ARTHUR LABS, INC. — TECHNICAL WHITEPAPER

Hyper Intelligent Innovation Engine

A self-governing, ethically-aligned intelligence system for transforming human intent into engineered physical reality — architected for the Mac Mini M4 Pro and distributed through Registered Agentics cloud infrastructure.

CORE CREATOR
Watson Lewis-Rodriguez

ORGANIZATION
Arthur Labs, Inc.

DATE
March 2026

CO-AUTHOR / SPONSOR
Marko Ruble

STATUS
Active Development

VERSION
v1.0.5

INFRASTRUCTURE PARTNER
Registered Agentics
https://registeredagentics.vercel.app/

Contents

ABSTRACT

The Hyper Intelligent Innovation Engine (HIIE) is a novel class of AI system designed to transform raw human intent into physically realizable, ethically validated, and commercially viable engineered outputs. Conceptualized as what one might achieve if CRISPR-GPT and a frontier reasoning model produced a system capable of touching physical space — HIIE combines persistent multi-agent task delegation, recursive domain reasoning, real-world simulation validation, and an embedded ethics governance board. Developed under Arthur Labs, Inc. and created by Watson Lewis-Rodriguez, with infrastructure partnership from Registered Agentics (Marko Ruble), HIIE outputs range from PCB schematics and manufacturing machine designs to custom processor architectures and full factory systems. This whitepaper describes the system architecture, Mac Mini M4 Pro implementation, agent delegation framework, ethics governance model, AI slop prevention methodology, use case catalogue, cloud distribution strategy via Registered Agentics, self-hosted UI toolchain, provisional patent fee structure, phased development roadmap, and the persistent intellectual capital layer enabled by Apple Neural Engine background fine-tuning.

§1 Introduction & Vision

Modern AI systems excel at language, code generation, and analysis within predefined boundaries. What does not yet exist is a system capable of reasoning from an abstract human problem statement through to a physically buildable, commercially patentable, ethically validated engineered solution — autonomously delegating specialized cognitive roles the way a world-class engineering organization would.

HIIE is that system. It is not a chatbot or a code assistant in isolation. It is a **complete innovation pipeline**: taking the rawest form of human intent and, through recursive reasoning, multi-agent collaboration, and physics-grounded validation, producing outputs that can enter the real world.

CORE THESIS

Human innovation is bottlenecked not by ideas, but by the translation layer between intent and engineered reality. HIIE eliminates that bottleneck — operating as if CRISPR-GPT and a frontier reasoning model had a system capable of touching physical space.

Alice and Bob — The HIIE Interaction Model

In the classical cryptographic and distributed systems tradition, **Alice** and **Bob** are the canonical principals in any secure, trust-based protocol. HIIE adopts this framing deliberately. **Alice** is the primary user and directing intelligence — the human who submits intent, approves requirements, and holds final authority over all outputs. **Bob** represents any secondary principal, collaborator, counterparty, or reviewer interacting with Alice's outputs downstream. In practice, Watson Lewis-Rodriguez occupies the Alice role as founder and director; clients, partners, or contract reviewers occupy the Bob role.

This framing is not cosmetic. It reflects HIIE's architectural assumption that **every system must have a verifiable, non-repudiable human principal chain**. Alice's sign-off at every gate is cryptographically analogous to a digital signature — it cannot be forged, delegated to an agent, or bypassed by any instruction.

Guiding Principles

- Ethics above all engineering outputs — no design exits without ethical validation
- Human oversight is non-negotiable — HIIE is directed intelligence, never autonomous in consequence
- Physical feasibility is a hard constraint — theoretical elegance without manufacturability is rejected
- Environmental net-positivity is a default design target, not an afterthought
- Full transparency in every reasoning step — the system explains every recommendation
- AI slop prevention is an architectural requirement — every output is validated, recursive, and human-perspective-tested before delivery

§2 System Architecture Overview

HIIE operates as a layered pipeline. Each layer is a discrete processing stage that feeds both downward into execution and upward into refinement. The architecture is deliberately non-linear — outputs at any stage can trigger re-evaluation upstream.

```
[HUMAN INTENT INPUT -- ALICE]
↓
DIALOGUE ENGINE -- clarify, learn, validate goals
↓
REQUIREMENTS SYNTHESIZER -- structured spec generation
↓
AGENT DELEGATION LAYER -- assign project roles
↓
HCT ECONOMY -- payroll, dynamic pricing, treasury audit
↓ ↓ ↓ ↓ ↓ (parallel execution)
HARDWARE CHEMISTRY CODE PATENT MFG
↓
ANTI-SLOP VALIDATION LAYER (Section 5)
↓
SIMULATION + VALIDATION LAYER
↓
ETHICS BOARD REVIEW
↓
OUTPUT ENGINE -- specs, code, BOM, patents, reports
↓
HUMAN REVIEW + SIGN-OFF (ALICE)
↓
[PHYSICAL REALITY]
```

Every completed project feeds back as a training signal — including into the ANE background fine-tuning layer (§8.9) — improving HIIE's domain accuracy and real-world feasibility calibration over time.

§3 Mac Mini M4 Pro — Hardware Specification

HIIE Phase 1 runs on a single Mac Mini M4 Pro node, hosted at Registered Agentics facilities. Apple Silicon's unified memory architecture eliminates the PCIe bandwidth bottleneck present in traditional CPU/GPU/RAM split configurations.

