# Winnex Tracer Series

## Mathematical-Proof Retrieval for Regulated Domains

**Government · Legal · Healthcare · Finance**

**A product family of the Winnex AI stack — built on the Winnex Madhava Engine**

**Pre-Patent Technical Specification — Zenodo Submission**

| Field | Value |
|---|---|
| **DOI** | 10.5281/zenodo.21826158 |
| **License** | Business Source License 1.1 (BSL 1.1) — source-available, not OSI open-source |
| **Authors** | Klenio Araujo Padilha · WINNEX BRASIL SOLUCOES EMPRESARIAIS LTDA - ME |
| **CNPJ** | 58.364.637/0001-47 |
| **Contact** | pay@winnex.ai |
| **Zenodo** | https://zenodo.org/records/21826158 |
| **Winnex stack** | 10.5281/zenodo.20970487 (Winnex Madhava Cascade) · 10.5281/zenodo.19630736 (Winnex AI Inference Stack) |

## Abstract

This document is the pre-patent technical specification for the **Winnex Tracer Series** — four **source-available** products of the **Winnex AI stack**: **Tracer-GOV**, **Tracer-JUS**, **Tracer-MED**, and **Tracer-GAP**. All four are built on the **Winnex Madhava deterministic vector-search engine** and share the **tracer-gov audit framework** from Winnex. Each product provides:

1. **Soundness-guaranteed retrieval** — every excluded document carries a per-document **Cauchy-Schwarz exclusion bound** proving it could not be in the top-K results *by vector similarity* (0 bound violations, by construction).
2. **A tamper-evident evidence chain** — every search, result, and proof is recorded in an append-only, SHA3-256 hash-chained store; tampering is detected (scope and limits in §3.2).
3. **Compliance-ready reports** — self-assessment templates mapped to the regulatory frameworks of each domain.

The engine is published on **PyPI as** `winnex-madhava` (native C++20 core). Real-data benchmarks on **Kaggle** across all four products report **recall@10 = 1.0000 with 0 bound violations**. Section 6 states plainly what these numbers do and do **not** show: at the published scale the system

evaluates every document for every query (a linear scan with per-document bounds), so the benchmarks demonstrate **correctness and auditability**, not a sublinear speed advantage. Demonstrating the efficiency gain at production scale (millions of documents) is explicitly future work.

---

## 1. The Winnex Stack (positioning)

The **Winnex Tracer Series** is a product family of the **Winnex AI stack**. It composes two already-published Winnex layers and adds the domain layer:

| Layer | Component | DOI / Link |
|---|---|---|
| Application / inference stack | Winnex AI Inference Stack | 10.5281/zenodo.19630736 |
| Deterministic vector engine | Winnex Madhava (Cascade) | 10.5281/zenodo.20970487 · PyPI `winnex-madhava` |
| **Product family (this document)** | **Winnex Tracer Series** | **10.5281/zenodo.21826158** |

Tracer does not re-implement retrieval or audit: it composes the **Winnex Madhava** engine (the proof) and the **tracer-gov** audit framework (the evidence chain) with a domain layer (GOV / JUS / MED / GAP). Every product therefore inherits the same proof engine, the same zero-violation property, and the same evidence chain, while adding only domain-specific metadata, embeddings, and compliance templates.

---

## 2. The Problem

### 2.1 Retrieval without proof

Vector-similarity search is used for information retrieval in every domain. In **regulated markets** (government, legal, healthcare, finance), a missed record is not an inconvenience — it is a **material risk**: an unproven exclusion, a lost evidence item, a missed risk factor, a patient-safety gap.

The dominant ANN techniques — **HNSW** (graph), **IVF** (clustering), **PQ** (quantization) — discard candidates by **probabilistic heuristic**. They cannot **prove** that what was discarded was irrelevant. In a courtroom, a regulator, or a clinical review, "the algorithm decided so" is not an acceptable answer.

