WINNEX AI

Running Agent Instance (RAI) Architecture

Enterprise Multi-Agent Orchestration Framework

with Mathematically Guaranteed Semantic Retrieval

Pre-Patent Technical Specification

DOI: 10.5281/zenodo.TBD

Author: Klenio Araujo Padilha
Organization: Winnex Brasil Solucoes Empresariais LTDA - ME
CNPJ: 58.364.637/0001-47
Contact: pay@winnex.ai

This document describes a patentable orchestration architecture for enterprise AI agents.
All mathematical claims are independently verifiable in the referenced prior art.

1. Abstract

This document describes the Running Agent Instance (RAI) architecture - a mathematically grounded, enterprise-grade multi-agent orchestration framework developed by Winnex AI over 18 months (Dec 2024 - Jun 2026) with zero external capital. The architecture addresses a fundamental gap in enterprise AI deployment: existing agent systems provide no mathematical guarantee of retrieval completeness, no auditable decision trail, and no deterministic multi-agent coordination protocol for regulated environments.

The RAI framework introduces three interconnected innovations: (1) a hierarchical agent taxonomy operating across 10 levels of autonomy (Level 0-9), from system-level orchestration through specialist AI agents to human-in-the-loop validation; (2) the WorkRAI atomic task engine, which decomposes business processes into five primitive task types with mathematically provable execution guarantees; and (3) the Strategy Room protocol, a persistent multi-agent collaboration space with formal facilitator-mediated deliberation and cryptographic audit trails.

Each RAI is backed by the Madhava deterministic vector search engine (documented in Zenodo 10.5281/zenodo.20970487), which provides per-document Cauchy-Schwarz upper bounds on cosine similarity with zero violations across 254M+ query-vector pairs. Every document excluded from search results carries a mathematical proof that it cannot be among the top-K results - transforming retrieval from a probabilistic black box into a mathematically constrained, auditable operation.

The full stack consists of 19 independently deployable microservices orchestrated by an asynchronous unified router, with 7 AI providers in automatic failover chain. The system operates on commodity hardware with <5GB VRAM footprint, eliminating dependency on specialized GPU clusters. This document is structured to serve as a pre-patent technical specification covering the agent architecture, mathematical foundations, orchestration protocols, and enterprise integration patterns.

2. The Global Enterprise AI Agent Problem

Enterprise AI deployment in regulated industries faces a crisis of accountability. Current agent architectures - whether built on LangChain, AutoGPT, CrewAI, or proprietary frameworks - share a common limitation: they treat retrieval as an opaque subroutine whose internal decisions cannot be audited or mathematically verified.

2.1 The Accountability Gap

When a financial analyst requests "find all credit applications with suspicious patterns," the question is not just "which ones did you find" but "can you prove you did not miss any." Existing systems answer the first question with a ranked list. None can answer the second with mathematical certainty.

This distinction is not academic. Under the EU AI Act (Articles 13-15), high-risk AI systems must provide meaningful information about the logic underlying their automated decisions. Under Brazil's LGPD (Article 20), data subjects have the right to request review of automated decisions. Under GDPR (Article 22) and HIPAA Privacy Rule, similar requirements hold. Across all these frameworks, the core requirement is the same: algorithmic accountability demands mathematical proof, not probabilistic assurance.

Requirement	Current Systems	RAI Framework
Per-document exclusion proof	None (HNSW/IVF: black box)	Cauchy-Schwarz upper bound
Deterministic execution	HNSW: non-deterministic	Guaranteed deterministic
Multi-agent coordination	Ad-hoc / no protocol	Strategy Room (formal)
Human-in-loop validation	Manual / no audit	4-layer pipeline
Cryptographic audit trail	Log files only	SHA3-256 hash chain
Cross-jurisdiction compliance	Custom per-region	GDPR + LGPD + HIPAA + EU AI Act

3. The RAI Taxonomy: 10 Levels of Autonomous Agency

The Winnex RAI framework defines a formal taxonomy of agent autonomy across 10 levels (0-9). Each level is defined by: credential requirements, allowable actions, override authority, and audit granularity. This taxonomy ensures that no agent operates beyond its authorized autonomy envelope.