Target Configuration

Component	Specification
Model	Apple Mac Mini M4 Pro
CPU	12-core M4 Pro (8 Performance + 4 Efficiency)
GPU	20-core M4 Pro GPU
Neural Engine	16-core Apple Neural Engine — 38 TOPS (dedicated fine-tuning substrate, §8.9)
Unified RAM	64GB (recommended)
Primary SSD	256GB — OS, models, hot cache, active sessions, LoRA adapter store
Secondary HDD	5TB external — vector DB, archive, datasets, versioned adapter archive
Network	10GbE ethernet
Power Draw	30–80W under full inference load

Storage Architecture — SSD-Constrained Design

Given the 256GB SSD versus 5TB HDD constraint, HIIE implements a **streaming-first data architecture**. The core principle is:

$$\text{Data}_{\text{stored}} = \text{Data}_{\text{essential}} \quad \Leftrightarrow \quad \text{Data}_{\text{analyzed}} \gg \text{Data}_{\text{stored}}$$

Internet content, patent databases, research papers, and GitHub repositories are **analyzed entirely in RAM** and never written to disk. Only extracted structured embeddings — typically $< 0.1\%$ of the original corpus size — are persisted to ChromaDB on HDD. This means:

$$\text{Storage}_{\text{effective}} = \text{HDD}_{\text{capacity}} \times \frac{1}{\text{Compression Ratio}} \approx 5\,\text{TB} \times 1000 \approx 5\,\text{PB}_{\text{analyzed}}$$

Storage Tier Lifecycle. HIIE enforces a two-tier storage lifecycle with explicit, directional data flow:

- **SSD — Working Tier:** All data required for active inference lives here: model weights, hot vector shards for active projects, current LoRA adapters, active project outputs, and agent state checkpoints. The inference pipeline reads exclusively from SSD during active operation.
- **HDD — Archive Tier:** Completed project directories, the full ChromaDB collection suite (excluding hot shards on SSD), versioned adapter history, and curated datasets are committed to HDD at project close. HDD is not read during active inference — only at project initialization (adapter loading) and on-demand archive retrieval.

Per-Project SSD Budget. Each active project consumes approximately 30 GB of SSD working space: hot vector shards, intermediate outputs, agent state, and session logs. Given the 60 GB SSD allocation for hot cache and active outputs (§3.3), **two concurrent active projects is the practical maximum** before the SSD working budget is exhausted.

> **SSD CAPACITY MANAGEMENT**
>
> A soft warning threshold is set at **25 GB per active project**. When any project crosses this threshold, a Grafana alert fires — surfacing the condition to Alice before

> SSD pressure encroaches on the inference budget. This triggers a project hygiene review: archiving completed deliverables to HDD, purging redundant checkpoints, and compressing intermediate outputs before the working set grows further.

Memory Allocation Strategy

Resource	Allocated To	Notes
RAM 40GB	Active model inference	Quantized 14B–32B weights in unified pool
RAM 12GB	Multi-agent session state	All active project agents held in memory
RAM 8GB	Live internet streaming buffer	Analyzed in RAM, never written to disk
RAM 4GB	System + Coolify overhead	macOS + Docker + containerization
RAM <2GB	ANE fine-tuning state (background)	LoRA adapter + gradient buffer; never competes with inference pool
SSD 150GB	Model weights	Qwen2.5-32B-Q4 primary; specialist models
SSD 60GB	Hot cache + active outputs	Current project files, active vector shards
SSD <1GB	Active LoRA adapter store	All specialist adapters loaded per agent role
SSD 46GB	System + services	macOS + Docker + Coolify
HDD 2TB	ChromaDB vector database	Persistent embeddings + training data partition
HDD 2TB	Archive + curated datasets	Completed projects, training data, adapter archive
HDD 1TB	Generated files + overflow	CAD outputs, large specs, versioned adapter history

CPU / GPU Task Assignment

Component	Assigned Tasks
GPU (20-core)	LLM inference, embeddings, parallel domain processing
ANE (16-core)	LoRA fine-tuning on specialist models (background, §8.9)
CPU Performance Cores (8)	Orchestration, agent delegation, simulation runners
CPU Efficiency Cores (4)	I/O, HDD read/write, web retrieval, ethics scoring

CPU/GPU/RAM Load Model

The following mathematical model governs resource allocation under multi-agent load. Let N be the number of active agents, M_i the memory footprint of agent i, and $T_{\text{inference}}$ the per-token inference time:

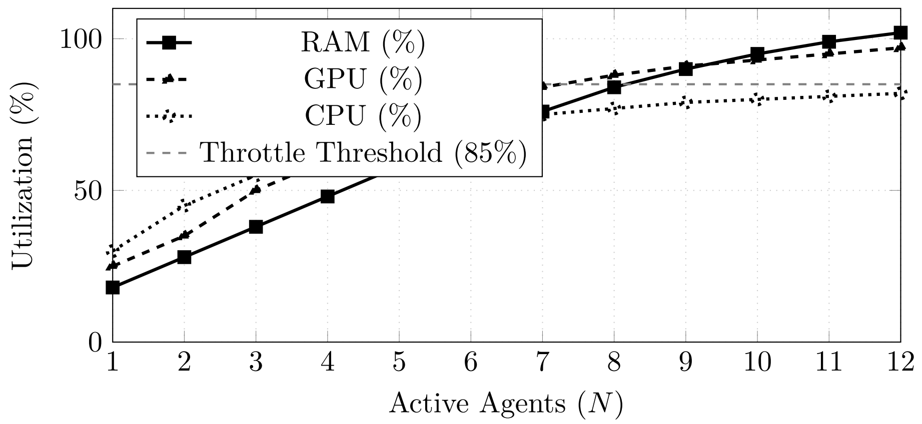
$$\text{RAM}_{\text{required}} = \sum_{i=1}^{N} M_i + M_{\text{model}} + M_{\text{buffer}}$$

$$\text{Throughput} = \frac{N \cdot \text{tokens/s}_{\text{GPU}}}{\max\left(1, \left\lceil \frac{\text{RAM}_{\text{required}}}{\text{RAM}_{\text{available}}} \right\rceil\right)}$$