### 2.2 The three structural gaps

| # | Gap | Consequence |
|---|---|---|
| 1 | **No proof of exclusion** | Cannot answer "did you miss a relevant document?" |
| 2 | **No immutable audit trail** | Cannot demonstrate who searched what, when, with what guarantee |
| 3 | **No compliance-ready output** | Cannot easily map a search to a regulatory requirement |

---

## 3. The Innovation

The Winnex Tracer Series replaces **probability with proof**, and adds the audit and compliance layers that regulated domains require.

### 3.1 Soundness guarantee (the mathematical core)

For a query vector `q` and a document vector `x` in `R^D` with `||x||=||q||=1`, and an orthogonal projection `P` (QR-orthogonal, e.g. from the Stiefel manifold), the Cauchy-Schwarz inequality gives a per-document **upper bound** on the true cosine:

```
cos(x, q) = <x, q>
          <= <Px, Pq> + ||x_perp|| · ||q_perp||        (B₁, upper bound)

x_perp = x − PᵀP x   (projection residual)
```

A document is **provably excluded** when its upper bound falls below the top-K threshold. By the inequality, such a document **mathematically cannot** be in the true top-K. This is the **soundness guarantee**: 0 bound violations by construction.

The engine implements a **cascaded two-stage scheme**: stage-1 (wide projection) prunes candidates; stage-2 (tighter projection) re-evaluates survivors, shrinking the residual and the uncertainty circle.

**Scope note — what is and is not claimed.** The pruning *capability* exists, but the published benchmarks do **not** exercise it. At the corpus sizes measured (187–2,000 documents) the system evaluates **every document for every query** — `bound pairs = |corpus| × queries`. The published numbers therefore correspond to a **linear scan with per-document bounds**: they prove **correctness**, not **sublinearity**. A practical efficiency advantage over brute force must be demonstrated at scales where pruning becomes active; that benchmark is planned, not yet published (§6.3).

### 3.2 The audit layer (evidence chain)

Every search, result, and proof is written to an **append-only, hash-chained evidence store**:

```
event_1 = { payload, prev_hash = "" }
event_2 = { payload, prev_hash = H(event_1) }
...
```

- Append-only JSONL, **SHA3-256 hash chain**, `fsync` before acknowledge.
- Any modification of a past event breaks every subsequent link — tampering is **detected**.
- The self-audit log is **HMAC-SHA3-256** chained (forgery-resistant when the HMAC key is set).

**Terminology.** We describe this store as **WORM (Write Once, Read Many) /tamper-evident**, which is what the hash chain mathematically guarantees. It is **not tamper-proof** in the absolute sense: a party with root access to the host (or control of the HMAC key) can rewrite the whole chain consistently. Genuine tamper-proofness requires external witnesses — timestamping authorities (RFC 3161), public anchoring, or hardware attestation — which are on the roadmap but not part of the current implementation. HMAC key management is a single point of failure and is addressed in the operational manual, not in this document.

This supports **chain of custody**, **auditability**, and **non-repudiation** (digital signatures: Ed25519 / ICP-Brasil).

### 3.3 Compliance-ready output

Each product maps a search to the regulatory frameworks of its domain, as **self-assessment templates** (explicitly not certifications):

| Product | Frameworks |
|---|---|
| Tracer-GOV | LGPD Art. 20, LAI 12.527/2011, TCU, CGU, GDPR Art. 22, FOIA |
| Tracer-JUS | LGPD, LAI, TCU, CGU (legal), GDPR |
| Tracer-MED | LGPD (health), GDPR Art. 9, HIPAA |
| Tracer-GAP | SOX, SEC, CVM, LGPD, GDPR (financial) |

### 3.4 Numerical rigor (float32)

The engine computes in IEEE-754 **float32**. In floating-point arithmetic a Cauchy-Schwarz bound can, in principle, be violated by a rounding error ε: `<Px,Pq> + ‖x_perp‖·‖q_perp‖` may round such that a document whose true bound is *just below* the threshold is reported *just above*, or vice-versa. The benchmarks report 0 violations; that is expected in float32 for the well-conditioned datasets used, but it is **not a guarantee** across all inputs.