Level	Type	Autonomy	Credentials Required	Override Authority
0	DireX (System)	TOTAL root +	master_key + system_ov	None
1	RAI Assistential	Supervised	api_key	Level 2+ or RH
2	RAI Specialist	Supervised	api_key + tenant_id	Level 7+ or RH
3	PicoClaw Agent	Confined	service_token + secret	Level 7+
4	PicoClaw Advanced	Scoped	+ tenant_access	Level 7+
5	PicoClaw Trusted	Extensive	orchestration_token	Level 7+
6	PicoClaw System	Broad	+ master_key	Level 7+
7	Strategic Agent	TOTAL	system_credential + internal_t	None (advisory)
8	RH Human Specialist	Validator	admin_credential + agent_contr	Human override
9	RH Human Executive	Approver	root_credential + master_ke	Final authority

The fundamental design principle is bounded autonomy: each agent level has a mathematically defined authority envelope that cannot be exceeded without documented override from a higher level. Override events are cryptographically signed and recorded in the audit chain.

3.1 RAI Lifecycle States

Each RAI instance follows a formally defined lifecycle with exactly 7 states. State transitions are atomic and logged. Invalid transitions are rejected at the database level via foreign key constraints.

State	Description	Valid Next States
instantiating	RAI record created, no runtime allocated	ready
ready	Runtime allocated, awaiting first assignment	active, paused
active	Executing WorkRAIs or monitoring	paused, standby, archived
paused	Suspended by operator or system	active, archived
standby	Idle cycle, resources held	active, ready
archived	Lifecycle terminated	(terminal)
failed	Irrecoverable error, audit triggered	archived

State transitions are governed by the RAI Controller service, which validates each transition against: current state machine rules, operator credential level, any active WorkRAI dependencies, and the global status of the Strategy Room associated with this RAI.

4. The WorkRAI Atomic Task Engine

WorkRAI is the universal task primitive in the RAI architecture. Every operation that an agent performs - whether automated or human-supervised - is decomposed into WorkRAI units. The decomposition is mathematically complete: any business process that involves retrieval, reasoning, validation, or notification can be expressed as a composition of the five primitive types.

4.1 The Five Primitives

Type	Executor	Semantics	Input	Output	Audit Level
ai_prompt	AI Agent	Execute an LLM prompt with	prompt + context + con	response + confidence	Full (tokens + retrieval proof)
human_validation	RH Human	Request specialist appro	context + options	approval + justification	Full (signed)
api_call	System	Call external API with ret	endpoint + payload	response + latency	Metadata + timing
data_processing	AI Agent	Transform, filter, or aggrega	data + transform spec	transformed data	Input/output hash
enviar_email	System	Send templated notificat	template + recipients	delivery status	Delivery log

4.2 WorkRAI Execution Pipeline

Every WorkRAI executes through a 4-stage pipeline, regardless of type:

```
Stage 1: Input Validation
  - Schema validation against type definition
  - Credential check (agent level vs. required permissions)
  - Context integrity (hash chain verification)
Stage 2: Sandbox Execution
  - Execute in isolated environment with timeout
  - Collect intermediate metrics
  - Detect anomalies (error rate > 5%, latency > 2s, CPU > 90%)
Stage 3: Validation Gate
  - Automated checklist validation
  - RH human approval if type == human_validation
  - Partner approval if critical flag set
Stage 4: Commit & Audit
  - Write output_data to database
  - Generate audit record: (input_hash, output_hash, decision_tree, timestamp)
  - Append to SHA3-256 chain
  - Notify Cronologia orchestrator
```