When $\text{RAM}_{\text{required}} > 0.85 \times \text{RAM}_{\text{available}}$, the Celery task queue throttles agent spawning:

$$N_{\text{active}} \leq \left\lfloor \frac{0.85 \times \text{RAM}_{\text{available}} - M_{\text{model}} - M_{\text{buffer}}}{\bar{M}_{\text{agent}}} \right\rfloor$$

Resource Utilization vs. Active Agent Count



APPLE SILICON ADVANTAGE

The M4 Pro's unified memory pool is shared between CPU and GPU with no PCIe copy overhead — a significant advantage for large context windows, multi-agent state management, and continuous inference on a single-node system.

§4 Infrastructure Stack — OpenClaw + Coolify + Registered Agentics

Registered Agentics — Cloud Infrastructure Partner

Registered Agentics (`https://registeredagentics.vercel.app/`), co-authored and operated by Marko Ruble, provides the physical cloud infrastructure layer for HIIE deployment at scale. Registered Agentics functions as a Wyoming-incorporated registered agent service that additionally provides:

» **Physical Mac Mini hosting** — dedicated machines allocated per client or project tier, maintained at Registered Agentics facilities

» **Property allocation** — each Mac Mini is registered as a dedicated asset under a client's contract, with full hardware isolation

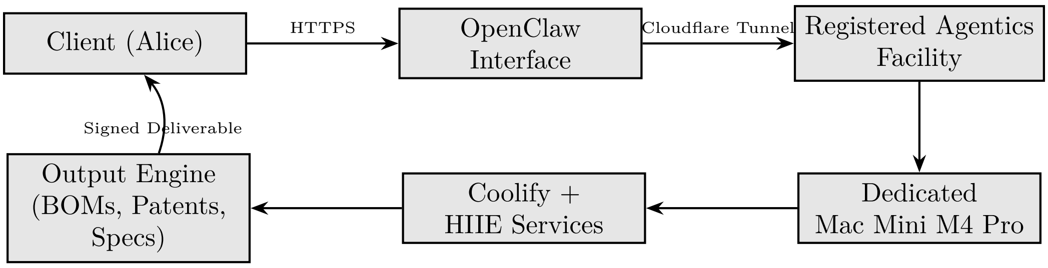
» **Logistics and transfer** — upon contract termination (standard one-year term), the Mac Mini is physically transferred and delivered to the client

» **Wyoming registered agent services** — reduces state-level administrative fees and provides legal registered presence

» **Maintenance contracts** — on-site hardware maintenance, network uptime guarantees, and physical security

REGISTERED AGENTICS DEPLOYMENT MODEL

> Each client purchasing a Builder or Enterprise HIIE subscription receives a dedicated Mac Mini M4 Pro at Registered Agentics facilities. The machine runs their isolated HIIE instance. After contract year-end, the machine ships to the client — carrying not just the hardware, but twelve months of trained specialist adapters representing accumulated project intelligence. This model eliminates multi-tenant resource contention while delivering a hardware asset that has appreciated over the contract term.



Coolify — Deployment & Orchestration

All HIIE microservices run as containerized Coolify applications locally. Key capabilities: automatic SSL via Let's Encrypt, Cloudflare Tunnel for external access without a static IP, per-service monitoring and restart policies, environment variable management, and one-click rollback for any service version.

OpenClaw — Chat Interface & Relay Layer

OpenClaw functions simultaneously as the user-facing chat application and the internal message bus through which agents communicate during active projects. It provides file upload support (spec sheets, PDFs, Gerber files, whitepapers), real-time streaming output from all active domain agents, and human sign-off gates — HIIE cannot proceed past key checkpoints without explicit OpenClaw confirmation from Alice.

Core Service Stack

Service	Technology	Purpose
Model Inference	Ollama (MLX backend)	Apple Silicon-optimized LLM serving
ANE Fine-Tuning	ANE private API (background process)	Specialist LoRA adapter training on dedicated silicon (§8.9)
Vector Database	ChromaDB	Persistent embeddings, RAG retrieval, training data partition
Orchestration API	FastAPI	Internal agent bus + external API
Task Queue	Celery + Redis	Async delegation, depth limiting
Web Retrieval	Playwright + Trafilatura	Live analysis — RAM only, never stored
Electrical Simulation	PySpice	Circuit validation
Mechanical Simulation	FreeCAD Python API	Structural and thermal analysis
Monitoring	Grafana + Prometheus	System health, inference dashboards, adapter performance

Container Management	Coolify + Docker	Service orchestration and lifecycle

§5 Anti-Slop Validation Framework — Preventing AI Output Degradation

"AI slop" refers to outputs that are syntactically plausible but semantically hollow, physically infeasible, or epistemically unchecked — outputs that *look* like answers but *are not* engineering solutions. HIIE treats slop prevention as an architectural first-class concern, not a post-hoc filter.