A production implementation must make the guarantee rigorous against rounding by:

- adding an **ε-slack** to the bound (exclude only when `bound + ε < threshold`), with ε derived from an error analysis of the dot products and projections; and/or
- using **interval arithmetic** for the final comparison.

The Winnex engineering roadmap includes making the ε-slack explicit and documenting the error analysis per metric (cosine vs. L2). This document does **not** claim that a float32 bound is exact without such a safety margin.

---

## 4. The Winnex Madhava Engine

### 4.1 On PyPI

The engine is published as `winnex-madhava` on **PyPI**:

```
pip install winnex-madhava
```

- Native **C++20 core** (pybind11 bindings), float32, cosine/L2 metrics.
- QR-orthogonal projections, Cauchy-Schwarz bounds, cascaded stages.
- Reports per query: `bound_violations` (must be 0) and `bound_pairs` (the number of (document, bound) evaluations = the audit record size).

### 4.2 Repository

- **Source**: github.com/winnex-ai/winnex-madhava (also mirrored at github.com/klenioaraujo/winnex-madhava)
- **Pre-patent spec**: Winnex Madhava Cascade, DOI 10.5281/zenodo.20970487
- **License**: BSL 1.1

### 4.3 Why the engine is embedded in every product

The products do **not** re-implement the proof. They depend on `winnex-madhava` (pip) and `tracer-gov` (the shared Winnex audit framework), and add only the domain layer. This ensures:

- One **mathematically verified** proof engine across all domains.
- One **audit framework** (WORM, signatures, self-audit log).
- Consistency of the guarantee — the same bound, the same zero-violation property, in every product.

---

## 5. The Four Products

### 5.1 Tracer-GOV — Government Audit

- **Purpose**: audited vector search for public-sector bodies with mathematical proof of exclusion.
- **Repository**: github.com/winnex-ai/tracer-gov — **private**, source-available; Winnex will make it public at an opportune time.
- **Domain**: government, oversight (TCU, CGU), transparency (LAI), data protection (LGPD).

- **Key features**: government metadata, WORM + hash chain, ICP-Brasil / Ed25519 signatures, retention policies, localized reports (8 jurisdictions, 9 jurisdiction policies with hierarchical inheritance).

### 5.2 Tracer-JUS — Legal e-Discovery

- **Purpose**: legal e-Discovery and research with soundness guarantee — no document **within the proven similarity bound** is lost in triage (§3.1 states precisely what the proof does and does not guarantee).
- **Repository**: github.com/winnex-ai/tracer-jus — **private**, source-available; Winnex will make it public at an opportune time.
- **Domain**: law firms, courts, discovery teams.
- **Key features**: evidence chain (WORM), case metadata (docket, court, judge, parties), legal compliance reports, e-Discovery triage with proof.

### 5.3 Tracer-MED — Clinical Triage

- **Purpose**: clinical triage and research with soundness guarantee — no clinical record **within the proven similarity bound** is lost (§3.1).
- **Repository**: github.com/winnex-ai/tracer-med — **private**, source-available; Winnex will make it public at an opportune time.
- **Domain**: hospitals, clinical research, regulatory review.
- **Key features**: biomedical embeddings (BlueBERT), structured metadata (CID-10, document type, date, **pseudonymized patient**), **HL7 FHIR** interoperability, health-data compliance (LGPD/GDPR Art. 9/HIPAA).

### 5.4 Tracer-GAP — Financial Gap Identification

- **Purpose**: gap identification in financial documents (insurance, banking, capital markets) with proof.
- **Repository**: github.com/winnex-ai/tracer-gap — **private**, source-available; Winnex will make it public at an opportune time.
- **Domain**: risk, compliance, due diligence, disclosure review.
- **Key features**: SEC EDGAR 10-K loader, structured metadata (sector, instrument, CIK, period, risk category), SOX/SEC/CVM reports.