4.3 WorkRAI SQL Schema (Simplified)

```
CREATE TABLE workrai (
  id INT AUTO_INCREMENT PRIMARY KEY,
  cronologia_id INT NOT NULL,
  rai_id INT NOT NULL,
  type ENUM('ai_prompt','human_validation','api_call',
```

```
              'data_processing','enviar_email') NOT NULL,
    status ENUM('pending_rh','awaiting_dev','in_progress',
                'completed','failed') NOT NULL,
    input_data JSON NOT NULL,
    output_data JSON,
    audit_proof JSON NOT NULL,
    previous_hash VARCHAR(64) NOT NULL,
    current_hash VARCHAR(64) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (cronologia_id) REFERENCES cronologia_rai(id),
    FOREIGN KEY (rai_id) REFERENCES rais(id)
  );
```

4.4 Cronologia: Multi-Step Process Orchestration

Cronologia_rai is the orchestration entity that groups WorkRAIs into structured business processes. Unlike simple DAG-based workflow engines, Cronologia supports: automatic and human stage interleaving, conditional branching based on WorkRAI outputs, parallel execution across multiple RAIs, timeout and escalation policies, and automatic rollback on critical failure.

The CronologiaOrchestrator is a background daemon that polls every 10 seconds. It advances processes through their state machine:

```
  Status Lifecycle:
    planejamento -> iniciada -> em_andamento -> aguardando_rh -> concluida
                                          -> failed -> rollback_attempted

  Stage Types:
    automatica: Executed by AI agents (Level 1-2)
    humana: Pauses, notifies RH (Level 8), waits for validation
    critica: Requires partner approval (Level 9)
    paralela: Multiple WorkRAIs execute simultaneously
```

The orchestrator maintains a priority queue: WorkRAIs with type=human_validation have highest priority (human time is expensive). ai_prompt tasks have standard priority. api_call and data_processing tasks execute in background threads.

5. The Strategy Room: Multi-Agent Collaboration Protocol

The Strategy Room is a persistent, facilitator-mediated collaboration space where multiple agents (AI and human) work together on complex business problems. Unlike ad-hoc agent chat systems, the Strategy Room follows a formally defined protocol with cryptographic audit of every deliberation step.

5.1 Protocol Structure

Each Strategy Room has exactly one Facilitator agent (DireX, Level 7), who manages the discussion according to a structured agenda. The protocol guarantees that every participant can: express their analysis, respond to other participants' findings, vote on proposed decisions, and formally dissent with recorded justification.

The protocol proceeds through five ordered phases:

```
Phase 1: Context Assembly
  - Facilitator loads RAI context + WorkRAI history
  - Madhava retrieval with mathematical bounds
  - Context hash committed to audit chain
Phase 2: Analysis Round
  - Each specialist agent (RAI Level 1-2) presents analysis
  - Each analysis includes: findings, confidence scores, cited sources
  - Sources verified against Madhava bound thresholds
Phase 3: Deliberation
  - Facilitator identifies points of agreement and disagreement
  - Agents may request additional context (triggers Phase 1 re-run)
  - Human RH (Level 8-9) may provide domain expertise
Phase 4: Decision
  - Facilitator synthesizes three options: recommended, alternative, fallback
  - Each option scored: expected outcome, risk level, resource cost
  - Human RH (Level 8-9) approves or requests revision
Phase 5: Commitment
  - Decision documented with full audit trail
  - Action items generated as WorkRAIs
  - All participants cryptographically sign the decision record
```

5.2 Formal Properties

The Strategy Room protocol guarantees:

- Completeness: Every participant must explicitly state their position. Silence is recorded as abstention.
- Auditability: Every message is logged with: agent_id, level, timestamp, hash chain link. The transcript can be independently verified by any regulator.
- Non-repudiation: Decisions are signed via Ed25519 digital signatures by all participants. Override events require signatures from both the override initiator and the authorizing level.

- Temporal ordering: All events are ordered by a monotonically increasing Lamport clock maintained by the Facilitator, ensuring causal consistency across distributed participants.

5.3 Facilitator Escalation Rules

The Facilitator escalates to Level 9 (Human Executive) when: no consensus reached after 3 deliberation rounds, confidence scores remain below 0.7 for all options, the decision involves regulatory risk exceeding the tenant's defined threshold, or a participant formally dissents with substantive justification.