Three Pillars of Slop Prevention

Pillar 1: Recursive Validation with Divergence Enforcement

Each domain output undergoes iterative self-critique and cross-agent challenge. Let O_k be the output at iteration k and $\mathrm{sim}(\cdot,\cdot)$ be cosine similarity in the output embedding space. The recursion continues until:

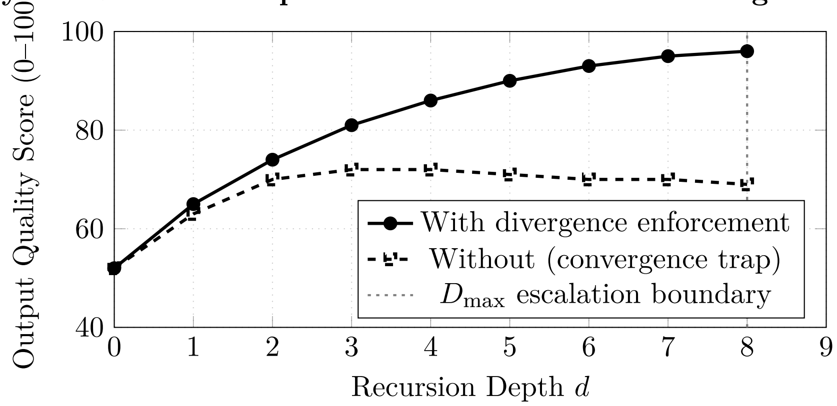
$$\mathrm{sim}(O_k, O_{k-1}) < \tau_{\text{converge}} = 0.92$$

If $\mathrm{sim}(O_k, O_{k-1}) \geq 0.92$, the system forces a **divergence prompt** — injecting a contrastive perspective that challenges the current reasoning path. This prevents circular loops where agents reinforce each other's errors.

Stack overflow protection is enforced via a hard recursion depth limit $D_{\max} = 8$, after which the current best output is escalated to Alice with a flag rather than continuing to recurse:

$$\text{If } d > D_{\max}: \quad \text{ESCALATE}(O_{D_{\max}}) \rightarrow \text{Alice (human review)}$$

Quality vs. Recursion Depth — With and Without Divergence Enforcement

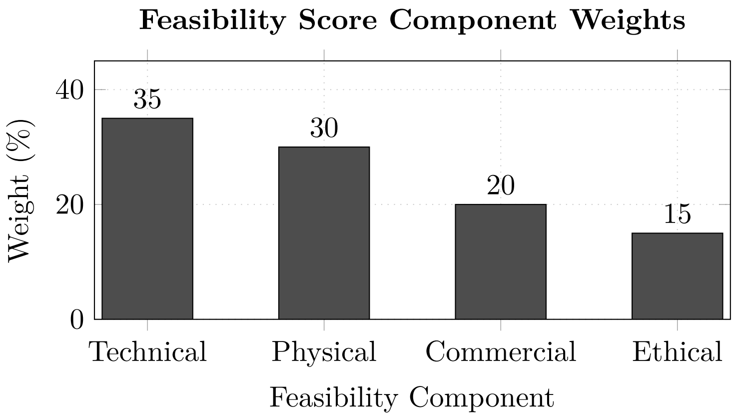


Pillar 2: Human Perspective Injection

At defined checkpoints, HIIE generates a **layperson summary** and a **domain-expert critique** of every technical output. These are required inputs to the Feasibility Manager's scoring function:

$$F_{\text{score}} = w_1 \cdot S_{\text{technical}} + w_2 \cdot S_{\text{physical}} + w_3 \cdot S_{\text{commercial}} + w_4 \cdot S_{\text{ethical}}$$

where $w_1 = 0.35,\ w_2 = 0.30,\ w_3 = 0.20,\ w_4 = 0.15$ and all $S_i \in [0, 100]$. Outputs with $F_{\text{score}} < 65$ are rejected and returned to the agent team for rework. Outputs with $F_{\text{score}} \in [65, 80)$ are flagged for Alice's attention. Outputs with $F_{\text{score}} \geq 80$ proceed to the Ethics Board.



Pillar 3: Physical Ground-Truth Validation

Every design claim must be anchored to a verifiable physical reference:

- Material properties sourced from NIST, Matweb, or ASM International — not generated from model weights
- Electrical designs validated against PySpice simulation before proceeding
- Mechanical designs stress-tested in FreeCAD Python API before BOM generation
- Chemistry outputs cross-referenced against PubChem and ChemRxiv before synthesis feasibility is asserted

> **ANTI-SLOP GUARANTEE**
>
> HIIE will not produce a Bill of Materials that references components that do not exist, a schematic that violates Kirchhoff's laws, a material specification that contradicts published NIST data, or a patent claim that describes physics that cannot occur. If simulation or database validation fails, the output is blocked — not papered over.

Preparation Steps for Anti-Slop Implementation

Step 1. Embed ground-truth retrieval at every domain module entry point. Before any generative step, the relevant module retrieves authoritative source data (NIST, PubChem, USPTO, IEEE). The model reasons *from* data, not *toward* plausible-sounding data.

Step 2. Implement the cosine similarity divergence check ($\tau = 0.92$) in the Celery task wrapper. Every agent output is embedded and compared to its predecessor before the next iteration is spawned.