---

## 6. Benchmarks (real data, reproducible on Kaggle)

All benchmarks use the **real** `winnex-madhava` **engine (pip)** and **real datasets** — no synthetic data. Ground truth is fair per method: Madhava → `search_exact`; HNSW/IVF → cosine full scan.

| Product | Dataset (real) | Method | recall@10 | Bound viol. | Bound pairs | ms/ query |
|---|---|---|---|---|---|---|
| **Tracer-GOV** | AG News (2,000 articles, 384d) | HNSW | 1.0000 | n/a | — | ~1.7 |
| | | IVF | 0.9780 | n/a | — | ~0.2 |
| | | **Madhava** | **1.0000** | **0** | **200,000** | **0.14** |
| **Tracer-JUS** | Supreme Court of India (2,000 judgments, 512d) | HNSW | 1.0000 | n/a | — | ~4.5 |
| | | IVF | 0.9860 | n/a | — | ~0.5 |
| | | **Madhava** | **1.0000** | **0** | **100,000** | **0.23** |
| **Tracer-MED** | MTSamples (2,000 transcriptions, BlueBERT 768d) | HNSW | 0.9980 | n/a | — | ~4.7 |
| | | IVF | 0.9920 | n/a | — | ~1.1 |
| | | **Madhava** | **1.0000** | **0** | **100,000** | **0.20** |
| **Tracer-GAP** | SEC 10-K (187 risk-factor sections, 512d) | HNSW | 1.0000 | n/a | — | ~0.1 |
| | | IVF | 0.9880 | n/a | — | ~0.08 |
| | | **Madhava** | **1.0000** | **0** | **9,350** | **0.08** |

### 6.1 What `bound pairs` means (read the numbers honestly)

The `bound pairs` column is the number of (document, bound) evaluations per benchmark. In all four rows it equals `|corpus| × number_of_queries`:

| Product | Corpus | Queries | Bound pairs | Interpretation |
|---|---|---|---|---|
| Tracer-GOV | 2,000 | 100 | 200,000 | every document evaluated per query |
| Tracer-JUS | 2,000 | 50 | 100,000 | every document evaluated per query |
| Tracer-MED | 2,000 | 50 | 100,000 | every document evaluated per query |
| Tracer-GAP | 187 | 50 | 9,350 | every document evaluated per query |

At this scale the cascade does **not** prune: the search is a full linear evaluation with per-document bounds. Consequently:

- `recall@10 = 1.0000` is **trivially achieved** — it is equivalent to exact search, which any correct method reaches at 100% evaluation.
- `0 bound violations` is **expected by construction** (the bound is an inequality, and every document is evaluated).

- The **efficiency advantage** that motivates a cascaded/projection scheme is **not demonstrated** by these numbers.

The benchmarks are a **correctness and auditability demonstration at small scale**, not an efficiency demonstration. That is their honest value — and it is the basis of the regulatory story: an auditor can verify that every exclusion carries a proof.

### 6.2 Benchmark links and datasets (public, reproducible)

| Product | Kaggle notebook |
|---|---|
| Tracer-GOV | kaggle.com/code/kleniopadilha/tracer-gov-benchmark-provable-exclusion-vs-ann |
| Tracer-JUS | kaggle.com/code/kleniopadilha/tracer-jus-benchmark-legal-e-discovery-with-proof |
| Tracer-MED | kaggle.com/code/kleniopadilha/tracer-med-benchmark |
| Tracer-GAP | kaggle.com/code/kleniopadilha/tracer-gap-benchmark |

| Product | Dataset | Source |
|---|---|---|
| Tracer-GOV | AG News (news articles) | Hugging Face `ag_news` |
| Tracer-JUS | OpenNyaya Indian Supreme Court Judgments | Kaggle `gaurav41/opennyaya-indian-supreme-court-judgement-clean` |
| Tracer-MED | MTSamples (medical transcriptions) | Kaggle `tboyle10/medicaltranscriptions` |
| Tracer-GAP | SEC EDGAR 10-K filings | Kaggle `pranjalverma08/sec-edgar-annual-financial-filings-2021` |

