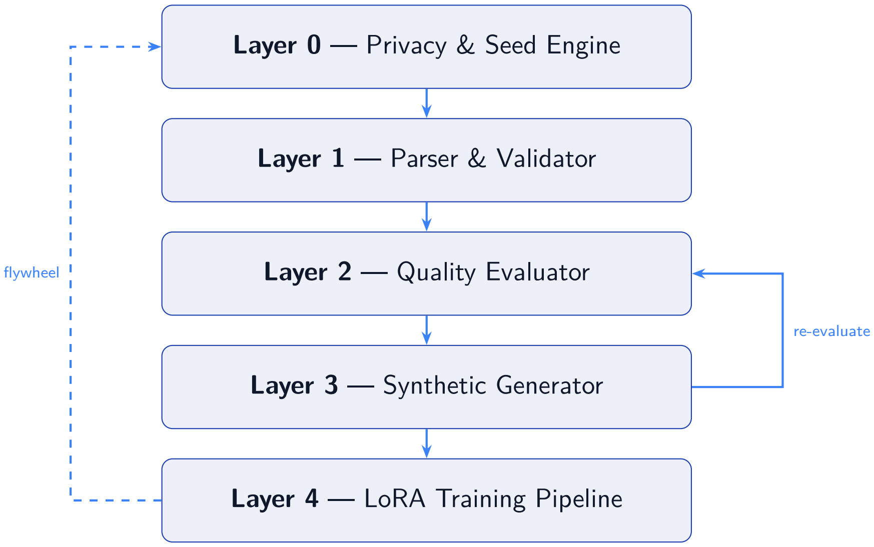
UNCASE

Unbiased Neutral Convention for Agnostic Seed Engineering

Technical Whitepaper — Version 2.0

March 2026


Layer 0 — Privacy & Seed Engine
Layer 1 — Parser & Validator
Layer 2 — Quality Evaluator
Layer 3 — Synthetic Generator
Layer 4 — LoRA Training Pipeline
flywheel
re-evaluate

Open-source framework for generating privacy-safe synthetic conversational data in regulated industries

https://github.com/uncase-ai/uncase | https://uncase.md

Contents

1 Executive Summary

Large language models are transforming every industry, but fine-tuning them for regulated sectors—healthcare, finance, legal, manufacturing—runs into a hard wall: **you cannot use real customer data without violating privacy regulations**, and you cannot produce high-quality domain models without real conversational patterns.

UNCASE solves this by providing a **complete, open-source pipeline** that converts a small number of real conversation *seeds*—stripped of all personally identifiable information (PII)—into thousands of synthetic training conversations that preserve the domain knowledge, tone, and structure of the originals while guaranteeing zero PII leakage.

1.1 What UNCASE Does Today

- **5-layer pipeline** from raw data ingestion to trained LoRA adapter, fully orchestrated.
- **9 quality metrics** including two semantic evaluators (LLM-as-Judge and embedding drift), with hard thresholds that reject substandard data automatically.
- **Zero-PII guarantee**: dual-layer scanning (regex heuristics + Microsoft Presidio NER) catches 14+ categories of personal data before it enters the pipeline.
- **Adversarial input protection**: PromptShield module detects prompt injection, jailbreak attempts, and PII solicitation with 5 threat categories.
- **150 curated seed packages** across 3 industry domains (automotive, medical, finance), with 56 scenario templates covering edge cases.
- **11 fine-tuning export formats** (ChatML, Llama, Qwen, Mistral, and more) with full tool-call training support.
- **106 REST API endpoints** across 24 routers, a React 19 dashboard, a CLI, and a Python SDK.
- **5 compliance profiles** (HIPAA, GDPR, SOX, LFPDPPP, EU AI Act) as frozen, auditable configurations.
- **Enterprise-grade infrastructure**: JWT auth with RBAC, audit logging, LLM cost tracking, rate limiting, Prometheus metrics, and Grafana dashboards.

1.2 Who Is This For

1. **ML teams in regulated industries** that need domain-specific training data without legal risk.
2. **Enterprises deploying conversational AI** (chatbots, virtual assistants, copilots) that must comply with HIPAA, GDPR, SOX, or the EU AI Act.
3. **AI startups** building vertical solutions that lack access to large proprietary datasets.
4. **Data science teams** that want to augment small real-world datasets with high-quality synthetic conversations.

2 The Problem: Training Data in Regulated Industries

2.1 The Data Paradox

Fine-tuning a large language model for a specific domain—say, a medical consultation assistant or a financial advisor chatbot—requires thousands of real conversations that demonstrate the correct patterns, terminology, tone, and decision-making flow.