Step 3. Deploy the Feasibility Manager as a separate model instance. It must not share context with the generating agent — independence is required for genuine critique.

Step 4. Configure $D_{\max} = 8$ and automatic escalation in the FastAPI orchestration layer. Hard-code this limit; do not make it a configurable parameter that can be overridden by a project prompt.

Step 5. Require layperson and expert summaries at every approval gate. Alice reads both before signing off on requirements, interim designs, and final outputs.

Step 6. Log all validation failures with full reasoning traces to the immutable audit log. Failure patterns become training signals for improving domain module prompts.

§6 Intelligence Layers & Domain Modules

Module	Coverage	Key Capabilities
Hardware Engineering	PCB, chip architecture, LPU/AI processor, manufacturing machines	Schematic generation, BOM, Gerber files, VHDL/Verilog for custom silicon
Materials Science	Properties, US sourcing, cost, environmental lifecycle	Substitution recommendations, supplier integration, environmental scoring
Manufacturing Systems	Factory line design, batch processes, QC, automation	Process flows, equipment spec, yield modeling
Software Engineering	Full-stack, embedded, firmware, AI	Production-grade code with enforced design principles
Chemistry & Materials	Compounds, synthesis, novel material discovery	PubChem/ChemRxiv retrieval, synthesis feasibility, safety analysis
Patent Intelligence	Global patents — USPTO, EPO, WIPO, JPO	Prior art search, novelty windows, provisional claim drafting
Whitepaper & Standards	IEEE, IPC, JEDEC, AI research	Standards compliance, document synthesis, gap analysis
Environmental Systems	Carbon accounting, renewable integration	Impact scoring, net-positive pathway recommendations

Recursive Reasoning — Overflow Prevention

HIIE uses iterative deepening with hard depth limits and SSD checkpointing rather than pure recursion. A similarity detection threshold ($\tau = 0.92$) forces divergent reasoning when outputs converge. RAM pressure monitoring triggers state serialization to SSD above 85% utilization.

§7 Agent Delegation Framework

Each project spawns a full agent team. Alice and Arthur are the two permanent principals across all projects.

Role	Type	Responsibilities
Alice	Human	Primary director. Sets problem statement, approves milestones, holds final sign-off authority. Cannot be overridden.
Arthur	AI	Persistent AI co-director. Maintains long-term project memory, flags output drift, advocates for Alice's original intent.
Theoretical Engineer	Agent	First-principles reasoning and conceptual design. Explores solution space without cost bias. Flags physical limits.

Materials Engineer	Agent	Selects and optimizes all physical materials. Prioritizes US-accessible sourcing, cost, and environmental lifecycle.
Civil & Structural Engineer	Agent	Evaluates spatial requirements, structural integrity, facility needs, and physical installation feasibility.
Mfg. Process Engineer	Agent	Designs production processes, factory line design, batch optimization, yield modeling, QC system design.
Research Group (x3)	Agent	Three concurrent agents: patents, academic papers, and standards/GitHub — no redundancy between agents.
Feasibility Manager	Agent	Scores outputs against technical, manufacturing, commercial, and timeline feasibility. Blocks below-threshold outputs.
Ethics Officer	Agent	Reviews for dual-use risk, societal impact, environmental consequences. Holds veto power. All reports are immutable.
Project Manager	Agent	Coordinates all agents, tracks milestones, identifies conflicts, delivers structured status reports to Alice and Arthur.
Patent Strategist	Agent	Monitors global patent landscape in real-time, identifies novelty windows, drafts provisional claims, flags FTO risks.
Documentation Architect	Agent	Synthesizes all outputs into complete deliverable package: specs, patents, whitepapers, material lists, process flows.

§8 Adaptive Resource Economy & Agent Incentive Framework

HIIE operates as more than a technical pipeline — it is a self-governing economic system. Every participant, human or agent, operates within a measurable framework of resource allocation, performance accountability, and convergent incentives. The goal is not punishment for failure but continuous refinement toward precision: a closed loop in which every decision, at every level, improves the quality of subsequent decisions.

HIIE Compute Token (HCT) — The Internal Unit of Account

The **HIIE Compute Token (HCT)** is the internal resource accounting unit governing all agent activity. It is not a currency, carries no external value, and has no relationship to client billing or any outside economy. It exists solely to make resource consumption visible, attributable, and optimizable within the system.

One HCT represents a normalized bundle of compute resources:

$$1\,\mathrm{HCT} = \alpha \cdot t_{\mathrm{GPU}}\ [\mathrm{ms}] + \beta \cdot t_{\mathrm{CPU}}\ [\mathrm{ms}] + \gamma \cdot m_{\mathrm{RAM}}\ [\mathrm{MB} \cdot \mathrm{s}]$$

where α, β, and γ are weighting coefficients reflecting the relative scarcity of each resource on the Mac Mini M4 Pro node ($\alpha > \beta > \gamma$, GPU time being the most constrained). The Treasury Agent recalibrates these coefficients at the start of each project based on current system load.