6. Agent Marketplace: Installation and Supply Chain

The Marketplace is a catalog-based agent distribution system. Each agent profile in the catalog specifies: agent identity (name, specialty, level), workflow definition (stages that become cronologia entries), authentication requirements, pricing model, and compatibility constraints.

6.1 Installation Protocol

When an agent is purchased from the Marketplace, the installation follows a formal FK-constrained protocol enforced at the database level:

```
Step 1: Load agent profile + workflow_definition
Step 2: INSERT INTO rais (agent_profile, tenant_id, level, status='instantiating')
Step 3: INSERT INTO cronologia_rai (rai_id, status='planejamento',
          stages=workflow_definition)
Step 4: For each stage: INSERT INTO workrai (cronologia_id, type,
          status='pending_rh')
Step 5: INSERT INTO strategy_room (rai_id, facilitator='DireX', status='active')
Step 6: UPDATE rais SET status='ready'

Foreign key chain enforced by MariaDB:
  marketplace.id -> rais.agent_profile_id
  rais.id -> cronologia_rai.rai_id
  cronologia_rai.id -> workrai.cronologia_id
  rais.id -> strategy_room.rai_id
```

This FK cascade guarantees referential integrity: no WorkRAI exists without a parent Cronologia, no Cronologia exists without a parent RAI, and no RAI exists without a Marketplace profile. The entire system is referentially sound by construction.

6.2 Agent Provisioning and Resource Isolation

Each RAI instance receives an isolated execution environment with the following resources: dedicated PostgreSQL schema (multi-tenant isolation), 8GB locked RAM for CPU caching (zero-copy inference), configurable LLM provider chain with per-tenant API key encryption, and separate Redis state cache. The resource provisioning is enforced at the operating system level via cgroups, ensuring that no RAI can consume resources allocated to another.

7. Mathematical Foundations of RAI Retrieval

The RAI architecture inherits its retrieval guarantees from the Madhava deterministic vector search engine. Every search operation performed by any RAI carries per-document mathematical proof. This section summarizes the mathematical framework; full proofs are in Zenodo 10.5281/zenodo.20970487.

7.1 The Cauchy-Schwarz Upper Bound

For any orthogonal projection P: R^D -> R^k and any vectors v (corpus), q (query):

```
|<v,q> - <Pv,Pq>| <= ||v - P^T P v|| * ||q - P^T P q||

Therefore:
<v,q> <= <Pv,Pq> + ||v - P^T P v|| * ||q - P^T P q||
```

If the right-hand side (the upper bound) falls below the top-K threshold, the document mathematically cannot be among the top-K results. This is not probabilistic. It is a deterministic consequence of the Pythagorean theorem applied to orthogonal projections.

7.2 QR-JL Orthogonalized Projections

Johnson-Lindenstrauss random projections preserve pairwise distances with high probability but are not orthogonal. Non-orthogonal projections break the Pythagorean guarantee. Madhava solves this by QR-orthogonalizing the random projection matrix:

```
R = randn(d_in, d_out)     # Random matrix (JL)
Q, _ = np.linalg.qr(R)    # QR decomposition
P = Q[:, :d_out].T         # Orthogonal projection
assert ||P @ P.T - I|| < 1e-5  # Production assertion
```

The production code assertion fails explicitly if the orthogonality condition is violated. There is no graceful degradation for mathematical guarantees.

7.3 Per-Document Audit Trail

For every search query executed by a RAI, the Madhava engine produces a complete audit trail. A regulator, auditor, or court can independently recompute every number:

```
DOCUMENT #42042  --  Audit Record
  True cosine:            0.2317
  Projected cosine (64D): 0.2281
  Pythagorean residual:   0.0314
  Cauchy-Schwarz bound:   0.2595
  Threshold (10th result):0.4500
  Verdict: 0.2595 < 0.4500
         -> PROVABLY OUTSIDE TOP-10
  Mathematical certainty: TRUE
  Applicable standard: EU AI Act Art. 13
```

7.4 Empirical Validation

The Madhava engine has been validated across 254M+ query-vector pairs on the SIFT-1M dataset with zero bound violations. The [64,128] cascaded configuration achieves NDCG@10=1.000 and Recall@10=1.000 with build time of 2.57s for 1M vectors on CPU.