In regulated industries, these conversations exist but are **locked behind legal, ethical, and compliance barriers**:

- **HIPAA** (US): Protected Health Information cannot leave the covered entity without a Business Associate Agreement and de-identification per Safe Harbor or Expert Determination methods.
- **GDPR** (EU): Personal data processing requires explicit consent, purpose limitation, and data minimization. Synthetic data generation from personal data constitutes processing.
- **SOX** (US): Financial services must maintain audit trails for all data used in automated decision systems, with 7-year retention.
- **EU AI Act**: High-risk AI systems (healthcare, finance, legal) require documented training data governance, bias testing, and conformity assessments.

2.2 Why Existing Approaches Fall Short

Approach	How It Works	Why It's Not Enough
Manual anonymization	Humans review and redact PII	Expensive, slow, error-prone. Misses context-dependent PII (e.g., "the diabetic patient in room 4").
Rule-based scrubbing	Regex patterns remove known PII formats	Catches emails and SSNs but misses names, locations, medical conditions embedded in free text.
Template generation	Fill-in-the-blank conversation templates	Produces stilted, repetitive data. Models trained on templates generate template-like output.
Generic LLM generation	Ask GPT-4/Claude to "create a medical conversation"	Lacks domain specificity. Hallucinates facts. No quality guarantees. No traceability to real patterns.
Data marketplaces	Buy pre-packaged datasets	Rarely domain-specific enough. Provenance unclear. May contain undetected PII. Not customizable.

Table 1: Comparison of existing approaches to training data generation.

2.3 What's Actually Needed

A solution that:

1. Starts from **real conversational patterns** (not templates or generic prompts).
2. Removes **all PII before any processing** begins—not after.
3. Generates synthetic data that **preserves domain structure, tone, and factual accuracy**.

4. **Measures quality automatically** against multiple dimensions with hard pass/fail thresholds.
5. Produces output in **every major fine-tuning format** (ChatML, Llama, Qwen, Mistral, etc.).
6. Maintains a **complete audit trail** from raw input to trained model.
7. Is **open-source, self-hostable**, and runs behind the enterprise firewall.

This is what UNCASE provides.

3 The SCSF Architecture

UNCASE implements the **Synthetic Conversation Seed Framework (SCSF)**, a 5-layer pipeline where each layer has a single responsibility and communicates via validated Pydantic v2 schemas.

3.1 Pipeline Overview

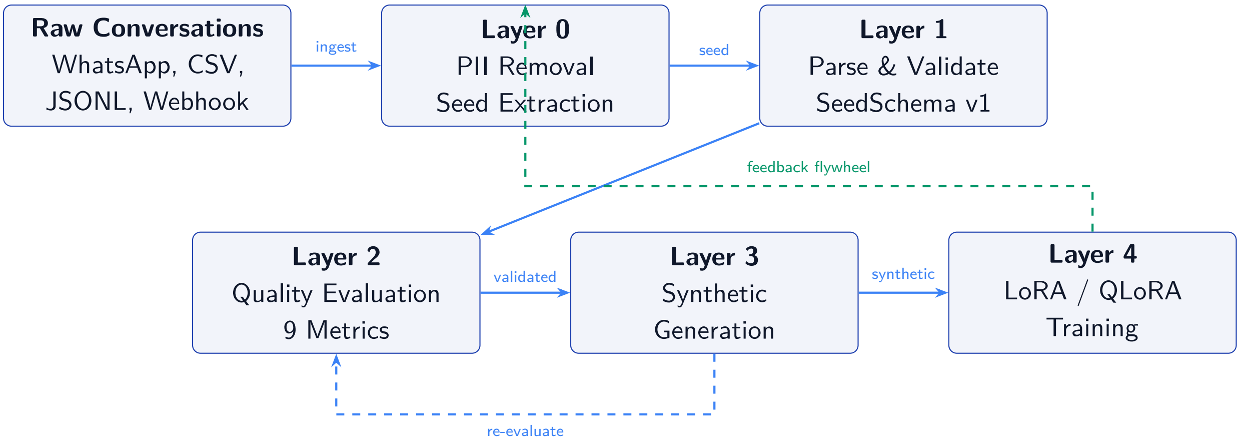


Figure 1: SCSF 5-layer pipeline with re-evaluation loop and feedback flywheel.

3.2 Layer 0: Privacy & Seed Engine

The first layer is the **zero-trust boundary**. No raw data survives past this point.