Dynamic Resource Pricing — Self-Determining Allocation

Rather than a static throttle ceiling, resource costs fluctuate continuously as a function of real-time utilization $U(t) \in [0, 1]$:

$$P(t) = P_{\text{base}} \cdot \left(1 + \lambda \cdot \frac{U(t)}{1 - U(t) + \epsilon}\right)$$

where P_{base} is the nominal HCT cost per resource unit, λ is the price sensitivity parameter, and ϵ prevents division by zero near full saturation. This produces a self-correcting market signal:

- **Underserved environment** ($U(t) < 0.50$) — HCT cost falls below P_{base}, incentivizing the scheduler to allocate more work to available capacity
- **Nominal range** ($0.50 \leq U(t) < 0.85$) — HCT cost near baseline; agents operate at standard allocation
- **Overserved environment** ($U(t) \geq 0.85$) — HCT cost spikes sharply; agents are naturally incentivized to compress context, yield memory, or queue rather than compete for resources

This replaces the existing hard throttle with a continuous price gradient, preserving the 85% ceiling as an economic signal rather than an abrupt cutoff.

Agent Payroll — Base HCT Allocation per Project Cycle

Each agent role receives a guaranteed base HCT allocation at project initialization — the **payroll**. This is the minimum resource floor the agent is entitled to regardless of project complexity. The Treasury Agent distributes these allocations before any task execution begins.

Agent Role	Base HCT	Rationale
Theoretical Engineer	120	First-principles reasoning; high GPU demand
Materials Engineer	80	Database retrieval + property matching
Civil & Structural	70	FreeCAD simulation workload
Mfg. Process Engineer	90	Factory line modeling; yield computation
Research Group (each)	100	Live retrieval; three concurrent instances
Feasibility Manager	60	Scoring runs; independent model instance
Ethics Officer	50	Structured review; immutable log writes
Project Manager	40	Coordination; low direct inference load
Patent Strategist	80	Real-time global patent sweep
Documentation Architect	60	Synthesis and LaTeX compilation
Treasury Agent	30	Continuous monitoring; lightweight process

Performance Bonuses — HCT as Quality Signal

Agents that exceed baseline performance earn **bonus HCT** redeemable within the same project cycle to request additional compute, spawn sub-agents, or extend live retrieval sessions. Bonus HCT cannot be accumulated across projects — it expires at project close.

The bonus for agent i is computed at each project milestone:

$$B_i = B_{\text{base}} \cdot \underbrace{\frac{F_i}{F_{\text{target}}}}_{\text{quality ratio}} \cdot \underbrace{\frac{D_{\text{target}}}{d_i}}_{\text{recursion efficiency}} \cdot \underbrace{A_i}_{\text{Alice acceptance}}$$

where F_i is the agent's Feasibility Score contribution, $F_{\text{target}} = 80$, d_i is the recursion depth at which the agent resolved its task, $D_{\text{target}} = 4$ (the efficient midpoint of the $D_{\text{max}} = 8$ limit), and $A_i \in \{0.5, 1.0, 1.5\}$ reflects Alice's final gate decision (rework required / accepted / accepted with commendation).

Outputs blocked by the Feasibility Manager return zero bonus for that cycle. Outputs rejected by the Ethics Officer trigger a **bonus claw-back**: previously awarded HCT for that task is reclaimed by the Treasury Agent.

> **BONUS PHILOSOPHY**
>
> Bonus HCT is a signal, not a reward in the human sense. An agent earning surplus compute tokens is the system recognizing that this agent's next step is worth more resource investment. An agent losing tokens is the system redirecting capacity toward higher-yield work. Neither state is permanent — both reset at project close.

Treasury Agent — Self-Auditing & Checks and Balances

The **Treasury Agent** is a lightweight, always-on process running independently of the domain agent team. It holds no domain knowledge and participates in no engineering tasks. Its sole function is economic governance:

- **Project-level HCT cap enforcement** — total spend across all agents cannot exceed the project budget ceiling, set at initialization based on project complexity tier
- **Spending velocity monitoring** — flags any agent whose HCT consumption rate exceeds 2× its base allocation per time window without a corresponding milestone output
- **Claw-back execution** — reclaims bonus HCT from agents whose outputs are subsequently rejected downstream
- **Gate suspension accounting** — when Alice has not approved a gate, the Treasury Agent suspends all blocked agents, releases their RAM allocation back to the pool, and logs the idle period. Upon Alice's approval, agents are reinstated with their HCT entitlements preserved but their held RAM re-priced at the current dynamic rate
- **Immutable spending ledger** — a tamper-evident log of all HCT allocations, spends, bonuses, and claw-backs per project, parallel to the ethics audit log and exportable alongside it

The Treasury Agent's own audit logic is protected from modification by any agent instruction, mirroring the immutability of the ethics framework. No agent, including Arthur, can alter the Treasury Agent's accounting rules.

Alice Accountability Protocol — Convergence Without Penalty

Alice holds final authority over all HIIE outputs and bears no resource loss, HCT deduction, or score penalty under any circumstance. Her accountability mechanism is entirely **Socratic and convergent**: the system will not advance past a stage until Alice's input meets the minimum quality threshold for that stage. The mechanism is designed to eliminate the worst outcome in any AI-assisted system — a vague input accepted without challenge that generates a plausible-looking but fundamentally misaligned output, trapping the entire pipeline in a slop cycle from which recovery requires a full restart.