Metric	Madhava [64,128]	HNSW (ef=128)	IVF (nprobe=20)
NDCG@10	1.000	1.000	0.987
Recall@10	1.000	1.000	0.980
Bound violations	0 in 254M+	Cannot measure	Cannot measure
Deterministic	Yes	No (random graph)	Yes
Audit trail	Per-document proof	None	None
CPU build (1M)	2.57s	~40s	<1 min

8. Microservice Architecture and Communication

The full Winnex RAI stack consists of 19 independently deployable microservices organized into 4 tiers. Each service exposes a /health telemetry endpoint. The Unified Router uses asynchronous HTTP clients and implements try/catch fallback routing for zero-downtime resilience.

Tier	Function	Key Components
Tier 1: Core Infrastructure	Quantized inference & state persistence	Shimang Engine, GPTQ 4-bit, Redis, PostgreSQL + pgVector
Tier 2: Semantic Pipeline	Context compression & navigation	Qdrant Index Vector DB, HMC Refinement, L4 Cache, Embedding Elite
Tier 3: Reasoning Engine	Cognitive isolation & symbolic processing	Enterprise Reasoning, Math Solver, Probabilistic OCR, Web Search Proxy
Tier 4: Connectivity & Gateways	Routing, security & output synthesis	Unified Router (async), Dual API Gateway, WireGuard, TTS Service

8.1 AI Provider Chain with Auto-Failover

The AI Gateway (winnex_ai_server) routes LLM requests through 7 providers in failover order:

```
wireguard -> winnex_local -> deepseek -> zai -> openai -> anthropic -> google
```

Each provider has: encrypted API key storage (AES-256-GCM), per-tenant quota tracking, automatic failover on timeout/error, and execution logging with latency and token counts. The failover is transparent to the RAI - if DeepSeek is unavailable, Z.ai receives the request automatically without the agent's knowledge.

8.2 Audit Trail Backend

Every WorkRAI execution and Strategy Room decision generates a cryptographic audit record. The audit backend supports three storage modes:

- Mode 1 - Append-Only Hash Chain (Default): SHA3-256 hash chain with <1ms latency. Each entry contains the previous hash, creating a tamper-evident chain. Admissible as business record.
- Mode 2 - Ed25519 Signed Chain: Adds NIST-standard digital signatures. Each entry is signed by the generating agent's private key. Admissible in court.
- Mode 3 - Blockchain Smart Contract: Stores proof hashes on-chain for regulators who explicitly require it. $0.01-0.50 per transaction.

9. Human-Agent Interaction Framework

The RAI architecture is not fully autonomous. It is designed around a principle of graduated autonomy, where human judgment is interleaved at multiple points in the execution pipeline. This section describes the precise mechanisms by which human agents interact with the system.

9.1 The RH Registry (Human Agents)

Human specialists (RHs - Recursos Humanos, Level 8-9) are registered in the system with formal profiles: specialty (Fiscal, Legal, Financial, Technical, Medical, Compliance), experience level (Junior, Mid, Senior, Principal, Executive), availability schedule, current workload capacity, hourly cost, and qualification certifications.

RHs are discovered by the CronologiaOrchestrator based on: specialty match with WorkRAI type, availability (not currently assigned), workload (below capacity threshold), and cost (within tenant budget). Assignment can be automatic (type==human_validation triggers nearest-match RH) or manual (operator selects specific RH).