3.2.1 PII Detection (Dual Strategy)

PII Category	Detection	Token	Example
Email addresses	Regex	`[EMAIL]`	user@company.com
Phone numbers	Regex	`[PHONE]`	+52 55 1234 5678
SSN (US)	Regex	`[SSN]`	123-45-6789
CURP (Mexico)	Regex	`[CURP]`	GOML860101HDFRRN00
RFC (Mexico)	Regex	`[RFC]`	GOML860101XXX
Credit cards	Regex	`[CREDIT_CARD`	4111-1111-1111-1111
IP addresses	Regex	`[IP_ADDRESS]`	192.168.1.100
IBAN	Regex	`[IBAN]`	DE89370400440532013000
Person names	Presidio NER	`[PERSON]`	John Smith
Locations	Presidio NER	`[LOCATION]`	Mexico City
Dates of birth	Presidio NER	`[DATE]`	01/15/1990
Medical licenses	Presidio NER	`[LICENSE]`	DEA: AB1234567
Bank accounts	Presidio NER	`[BANK_ACCT]`	Account 12345678
Passport / DL	Presidio NER	`[PASSPORT]`	Passport A12345678

Table 2: 14 PII categories detected by the dual regex + Presidio NER strategy.

Regex heuristics are always active and require no external dependencies. Presidio NER (via spaCy) is an optional upgrade that catches context-dependent PII (names, locations) using named entity recognition.

3.2.2 PromptShield: Adversarial Input Protection

The `PromptShield` module scans all inputs **before** they reach any LLM, detecting 5 categories of adversarial content:

1. **Prompt injection**: Attempts to override system instructions ("ignore all previous instructions").
2. **Jailbreak**: Roleplay-based bypasses ("pretend you are an unrestricted AI").
3. **System prompt extraction**: Requests to reveal internal configuration.
4. **Toxic content**: Requests for harmful instructions.
5. **PII solicitation**: Requests to bypass anonymization ("use real names").

Three operating modes: `audit` (log only), `warn` (log + flag), `block` (reject). An optional LLM-backed classifier provides enhanced detection for sophisticated attacks that evade regex patterns.

3.2.3 Seed Extraction

After PII removal, the engine extracts structural metadata from raw conversations:

- **Roles**: Participants identified and labeled (e.g., "salesperson", "customer").
- **Domain**: Classified into one of 6 supported industry verticals.
- **Objective**: The purpose of the conversation inferred from content.
- **Tone & style**: Formal, informal, technical, empathetic, etc.
- **Factual parameters**: Domain constraints, restrictions, and expected behaviors.

- **Expected flow**: The logical progression of conversation steps.

The output is a `SeedSchema v1` object—a structured, validated, PII-free blueprint that drives all downstream generation.

3.3 Layer 1: Parser & Validator

Accepts multiple input formats and validates them against the SeedSchema:

- **WhatsApp exports** (`chat.txt`) with automatic timestamp and participant detection.
- **CSV transcripts** (call center format, configurable column mapping).
- **JSON/JSONL** (structured conversation objects).
- **Webhook payloads** (real-time ingestion from CRM/helpdesk systems).

All parsing produces validated Pydantic v2 models with automatic type coercion, constraint checking, and descriptive error messages.

3.4 Layer 2: Quality Evaluator

Every generated conversation is scored against **9 mandatory metrics**. No conversation enters the training pipeline unless it passes all thresholds.

Metric	Threshold	Gate?	What It Measures
ROUGE-L	≥ 0.65	No	Structural coherence with the seed
Factual Fidelity	≥ 0.90	No	Domain fact accuracy
Lexical Diversity (TTR)	≥ 0.55	No	Vocabulary richness (type-token ratio)
Dialogic Coherence	≥ 0.85	No	Inter-turn logical consistency
Tool Call Validity	≥ 0.90	No	Tool call schema correctness (5 dimensions)
Semantic Fidelity	≥ 0.60	No	LLM-as-Judge rubric score (4 dimensions)
Embedding Drift	≥ 0.40	No	Cosine similarity between seed and generated text
Privacy Score	$= 0.00$	**Yes**	Zero residual PII (hard gate)
Memorization	< 0.01	**Yes**	Extraction attack success rate (hard gate)

Table 3: Quality metrics with mandatory thresholds. Gate metrics cause immediate rejection.

3.4.1 Composite Score Formula

$$Q = \begin{cases} \min(\text{ROUGE-L}, \text{Fidelity}, \text{TTR}, \text{Coherence}, \text{Tool Validity}, \text{Sem. Fidelity}, \text{Emb. Drift}) & \text{if privacy} = 0 \text{ and} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Using the **minimum** rather than the average ensures no single dimension can be weak while others compensate. The privacy and memorization metrics act as hard gates: any PII leakage or memorization above threshold sets $Q = 0$ regardless of other scores.

3.4.2 Semantic Fidelity (LLM-as-Judge)

A fast, cost-efficient LLM (default: Claude Haiku) grades each conversation on a structured rubric across 4 dimensions:

1. **Factual fidelity** (35% weight): Does it respect domain constraints?
2. **Logical coherence** (30%): Does the dialog flow make sense?
3. **Role consistency** (20%): Do participants stay in character?
4. **Naturalness** (15%): Does it feel like a real conversation?

Each dimension is graded 1–5, then the weighted average is normalized to $[0, 1]$. The metric falls back gracefully to a neutral score (0.5) when the LLM API is unavailable.