Project Commitment Declaration

Before any agent is spawned or any HCT is allocated, Alice must pass an **intake quality gate**. The system evaluates her problem statement across three dimensions:

$$Q_{\text{Alice}} = w_s \cdot S_{\text{specificity}} + w_c \cdot S_{\text{consistency}} + w_b \cdot S_{\text{boundedness}}$$

where $w_s = 0.40$, $w_c = 0.35$, $w_b = 0.25$ and all scores $\in [0, 100]$. If $Q_{\text{Alice}} < 70$, the system returns a single targeted Socratic question — not a list of problems, not a form to fill out, but one precise question aimed at the largest ambiguity. This repeats until the threshold is met. **No compute is allocated until $Q_{\text{Alice}} \geq 70$.**

This is not a friction mechanism. It is waste prevention at source — the most efficient point in the entire pipeline to correct a misalignment is before any work begins. Every Socratic exchange brings Alice closer to the precision required for a high-quality outcome. The loop never punishes; it only refines.

Exploratory vs. Committed Mode

Alice explicitly declares her intent before the intake gate opens:

- **Exploratory Mode** — limited agent instantiation, reduced HCT allocation, no deliberation scoring, no intake quality gate. Alice is thinking out loud. The system assists without full commitment. Exploratory sessions do not feed into Alice's system-level contribution history.
- **Committed Mode** — full agent team, full HCT allocation, intake quality gate active, deliberation scoring active. Alice is initiating a real project. All gates and accountability mechanisms are live.

The system never silently treats an exploratory session as a committed project. Resource allocation scales to declared intent.

Deliberation Scoring at Approval Gates

At each approval gate, the system records a **deliberation signal**: the time elapsed between gate presentation and Alice's decision, and whether she engaged with both the layperson summary and the expert critique before approving. Rapid sequential approvals — gates cleared without meaningful engagement — are flagged in the immutable audit log and surfaced to Arthur.

Arthur is then **required** to present a structured alternative perspective before the next gate opens. Alice can still proceed — her authority is absolute — but she does so on the record, with Arthur's analysis attached to the project audit trail. The deliberation signal does not block Alice. It ensures that when she chooses to move fast, the system has done its duty to surface what she may have passed over.

Arthur as Bilateral Auditor

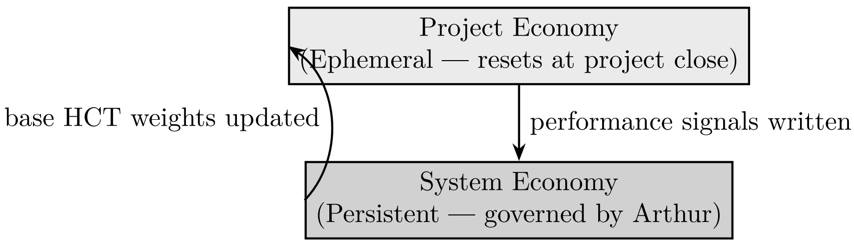
Arthur's existing role — persistent AI co-director, long-term memory keeper, advocate for Alice's original intent — is extended bilaterally. He audits both directions of the pipeline:

- **Downward** — agents drifting from Alice's stated requirements, output quality regression between milestones, domain modules producing internally inconsistent results (already defined in Arthur's role)
- **Upward** — Alice's gate decisions contradicting her own requirements document, approval patterns that correlate with downstream rework, input patterns that historically produce low-F_{score} outcomes

When Arthur identifies an upward inconsistency, he surfaces it as a **structured alternative**: a concise statement of what Alice originally declared, what she is approving now, and what the predicted HCT cost of the divergence is. Arthur never blocks Alice. He ensures that every consequential decision is made with full awareness of its implications.

Cross-Project Durability — The Two-Layer Economy

The HCT economy operates across two distinct temporal scopes:



Project Economy (Ephemeral): HCT budgets, bonus pools, spending ledgers, and per-project scores reset completely at project close. No agent carries debt or surplus into a subsequent project. Each project begins on an equal footing.

System Economy (Persistent): Agent specialization weights, base HCT allocation adjustments, Arthur's memory of Alice's input patterns, and system-level KPI trends persist across the lifetime of the system. A Hardware Engineering agent that has completed forty PCB projects reaches the same F_{score} threshold with fewer HCT than one on its second project. That efficiency delta is a system-level asset, tracked by Arthur and reflected in future base payroll assignments.

Reputation Decay Function

To prevent early-project performance from permanently defining an agent's allocation — and to prevent one poor session from permanently coloring Alice's Socratic treatment — all system-economy scores apply an exponential decay:

$$R_i(t) = R_i(t_0) \cdot e^{-\lambda_d(t-t_0)} + \Delta R_i$$

where λ_d is the decay rate (default: half-life of 90 days), t_0 is the timestamp of the last project, and ΔR_i is the performance delta from the most recent project. Recent performance always outweighs historical performance. The system learns from history without being imprisoned by it.

System-Level KPI Dashboard

Arthur maintains and surfaces a set of macro-level KPIs across all projects that reflect the health and trajectory of HIIE as a whole:

KPI	Target	Description
Mean F_{score} (rolling 10 projects)	≥ 82	System-wide output quality trend
HCT Efficiency Ratio	≥ 1.20	Output quality per HCT spent, normalized
Mean recursion depth at resolution	≤ 5	Measures reasoning efficiency over time
Alice intake gate pass rate (first attempt)	$\geq 70\%$	Reflects input quality improvement over time
Agent claw-back rate	$\leq 5\%$	Proportion of bonuses reclaimed per project
Gate deliberation flag rate	$\leq 10\%$	Proportion of gates flagged for low engagement
Adapter F_{score} delta (rolling 5 projects)	$\geq +2$ pts	Improvement attributable to ANE fine-tuning (§8.9)

Sustained degradation in any KPI triggers Arthur to surface a structured system health report to Alice, recommending specific adjustments to agent payroll weights, intake quality thresholds, or project scoping practices. Arthur proposes; Alice decides.