9.2 Validation Workflow (4-Layer)

Every WorkRAI that requires human validation passes through up to 4 layers:

```
Layer 1: Sandbox (Automatic)
  - Isolated execution with metric collection
  - Anomaly detection: error rate >5%, latency >2s, CPU >90%
Layer 2: Checklist (Automatic)
  - Mandatory technical validations
  - Schema compliance, boundary checks
Layer 3: RH Validation (Human)
  - Manual approval with configurable timeout (default: 4h)
  - RH can: approve, reject with justification, request revision
  - Revision creates a new WorkRAI iteration
Layer 4: Partner Validation (Human, Critical Only)
  - Required when: financial impact > threshold, regulatory
    filing, cross-tenant data access, override of Level 7 decision
```

9.3 Override Hierarchy

The system enforces a strict override hierarchy:

```
Level 9 (Executive) can override: Any Level 0-8 decision
Level 8 (Specialist) can override: Level 1-7 AI agents
Level 7 (Strategic AI) can recommend: Level 0-6 actions
Level 1-6 AI agents: No override authority

All overrides are recorded with:
  - Override initiator ID + level
```

```
    - Target decision ID
    - Justification (free text, required)
    - Cryptographic signature (Ed25519)
```

10. Global Regulatory Compliance Mapping

The RAI architecture is designed from the ground up to satisfy regulatory requirements across multiple jurisdictions. Unlike systems that bolt on compliance after the fact, every architectural decision in RAI originates from a compliance requirement.

10.1 EU AI Act (Effective 2024-2026)

Article	Requirement	How RAI Addresses It
Art. 13	Transparency for high-risk AI systems	Strategy Room produces complete decision trees; every search result has per-document exclu
Art. 14	Human oversight	4-layer validation pipeline with mandatory RH (human) gates at configurable points
Art. 15	Accuracy and robustness	Canary bounds with zero violations; deterministic execution guarantees identical resu
Record-keeping	Automatic logging	SHA3-256 audit chain with cryptographic non-repudiation

10.2 LGPD (Brazil, Art. 20)

Brazil's General Data Protection Law grants data subjects the right to request review of automated decisions. The RAI architecture provides: complete and comprehensible explanation of every search decision, identification of which documents were considered and which were excluded (with proof), ability for a human reviewer to override any AI decision, and full audit trail in Portuguese (locale-aware output).

10.3 GDPR (EU, Art. 22)

GDPR Article 22 grants the right to not be subject to automated decision-making. RAI satisfies this through: meaningful information about the logic of decisions (via Madhava audit trail), human intervention on demand (RH system with SLA guarantees), ability to contest decisions (Strategy Room appeal protocol), and regular accuracy monitoring (automated bound verification).

10.4 HIPAA (US, Privacy Rule)

HIPAA's data minimization and access requirements are addressed by: deterministic search that can reproduce exactly what was retrieved for any past query, mathematical proof that no relevant document was excluded, access logs with full audit trail (required for 6-year retention), and on-premise deployment option for data sovereignty.

10.5 Cross-Jurisdiction Architecture

The RAI framework implements a unified compliance layer that adapts to the regulating jurisdiction per tenant. When a tenant is created, their jurisdiction is recorded. The system then: applies the strictest applicable regulation automatically, generates compliance reports in the required format, enables jurisdiction-specific audit modes, and alerts on regulatory changes that affect the tenant's obligations.

11. Global Enterprise Scenarios

This section presents four concrete scenarios that demonstrate the RAI architecture operating across different industries and regulatory jurisdictions. Each scenario traces the complete flow from trigger through WorkRAI execution to audit-ready resolution.

11.1 Scenario A: Global Bank -- Anti-Money Laundering Alert (UK/EU)

A London-based investment bank receives a regulatory inquiry from the FCA regarding a flagged transaction. Compliance officer activates the AML RAI.

```
1. RAI instantiated via Marketplace: 'Compliance Analyst -- AML'
2. Cronologia created with 5 stages:
   S1 (auto): Retrieve all transactions within 3-degree relationship
     -> Madhava search with mathematical completeness proof
   S2 (auto): Rank by risk score using enterprise risk model
   S3 (human): RH compliance officer reviews top-20 alerts
   S4 (auto): Generate SAR report draft
   S5 (human): Partner validation for FCA submission
3. Strategy Room activated: Facilitator (DireX) + AML RAI + RH Compliance
4. Decision: File SAR with specific transaction pattern description
5. Audit trail: All 14,832 documents from Stage 1 have mathematical
   exclusion proofs. RH signature on Ed25519. FCA-ready report generated.
6. Total: 3.2s ML search + 45m human review + cryptographic audit.
```

Without RAI: The bank would use HNSW-based search. The FCA asks 'how do you know you found all relevant transactions?' The answer is 'the graph returned these.' This is not defensible. With RAI: 'Document #42042 was excluded because its cosine similarity bound (0.2595) is provably below the top-K threshold (0.4500). Here is the mathematical proof.'