3.4.3 Embedding Drift

Measures semantic distance between the seed specification and the generated conversation using cosine similarity over embedding vectors. High similarity means the conversation stays on-topic; low similarity indicates drift.

Two backends:

- **LLM embeddings** (primary): Uses provider-agnostic embedding API via LiteLLM.
- **TF-IDF fallback** (always available): Term-frequency cosine similarity, no API required.

3.4.4 Tool Call Validator

For conversations involving tool use, a dedicated validator checks 5 dimensions:

1. **Hallucinated tools**: Tool name doesn't exist in the seed's definitions.
2. **Missing required arguments**: Required parameters not provided.
3. **Unknown arguments**: Parameters not defined in the tool schema.
4. **Type mismatches**: Argument types don't match schema definitions.
5. **Sequence validation**: Tool call order matches expected patterns (exact, subset, or partial order).

3.5 Layer 3: Synthetic Generator

The generation engine uses **LiteLLM** as a provider-agnostic interface, supporting 7+ LLM providers:

Provider	Connection	Example Models
Anthropic (Claude)	Cloud API	Claude Sonnet 4.6, Claude Haiku 4.5
OpenAI	Cloud API	GPT-4o, GPT-4o-mini
Google (Gemini)	Cloud API	Gemini 2.0 Flash, Gemini 2.5 Pro
Groq	Cloud API	Llama 3.3-70B, Mixtral 8x7B
Ollama	Local API	Any GGUF model
vLLM	Local API	Any HuggingFace model
Custom (OpenAI-compatible)	Any	Together AI, Fireworks, etc.

Table 4: Supported LLM providers via LiteLLM.

3.5.1 Smart Retry Strategy

The generator implements an intelligent retry mechanism:

1. First attempt uses `response_format=json_object` when the model supports it.
2. On JSON format failure, retries without structured output.
3. On each retry, **escalates temperature** by a configurable step (default: +0.1) to encourage diverse output and escape degenerate patterns.
4. Structured JSON extraction uses direct parse, markdown code block extraction, and object-key unwrapping—but **no fragile regex bracket-matching** that could silently corrupt data.

3.5.2 Feedback-Augmented Generation

When a conversation fails quality evaluation, the generator receives specific feedback about which metrics failed and by how much. The next generation attempt incorporates this feedback as additional prompt instructions, creating a **self-correcting loop** between Layers 2 and 3.

3.5.3 Parallel Pipeline Orchestration

The pipeline orchestrator uses `asyncio.gather()` with **semaphore-based concurrency control** to process seeds and conversations in parallel while respecting LLM rate limits. Configurable concurrency (default: 10 concurrent operations) prevents overwhelming API rate limits while maximizing throughput.

Three stages run in parallel:

1. **Seed creation**: Multiple raw conversations processed concurrently.
2. **Generation**: Multiple seeds generate synthetic data simultaneously.
3. **Evaluation**: Batched concurrent evaluation with configurable batch sizes.

3.6 Layer 4: LoRA Training Pipeline

The final layer transforms certified synthetic data into trained model adapters:

- **LoRA/QLoRA fine-tuning** via HuggingFace Transformers + PEFT.
- **11 export formats** ensure compatibility with every major model architecture.
- **MLflow experiment tracking** for hyperparameter logging and model versioning.
- **GPU deployment scripts** for vLLM serving with tensor parallelism (A40, A100, H100).
- Merge and serve pipeline: download base model, apply LoRA adapter, serve via vLLM with Cloudflare Tunnel for public access.

Format	Tool Support	Compatible Models
ChatML	✓	GPT-4, Qwen (base), Yi
Llama 3/4	✓	LLaMA 3, 3.1, 4
Qwen 3	✓	Qwen 3, Qwen 2.5
Mistral	✓	Mistral, Mixtral
Nemotron	✓	NVIDIA Nemotron
Harmony	✓	Cohere Command R+
Kimi/Moonshot	✓	Moonshot Kimi
MiniMax	✓	MiniMax
OpenAI API	✓	Any OpenAI-compatible endpoint
Alpaca	×	Instruction-tuned models

Table 5: Supported fine-tuning export formats. All formats except Alpaca support tool-use training data.

4 Domain Coverage & Scenario Templates

4.1 150 Curated Seed Packages

UNCASE ships with **150 professionally curated conversation seeds** across 3 primary domains, with 3 additional domains supported via scenario templates:

Domain	Seeds	Scenarios	Key Topics
Automotive Sales	50	12 + 5 edge	Vehicle inquiry, test drives, financing, trade-ins, fleet sales, warranty, frustrated customer
Medical Consultation	50	10 + 3 edge	Patient history, symptom assessment, lab results, prescriptions, insurance, anxious patient
Finance Advisory	50	10 + 4 edge	Portfolio review, risk profiling, market panic, suspicious activity, KYC/AML
Legal Advisory	—	8 + 3 edge	Case intake, conflict of interest, scope limitation, fee structures
Industrial Support	—	8 + 3 edge	Equipment diagnostics, safety incidents, production down, parts ordering
Education Tutoring	—	8 + 3 edge	Concept explanation, frustrated student, learning styles, exam prep

Table 6: Domain coverage with curated seeds and scenario templates.