### 6.3 Honest reading of the metrics

- **What the benchmarks show:** the engine is *correct* (recall = exact search) and *auditable* (per-document proof per exclusion) on real data, reproducible on Kaggle.
- **What they do not show:** sublinear search, or an efficiency gain over brute force at these corpus sizes (see §6.1). At 187–2,000 documents, ANN methods (HNSW, IVF, PQ) are at a disadvantage by construction: their graph / clustering overhead dominates, so raw timings are uninformative and the baseline parameters are not tuned.
- **The claim that is not made here:** that Madhava is faster than HNSW/IVF. The differentiator is **provability + auditability**, not speed.
- **Scalability is future work:** demonstrating that the cascade prunes (bound pairs << |corpus| × queries) and remains exact at million-scale corpora requires a dedicated benchmark with a tuned ANN baseline and documented parameters. That benchmark is planned, not yet published.

- **Embedding caveat:** the bound proves exclusion from the *top-K by vector similarity*. It does **not** prove that a document is irrelevant to a domain task (§3.1). Ranking quality remains a function of the embedding model.
- These are **single-corpus measurements**, not universal claims.

## 7. Use Cases

| Product | Use case | What it delivers |
|---|---|---|
| **Tracer-GOV** | Transparency request (LAI) | Proven search + audit trail + report |
| **Tracer-GOV** | Oversight audit (TCU/CGU) | Verified no relevant record missed |
| **Tracer-JUS** | e-Discovery triage | Soundness report + WORM chain of custody |
| **Tracer-JUS** | Case-law research | Audit report attachable to a brief |
| **Tracer-MED** | Clinical research triage | Proof no relevant study missed |
| **Tracer-MED** | Regulatory review (LGPD/GDPR/HIPAA) | Health-data compliance support |
| **Tracer-GAP** | Risk-factor review (10-K) | Proof no relevant risk missed |
| **Tracer-GAP** | Due diligence | Soundness-guaranteed gap search |

## 8. Patent Claims (draft)

The novelty claimed is the **applied combination**, not the bare use of a Cauchy-Schwarz bound (a well-established mathematical tool). The draft claims are scoped to the regulated-domain + audit + proof combination:

1. A retrieval system for a regulated domain that (a) computes a per-document mathematical exclusion bound so that every excluded document carries a proof it could not be among the top-K results, and (b) records each exclusion in a tamper-evident, hash-chained audit trail.
2. The system of claim 1, where the bound is computed via QR-orthogonal projection in reduced dimensionality.
3. The system of claim 1, where each exclusion additionally carries a digital signature and non-repudiation evidence.
4. The system of claim 1, where the audit trail is an append-only, SHA3-256 hash-chained store with fsync-before-acknowledge (tamper-evident).
5. The system of claim 1, applied to a regulated domain chosen from: government (Tracer-GOV), legal e-Discovery (Tracer-JUS), healthcare (Tracer-MED), and financial gap identification (Tracer-GAP).
6. A method of e-Discovery comprising: embedding a legal corpus, building a deterministic index with per-document bounds, running review queries with soundness guarantee, and recording results and proofs in a tamper-evident evidence chain.