11.2 Scenario B: US Hospital -- Clinical Trial Matching (HIPAA)

A multi-state hospital network uses RAI to match patients with clinical trials. The system must prove complete coverage for IRB (Institutional Review Board) compliance.

```
1. RAI 'Clinical Trial Matcher' processes 12,847 patient records
2. For each patient: Madhava search across 3,842 active trials
3. Exclusion criteria verified via Cauchy-Schwarz bound
4. Top-5 matches per patient with confidence scores
5. Each exclusion documented: 'Trial NCT04231 excluded for patient
   #7281 because inclusion criteria bound (0.312) < threshold (0.650)
6. IRB submission includes: complete search audit trail
7. All data stays on-premise (HIPAA requirement) -- CPU-only deployment
```

11.3 Scenario C: Japanese Patent Office -- Prior Art Search

A patent examiner at the Japan Patent Office (JPO) searches 120M+ patents globally. The RAI must

provide proof of search completeness for legal defensibility.

```
1. RAI 'Patent Prior Art Search' processes 120M patents (pre-computed)
2. 3-stage cascade costs: 4 * 120M + 55.2K ops = ~480M operations
   vs. exhaustive: 120M * 128 = 15.36B ops (32x reduction)
3. Search time: ~480ms on CPU-only hardware
4. All 119,999,950 excluded patents carry mathematical exclusions
5. Examiner reviews top-50 candidates, selects 12 relevant
6. Patent office certifies search methodology for court admissibility
```

11.4 Scenario D: Australian Mining Corp -- ESG Compliance

A Sydney-based mining company must prove its environmental disclosures are complete under ASX listing rules. 500K documents across 15 years of operations.

```
1. RAI 'ESG Compliance Auditor' deployed from Marketplace
2. Cronologia: audit reports -> incident logs -> remediation plans
3. For each category: exhaustive retrieval with mathematical proof
4. Human auditors (Level 8) validate a stratified sample (5%)
5. Strategy Room: RAI + internal auditor + external ESG consultant
6. Output: audited disclosure report with per-document proof
7. Board signs off with cryptographic verification
```

12. Intellectual Property and Patent Claims

The RAI architecture contains several independently patentable inventions. This section identifies the novel contributions and maps them to the architectural components described in this document.

12.1 Patent Claims Identified

Claim 1 -- Hierarchical Agent Autonomy Taxonomy (Levels 0-9): A system for classifying and constraining AI agent autonomy through mathematically defined authority envelopes, where each level has cryptographically enforced credential requirements and override provenance tracking. [Sections 3, 9.3]

Claim 2 -- WorkRAI Atomic Task Decomposition Protocol: A method for decomposing arbitrary business processes into five primitive task types (ai_prompt, human_validation, api_call, data_processing, enviar_email) with formally defined execution pipelines and validation gates. [Section 4]

Claim 3 -- Strategy Room Multi-Agent Deliberation Protocol: A facilitator-mediated, cryptographically auditable collaboration protocol for multi-agent decision-making with formal phases, voting mechanisms, dissent recording, and non-repudiable signing. [Section 5]

Claim 4 -- FK-Constrained Agent Lifecycle Cascade: A database-level referential integrity chain (Marketplace -> RAI -> Cronologia -> WorkRAI -> Strategy Room) that guarantees process consistency through foreign key constraints rather than application-level orchestration. [Sections 3.1, 6.1]

Claim 5 -- Cauchy-Schwarz Upper Bound as Audit Primitive: The use of Cauchy-Schwarz inequality applied to QR-orthogonalized Johnson-Lindenstrauss projections as a per-document mathematical exclusion proof in agent-driven retrieval. [Section 7]

Claim 6 -- 4-Layer Validation Pipeline with Auto-Rollback: A multi-layer validation system combining automatic sandboxing, checklist validation, human approval, and partner review with automatic rollback on threshold exceedance. [Section 9.2]

12.2 Prior Art References

All mathematical foundations are validated in prior Zenodo records and Kaggle benchmarks. The orchestration architecture is novel and represents the primary IP contribution.