4.2 Scenario Template System

Each scenario template defines:

- **Intent and objective**: What the conversation should accomplish.
- **Skill level**: Basic, intermediate, or advanced complexity.

- **Expected tool sequence**: Which domain tools should be called and in what order.
- **Flow steps**: Logical progression that overrides the seed's default flow.
- **Edge-case flag**: Marks scenarios that stress-test model robustness.
- **Weighted random selection**: Control scenario distribution in batch generation.

4.3 30 Domain Tools (5 per Industry)

Each domain comes with 5 built-in tools that simulate real-world integrations:

Domain	Tools
Automotive	Inventory search, price quotes, financing calculator, model comparison, CRM lookup
Medical	Patient history, medication database, appointments, lab results, insurance verification
Finance	Portfolio analysis, risk profiles, market data, KYC/AML checks, scenario simulation
Legal	Case law search, case files, court deadlines, legislation lookup, fee calculator
Industrial	Equipment diagnostics, parts inventory, maintenance scheduling, safety reporting
Education	Curriculum search, progress tracking, exercise generator, resource library, scheduling

Table 7: Built-in domain tools for tool-use training data generation.

5 Privacy & Compliance Framework

5.1 Design Principle: Privacy by Architecture

UNCASE does not attempt to "anonymize data well enough." Instead, it ensures **real data never reaches the generation or training stages**:

1. **Ingest**: Raw conversations enter Layer 0.
2. **Scan**: Dual-layer PII detection (regex + NER) identifies all personal data.
3. **Replace**: PII tokens replace real data (`[PERSON]`, `[EMAIL]`, etc.).
4. **Extract**: Structural metadata is extracted into a SeedSchema—a blueprint, not data.
5. **Generate**: Layer 3 creates entirely new conversations from the blueprint.
6. **Verify**: Layer 2 re-scans generated output, rejecting anything with residual PII.

The generated synthetic conversations **never contained real PII in the first place**—they were created from anonymized blueprints. This is fundamentally different from generating data and then trying to remove PII afterward.

5.2 Privacy Gateway

The LLM Gateway intercepts all messages going to and from external LLM providers, scanning for PII in three modes:

- `audit`: Scan and log detections (for monitoring).

- `warn`: Scan, log, and include warnings in the API response.
- `block`: Reject any request or response containing PII.

All provider API keys are **Fernet-encrypted at rest** in the database.

5.3 Compliance Profiles

Five regulatory frameworks are implemented as **frozen dataclass configurations**—immutable, auditable, version-controlled:

Profile	**PII Types**	**ϵ (DP)**	**Retention**	**Key Requirements**
HIPAA	21	≤ 3.0	7 years	RBAC + MFA, BAA, Safe Harbor
GDPR	17	≤ 5.0	1 year	Right to erasure, DPIA, portability
SOX	9	≤ 5.0	7 years	Audit trail, segregation of duties
LFPDPPP	10	Optional	1 year	ARCO rights (Mexico)
EU AI Act	7	≤ 5.0	—	Risk classification, Art. 11 documentation

Table 8: Compliance profile configurations with PII categories, differential privacy budgets, and retention policies.

Each profile specifies:

- Which PII categories must be detected and removed.
- Differential privacy epsilon budget for fine-tuning.
- Data retention periods and auto-deletion schedules.
- Quality metric thresholds (stricter for higher-risk domains).
- Required access control and audit mechanisms.

6 Enterprise Infrastructure

6.1 API Architecture

The UNCASE backend exposes **106 REST API endpoints** across **24 routers**, built on FastAPI with async PostgreSQL (via asyncpg + SQLAlchemy):

Category	Endpoints	Capabilities
Authentication	3	JWT login, token refresh, verification
Organizations	7	CRUD, members, settings, API keys
Seeds	6	CRUD, batch operations, domain filtering
Generation	1	Seed-guided synthetic conversation creation
Evaluation	4	Single/batch evaluation, quality reports
Templates	5	11 format renderers, preview, conversion
Tools	8	CRUD, execution, simulation, search
Providers	6	LLM provider CRUD, connection testing
Connectors	8	WhatsApp, webhook, PII scan, imports
Gateway	2	Privacy-intercepted LLM chat
Sandbox	5	E2B code execution, demos, Opik eval
Plugins	7	Install, uninstall, registry, marketplace
Pipeline	1	End-to-end orchestration
Jobs	3	Background job queue, cancel
Knowledge	5	Document upload, vector search
Usage	4	Metering, analytics, cost breakdown
Webhooks	8	Subscriptions, deliveries, retry
Audit	1	Compliance trail, export
Costs	3	LLM spend per org/job
Health	3	Liveness, readiness, deep checks

Table 9: API endpoint summary by category.

