeled missing values, typos, duplicates, and format variations for training data-cleansing models.

8 Evaluation & Auto-Tuning

DataSynth includes a built-in evaluation framework that measures synthetic data quality across four dimensions:

1. **Statistical Fidelity** — KS statistic, Wasserstein distance, Benford's Law MAD, amount distribution fit.
2. **Coherence** — Balance-sheet validation, intercompany matching, document chain integrity, subledger reconciliation.
3. **Data Quality** — Completeness, consistency, duplicate rates, format correctness, uniqueness.
4. **ML Readiness** — Feature distributions, label quality, graph structure, train/val/test split balance.

An **auto-tuning engine** analyses evaluation results and produces prioritized configuration patches with expected improvement estimates — closing the loop between generation and validation.

9 Getting Started

9.1 Quick Start (CLI)

```
# Generate with demo preset
datasynth-data generate -demo -output ./output

# Create an industry-specific config
datasynth-data init -industry manufacturing -complexity medium -o config.yaml

# Generate from config
datasynth-data generate -config config.yaml -output ./output
```

9.2 Quick Start (Python)

```
from datasynth_py import DataSynth
from datasynth_py.config import blueprints

config = blueprints.retail_small(companies=4, transactions=10000)
synth = DataSynth()
result = synth.generate(config=config)
```

9.3 Server Mode

```
# Start REST/gRPC server with 4 worker threads
cargo run -p datasynth-server - -port 3000 -worker-threads 4
```

Links & Resources

Repository	https://github.com/ey-asu-rnd/SyntheticData
Documentation	https://ey-asu-rnd.github.io/SyntheticData/
Crates.io	https://crates.io/crates/datasynth-core
PyPI	https://pypi.org/project/datasynth-py/

Contact: michael.ivertowski@ch.ey.com