> *A prior-art search is part of the pre-filing work. Exact-metric search with certified pruning (VP-trees, cover trees, metric trees, branch-and-bound) and "safe screening" / verified top-k literature use similar bounds; the defensible novelty is the regulated-domain application combined with the audit and compliance layer, not the inequality itself.*

## 9. Prior Art Distinction

| Method | Guarantee | Limitation | Relation to Tracer |
|---|---|---|---|
| HNSW | None (graph heuristic) | Not provable, not deterministic | Baseline in benchmarks |
| IVF | None (clustering) | Not provable | Baseline in benchmarks |
| PQ | None (quantization) | Not provable | Baseline in benchmarks |
| VP-trees / cover trees / metric trees | Exact, with branch-and-bound pruning | Certified exclusion via metric-space bounds — **prior art for "exclude by bound"** | The idea of certified exclusion is well established |
| Safe screening / verified top-k | Certified exclusion via Cauchy-Schwarz / duality | Established in optimization / IR literature | **The core inequality is prior art**; Tracer applies it in a regulated-domain + audit context |
| **Winnex Tracer Series** | **Per-document Cauchy-Schwarz bound + tamper-evident audit + compliance output** | Source-available (BSL 1.1); benchmarks demonstrate correctness at small scale | The applied combination is the claimed contribution |

**Honest positioning.** The Cauchy-Schwarz exclusion bound is a standard mathematical tool with known applications in exact-search pruning and safe screening. The Winnex Tracer Series does **not** claim novelty for the inequality. Its contribution — if any — is the *combination*: a deterministic, auditable, compliance-oriented retrieval product family for regulated domains, where every exclusion is provable and every search is recorded. Whether that combination is patentable is a legal question, not one this document answers.

## 10. Regulatory Compliance Mapping

| Product | Frameworks | Status |
|---|---|---|
| Tracer-GOV | LGPD Art. 20, LAI 12.527/2011, TCU, CGU, GDPR, FOIA | Self-assessment |
| Tracer-JUS | LGPD, LAI, TCU, CGU, GDPR | Self-assessment |

| Product | Frameworks | Status |
|---|---|---|
| Tracer-MED | LGPD (health), GDPR Art. 9, HIPAA | Self-assessment |
| Tracer-GAP | SOX, SEC, CVM, LGPD, GDPR | Self-assessment |

> *All reports are **technical self-assessment templates**, not certifications or legal opinions. Adoption requires institutional and legal review.*

---

## 11. License

All products of the **Winnex Tracer Series** and the **Winnex Madhava** engine are licensed under **Business Source License 1.1 (BSL 1.1)**.

**Terminology.** BSL 1.1 is **source-available, not "open-source"** under the OSI definition: it restricts commercial use until the Change Date. This document therefore describes the products as *source-available*, not open-source.

- **Free for Brazilian government agencies** — always, by design (Additional Use Grant).
- Becomes **GPL v2.0+** automatically on **2036-01-01** (Change Date).
- **Commercial use** requires an agreement with Winnex AI (pay@winnex.ai).
- **Research and development** use under BSL 1.1 (evaluation / non-commercial).

Repositories: - github.com/winnex-ai/winnex-madhava — public - github.com/winnex-ai/tracer-gov — private - github.com/winnex-ai/tracer-jus — private - github.com/winnex-ai/tracer-med — private - github.com/winnex-ai/tracer-gap — private

> *The four Tracer repositories are **private, source-available**; Winnex will make them public at an opportune time. The Madhava engine ( `winnex-madhava` ) is public on GitHub and PyPI.*

---

## 12. Honesty Statement

This document is a **pre-patent technical disclosure**, not a product certification, not investment advice, and not a medical opinion. It states explicitly the limits of what is claimed:

- The benchmarks demonstrate **correctness and auditability at small scale**, not sublinear search or a practical speed advantage (§6.1, §6.3).
- The exclusion bound proves **top-K by vector similarity**, not domain relevance; ranking quality depends on the embedding model (§3.1, §6.3).

- The evidence chain is **tamper-evident**, not tamper-proof against a root-capable attacker or key compromise (§3.2).
- The float32 guarantee requires an explicit ε-slack or interval arithmetic to be rigorous against rounding (§3.4).
- BSL 1.1 is **source-available**, not OSI open-source (§11).
- The mathematical core (Cauchy-Schwarz exclusion) is **prior art**; the claimed contribution is the regulated-domain + audit + compliance combination (§8, §9).

*"Replace probability with proof — in the service of government, law, health, and finance."*