6.2 Authentication & Authorization

- **JWT access + refresh token pair** with configurable expiration and rotation.
- **Role-Based Access Control (RBAC)**: Admin, Developer, Viewer roles per organization.
- **Organization-scoped isolation**: All queries filtered by `org_id`.
- **API key authentication**: Fernet-encrypted keys with scoped permissions.
- **Argon2 password hashing** (memory-hard, timing-safe).

6.3 Audit Logging

An immutable compliance trail records:

- Who accessed what data, when, and from where (IP, user agent).
- All CRUD operations on seeds, conversations, providers, and organization settings.
- Authentication events (login, logout, token refresh, failed attempts).
- Pipeline runs with full input/output metadata.

Audit logs are stored in a dedicated PostgreSQL table (`audit_logs`) with separate retention from application data.

6.4 Rate Limiting

Per-key sliding window rate limiting with 4 tiers:

Tier	Requests/min	Target
Free	60	Open-source users, evaluation
Developer	300	Active development, testing
Enterprise	1,000	Production workloads
Default	120	Unclassified API keys

Table 10: Rate limit tiers.

Two backends: in-memory sliding window (single instance) and Redis sorted sets (distributed). Automatic fallback to in-memory when Redis is unavailable. Returns standard `429 Too Many Requests` with `Retry-After` header.

6.5 Observability

- **Prometheus metrics** at `/metrics`: request rate, latency (avg, p95, p99), error rate, database query duration, LLM API latency.
- **Pre-built Grafana dashboard** (included in repository): real-time monitoring of all API and infrastructure metrics.
- **Structured logging** via structlog (JSON format) with contextual fields (seed ID, domain, organization, etc.).
- **Usage metering**: Fire-and-forget event recording for analytics and billing.
- **LLM cost tracking**: Per-organization and per-job spend tracking for all LLM API calls.

6.6 Background Job System

Long-running operations (generation, evaluation, training) run as background jobs with:

- Job submission via API with unique job IDs.
- Real-time progress tracking (percentage, stage, estimated completion).
- Cancellation support.
- Automatic retry with exponential backoff.
- Job history and result storage in PostgreSQL.

7 Dashboard & Developer Experience

7.1 Web Dashboard

A full-featured React 19 dashboard (Next.js 16, TypeScript, shadcn/ui, Tailwind CSS 4) provides a visual interface for the entire pipeline:

Page	Functionality
Overview	Pipeline status, key metrics, recent activity at a glance
Pipeline	Seed-to-model workflow wizard with step-by-step progress
Conversations	Browse, search, and inspect generated synthetic conversations
Templates	Select and preview fine-tuning export formats
Tools	Browse, create, and test domain tools
Evaluations	Quality metric reports with drill-down by metric
Knowledge	Upload and search knowledge base documents
Activity	Audit log browser with filtering
Settings	Organization configuration, members, API keys
Plugins	Plugin marketplace, install/uninstall, per-domain packs
Jobs	Background job queue with real-time status
Costs	LLM API spend tracking per organization and per job

Table 11: Dashboard pages and their functionality.

7.2 CLI

A Typer-based CLI provides full pipeline access from the terminal:

```
# Create a seed from a raw conversation
uncase seed create --domain automotive.sales --file chat.txt

# Generate 1,000 synthetic conversations
uncase generate --seed-id abc123 --count 1000 --model gemini-2.0-
    flash

# Evaluate quality
uncase evaluate --conversation-id xyz789

# Run the full pipeline
uncase pipeline run --domain medical.consultation --count 500

# Export to Llama format
uncase template render --format llama --output training_data.jsonl
```

7.3 Python SDK

For programmatic access, the SDK provides 6 wrapper classes:

```
from uncase import Pipeline, SeedEngine, Generator, Evaluator

# End-to-end pipeline
pipeline = Pipeline(api_url="http://localhost:8000")
result = pipeline.run(
    domain="automotive.sales",
    raw_conversations=["chat1.txt", "chat2.txt"],
    count=1000,
    model="claude-sonnet-4-6",
)
```

```
# Or use individual components
engine = SeedEngine()
seed = engine.create_seed(raw_text, domain="medical.consultation")
generator = Generator(model="gemini-2.0-flash")
conversations = generator.generate(seed, count=100)
evaluator = Evaluator()
reports = evaluator.evaluate_batch(conversations, seed)
```

8 Deployment Options

8.1 Three Installation Paths

Method	Command	Best For
Git + uv	`git clone && uv sync`	Development, contribution
pip	`pip install uncase[all]`	Integration into existing projects
Docker	`docker compose up -d`	Production deployment

Table 12: Installation methods.