13. Licensing and Commercial Terms

This document and the described architecture are distributed under the Business Source License 1.1 (BSL 1.1).

The BSL 1.1 permits: free use, modification, and distribution for non-production and testing purposes. Study, academic reference, and independent verification are explicitly allowed without license fee.

The BSL 1.1 requires a separate commercial license for: commercial deployment, production hosting, SaaS offering, or any use where the system processes production enterprise data.

The license includes a Change Date of January 1, 2036, after which the code will convert to GNU General Public License v2.0 or later, ensuring long-term ecosystem transparency while protecting enterprise investment during the active development phase.

IP licensing, technology transfer, and strategic partnership inquiries: pay@winnex.ai

14. References

- Padilha, K. A. (2026). Winnex AI: A Mathematical Anatomy for an Enterprise-Scale Inference Stack. Zenodo. DOI: 10.5281/zenodo.19630736
- Padilha, K. A. (2026). Winnex Madhava Cascade: Mathematically Guaranteed, Interpretable Vector Search. Zenodo. DOI: 10.5281/zenodo.20970487
- Padilha, K. A. (2026). O(K) Navigation Proof: Spectral Filter + QJL + Anchor-Based Retrieval. Zenodo. DOI: 10.5281/zenodo.20856138
- Padilha, K. A. (2026). Corrected O(K) Riemannian HMC Benchmark. Zenodo. DOI: 10.5281/zenodo.20852884
- Padilha, K. A. (2026). Winnex AI Audit Benchmark: Why Mathematical Search Guarantees Matter. Zenodo. DOI: 10.5281/zenodo.21088504
- Padilha, K. A. (2026). Open Letter to Strategic Partners and Investors. Zenodo. DOI: 10.5281/zenodo.21101148
- Padilha, K. A. (2026). Winnex Enterprise Stack: Research Foundation for the Regulated AI Market. Zenodo. DOI: 10.5281/zenodo.21107295
- Padilha, K. A. (2025). Quaternionic-Harmonic Framework (PsiQRH). Zenodo. DOI: 10.5281/zenodo.17171112
- Padilha, K. A. (2026). Lampreia Framework: SO(4) Quaternion Rotations and Spectral Filtering. Zenodo. DOI: 10.5281/zenodo.20754146
- European Union. (2024). Regulation (EU) 2024/1689 -- Artificial Intelligence Act.
- Brazil. (2018). Lei Geral de Protecao de Dados Pessoais (LGPD). Lei No 13.709/2018.
- European Union. (2016). General Data Protection Regulation (GDPR). Regulation (EU) 2016/679.
- U.S. Department of Health and Human Services. (1996). Health Insurance Portability and Accountability Act (HIPAA).
- Malkov, Y. A., & Yashunin, D. A. (2016). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arXiv:1603.09320.
- Dasgupta, S., & Gupta, A. (2003). An elementary proof of the Johnson-Lindenstrauss lemma. International Computer Science Institute.

This document is a technical specification of the Winnex AI Running Agent Instance (RAI) architecture, submitted as a pre-patent publication. All mathematical claims are independently verifiable. Source code is available under BSL 1.1 at github.com/winnex-ai. Commercial licensing and IP inquiries: pay@winnex.ai.

Winnex Brasil Solucoes Empresariais LTDA - ME

CNPJ: 58.364.637/0001-47 | Brazil

Contact: pay@winnex.ai

Document generated July 2026 | License: BSL 1.1