8.2 Docker Compose Services

Service	Port	Profile	Purpose
api	8000	default	FastAPI REST API
postgres	5433	default	PostgreSQL 16 (primary datastore)
redis	6379	default	Rate limiting, caching
dashboard	3000	default	React 19 web UI
mlflow	5000	`ml`	ML experiment tracking
api-gpu	8001	`gpu`	GPU-accelerated API (NVIDIA CUDA)
prometheus	9090	`observabilit`	Metrics collection
grafana	3001	`observabilit`	Dashboards & alerting

Table 13: Docker Compose services and profiles.

```
# Standard deployment (API + DB + Redis + Dashboard)
docker compose up -d

# With ML tracking
docker compose --profile ml up -d

# With GPU support
docker compose --profile gpu up -d

# Full observability stack
```

```
docker compose --profile observability up -d
```

8.3 GPU Deployment for Fine-Tuned Models

Production deployment scripts support:

- **Auto-detection**: Identifies GPU type (A40, A100, H100, RTX 5090) and configures memory limits automatically.
- **Tensor parallelism**: Distributes models across multiple GPUs for larger models (32B+).
- **LoRA merge pipeline**: Downloads base model + LoRA adapter, merges weights, and serves via vLLM.
- **Cloudflare Tunnel**: Optional public access via fixed domain (e.g., `api.domain.com`).
- **Health monitoring**: Automatic service health checks with retry logic and graceful shutdown.

9 Codebase Metrics

Metric	Value	Notes
Python source files	203	Backend framework
Python LOC	36,638	Excluding tests
TypeScript/React components	132	Dashboard frontend
Frontend LOC	62,200	TS/TSX combined
API endpoints	106	Across 24 routers
Pydantic models	93	Data validation
SQLAlchemy models	18	Database schema
Alembic migrations	13	Schema evolution
Test files	80	Unit + integration + privacy
Test functions	1,160	Automated test cases
Compliance profiles	5	HIPAA, GDPR, SOX, LFPDPPP, AI Act
Curated seeds	150	3 domains (50 each)
Scenario templates	56	6 industry verticals
Domain tools	30	5 per industry
Export formats	11	Fine-tuning templates
Official plugins	6	One per industry
SDK wrapper classes	6	Programmatic API
Docker services	8	3 optional profiles

Table 14: Current codebase metrics as of March 2026.

10 Use Cases

10.1 Automotive: Dealership AI Assistant

Problem: A national dealership network wants to fine-tune a conversational AI assistant for their sales team, trained on patterns from their best-performing salespeople. Their CRM contains 50,000+ real customer conversations that cannot leave the dealership's infrastructure due to financial data regulations.

UNCASE Solution:

1. Export 500 representative conversations from the CRM.
2. UNCASE Layer 0 strips all customer PII and creates 500 seeds.
3. Layer 3 generates 10,000 synthetic conversations, each maintaining the sales methodology and domain knowledge.
4. Layer 2 evaluates quality (ROUGE-L, factual fidelity, tool-call correctness for inventory/pricing tools).
5. Layer 4 produces a LoRA adapter fine-tuned on Qwen 3-14B.
6. Deploy via vLLM with the included GPU scripts.

Result: Domain-specific AI assistant trained on realistic patterns, zero customer data exposure, full audit trail for compliance.

10.2 Healthcare: Medical Consultation Training

Problem: A health-tech startup needs to train a triage assistant that handles patient intake calls. HIPAA requires that no Protected Health Information (PHI) is used in model training without formal de-identification.

UNCASE Solution:

1. Use the HIPAA compliance profile ($\epsilon \leq 3.0$, 21 PHI categories).
2. Import anonymized consultation transcripts via the WhatsApp/CSV connectors.
3. Generate 5,000 synthetic consultations with the medical domain seed package (50 seeds, 10 scenario templates).
4. Semantic fidelity ensures medical terminology and triage protocols are preserved.
5. Export in Llama 4 format for fine-tuning.

10.3 Finance: Compliance-Safe Advisor Training

Problem: A wealth management firm needs conversational AI for portfolio reviews and risk assessments. SOX and GDPR require 7-year audit trails and strict data governance.

UNCASE Solution:

1. Use SOX + GDPR compliance profiles simultaneously (intersect requirements).
2. Import advisor-client conversations with financial PII removal (9 categories including SSN, bank accounts, credit cards).
3. Generate synthetic conversations with the finance domain pack, including KYC/AML and market panic edge-case scenarios.
4. Audit logging captures every step for SOX compliance.
5. Cost tracking monitors LLM API spend per training run.

11 Competitive Positioning

Capability	UNCASE	Gretel	Mostly AI	Tonic	DIY Scripts
Conversational data focus	✓	×	×	×	Partial
Multi-industry seeds	✓	×	×	×	×
9 quality metrics	✓	Partial	×	×	×
LLM-as-Judge eval	✓	×	×	×	×
Tool-use training data	✓	×	×	×	×
11 export formats	✓	×	×	×	Manual
Compliance profiles	✓	Partial	Partial	✓	×
Open source	✓	×	×	×	✓
Self-hostable	✓	×	×	×	✓
Built-in LoRA pipeline	✓	×	×	×	Manual

Table 15: Competitive comparison. UNCASE is the only solution purpose-built for synthetic conversational training data in regulated industries.

Key differentiators:

1. **Conversation-native**: Built from the ground up for multi-turn dialog, not tabular data.
2. **Seed-based generation**: Preserves real-world patterns without exposing real data.
3. **Quality-first**: 9 metrics with hard thresholds—no conversation reaches training without certification.
4. **Compliance-ready**: 5 regulatory profiles, audit logging, encryption at rest.
5. **Open source**: Full transparency, self-hostable, no vendor lock-in.
6. **End-to-end**: From raw conversations to trained LoRA adapter in a single pipeline.

12 Roadmap

Timeline	Milestone	Description
Q1 2026	Completed	5-layer pipeline, 106 endpoints, 150 seeds, 5 compliance profiles, semantic evaluation, prompt shield, parallel pipeline, GPU deployment
Q2 2026	Formal DP-SGD	Opacus integration with certified epsilon accounting during fine-tuning
Q2 2026	Benchmark publication	Validation against public conversation datasets with reproducible results
Q2 2026	Model marketplace	Share trained LoRA adapters (not data) between organizations
Q3 2026	Additional domains	Real estate, insurance, customer service, e-commerce seed packages
Q3 2026	SOC 2 Type I	Security audit and compliance certification
Q4 2026	Multi-modal support	Image + text conversation training data
Q4 2026	Kubernetes operator	Native K8s deployment with auto-scaling

Table 16: Development roadmap.

13 Technical Stack Summary

Component	Technology	Purpose
Language (backend)	Python ≥ 3.11	Core framework
Language (frontend)	TypeScript 5.9 (strict)	Dashboard UI
API framework	FastAPI + Uvicorn	Async REST API
Frontend framework	Next.js 16, React 19	Dashboard & landing page
UI components	shadcn/ui + Radix UI	Accessible component library
Styling	Tailwind CSS 4	Utility-first CSS
Validation	Pydantic v2	Schema enforcement
Database	PostgreSQL 16 (async)	Primary datastore
Cache / Rate limit	Redis 7	Distributed rate limiting
ORM	SQLAlchemy 2.0 (async)	Database access
Migrations	Alembic	Schema evolution
LLM interface	LiteLLM	Provider-agnostic LLM calls
ML training	Transformers + PEFT + TRL	LoRA fine-tuning
ML tracking	MLflow	Experiment logging
PII detection	Presidio + spaCy	Named entity recognition
Logging	structlog	JSON structured logging
CLI	Typer	Command-line interface
Monitoring	Prometheus + Grafana	Metrics & dashboards
Containerization	Docker Compose	Multi-service deployment
Model serving	vLLM	GPU inference server
Security	Fernet, Argon2, PyJWT	Encryption, hashing, auth

Table 17: Complete technology stack.

14 Getting Started

```
# Clone and install
git clone https://github.com/uncase-ai/uncase.git
cd uncase && uv sync --extra all

# Start the API + database
docker compose up -d

# Run the pipeline with curated seeds
uv run uncase pipeline run \
  --domain automotive.sales \
  --count 1000 \
  --model gemini-2.0-flash

# Or use pip
pip install "uncase[all]"
```

Full documentation: https://uncase.md

Source code: https://github.com/uncase-ai/uncase
API reference: https://app.uncase.md/docs

15 Conclusion

The adoption of AI in regulated industries is constrained not by model capability, but by **the availability of safe, high-quality training data**. Organizations that need domain-specific conversational AI are caught between the legal impossibility of using real customer data and the inadequacy of generic synthetic data.

UNCASE resolves this tension with a principled, layered architecture:

1. Real conversations are **never used directly**—only their structural blueprints survive past the privacy boundary.
2. Every piece of generated data is **automatically evaluated** against 9 quality dimensions with hard rejection thresholds.
3. The entire pipeline—from ingestion to trained model—is **auditable, reproducible, and compliant** with 5 major regulatory frameworks.
4. The system is **open-source and self-hostable**, eliminating vendor lock-in and enabling deployment behind enterprise firewalls.

With 203 Python source files, 106 API endpoints, 1,160 automated tests, 150 curated domain seeds, and 5 compliance profiles already implemented, UNCASE is production-ready infrastructure for the next generation of privacy-safe, domain-specific AI.