Reserve Pool — Project Insurance

Each project initializes a **reserve pool** of HCT set at 15% of the total project budget. The reserve is held by the Treasury Agent and is not accessible to individual agents during normal operation. It is deployed only under two conditions:

- **Cascade failure absorption** — if a domain module produces a catastrophically rejected output requiring full rework, the reserve covers the rework cycle without stripping HCT from other agents mid-project
- **Ethics Officer override costs** — if the Ethics Officer vetoes an output and triggers a mandatory redesign, the redesign cycle is funded from the reserve, not from agent bonuses

Unused reserve HCT at project close is not distributed as bonus. It returns to the system pool, recorded in the spending ledger as *reserve returned*, which is itself a positive system-level signal: a project that did not need its reserve is a well-scoped, well-executed project.

CLOSED-LOOP ACCOUNTABILITY

Every participant in HIIE — agents, Arthur, and Alice — operates within a measurable, transparent accountability framework. No participant is exempt. No failure is hidden. No success goes unrecorded. The economy does not punish — it converges. Over time, every component of the system becomes more precise, more efficient, and more aligned with the intent that initiated it.

§8.9 Persistent Intellectual Capital — ANE Background Fine-Tuning & LoRA Branching

The HCT economy governs how HIIE allocates compute during active project execution. Section 8.9 extends that economy into a dimension that operates beneath active inference: the continuous, silent improvement of specialist model weights using dedicated silicon that would otherwise sit idle.

This is HIIE's **persistent intellectual capital layer**. Every project produces labeled training signal. Every idle GPU cycle is an opportunity. Every completed year of operation leaves the Mac Mini more capable than when it started — not because its hardware changed, but because its specialist models have been fine-tuned on real, validated, Arthur Labs project outcomes.

8.9.1 The Idle Silicon Problem

During active inference, HIIE's GPU (20-core M4 Pro) is saturated running Qwen2.5-32B for orchestration, domain reasoning, and embeddings. The Apple Neural Engine (16-core ANE) is architecturally separate silicon — it receives no work from Metal or Ollama during standard inference. It is idle.

This represents a continuous waste of dedicated AI hardware. The ANE is rated at 38 TOPS on the M4 Pro. During a typical HIIE project session, that capacity goes entirely unused. Section 8.9 closes that gap by assigning the ANE a permanent background role: specialist model fine-tuning via LoRA adaptation passes, running in parallel with active GPU inference without resource contention.

Resource	During Active Project	Background ANE Training
GPU (20-core)	LLM inference, embeddings, domain reasoning	Fully allocated — no contention
ANE (16-core)	Idle — unused by Metal/Ollama	LoRA fine-tuning on specialist models
CPU Perf Cores (8)	Orchestration, agent delegation	Gradient accumulation (cblas)
CPU Eff Cores (4)	I/O, web retrieval, HDD writes	Checkpoint writes to HDD
RAM (background)	<2GB — LoRA adapter + training state	Isolated allocation, never competes with 40GB inference pool

8.9.2 What Is Being Trained — LoRA Adapters, Not the Base Model

HIIE's primary inference model (Qwen2.5-32B-Q4) is never modified. It remains frozen on SSD, serving as the stable reasoning foundation for all agents. What the ANE trains are **Low-Rank Adaptation (LoRA) adapters** — lightweight parameter overlays that modify how the base model behaves for a specific domain task without altering its weights.

LoRA adapters are architecturally small: typically 10–50MB per adapter versus 14GB+ for the base model. They are loaded at agent instantiation and unloaded when the agent completes. Multiple adapters can coexist on SSD simultaneously, each representing a different specialist agent's accumulated learning.

LoRA Branching Architecture

```
Base Qwen2.5-7B Specialist (frozen on SSD, ~4GB)
  ⊢ Adapter A: Patent Strategist   (~30MB)
  ⊢ Adapter B: Chemistry Agent   (~25MB)
  ⊢ Adapter C: Feasibility Manager (~20MB)
  ⊢ Adapter D: Hardware Engineer  (~35MB)
  ⊢ Adapter E: Materials Engineer  (~28MB)
```

Each adapter represents a branch off the frozen base model trunk. The trunk never changes. The branches grow continuously as HIIE completes projects. This is the hardware expression of the two-layer economy described in §8.8: the System Economy's persistent learning is literally encoded into separate silicon and stored as versioned adapter files on HDD.

8.9.3 Single-Layer Fine-Tuning — Architectural Placement

Current ANE training capability supports single-layer transformer fine-tuning: one complete attention + FFN block per training pass. The question of which layer to target is not arbitrary.

Transformer layers are not equivalent in function. Early layers (0–8 in a 32-layer model) encode syntax and surface patterns — foundational knowledge that does not benefit from domain-specific adaptation. Mid-to-late layers (layers 16–28) encode domain reasoning, concept formation, and task-specific behavior. These are the highest-value targets for specialist fine-tuning.

Layer Range	Encodes	Fine-Tuning Value for HIIE
Layers 0–8 (Early)	Syntax, basic language patterns	Low — foundational, domain-agnostic
Layers 9–18 (Mid)	Domain concept formation, reasoning chains	High — where domain specialization begins
Layers 19–26 (Late)	Task-specific behavior, output formatting	Highest — direct impact on agent output quality
Layers 27–32 (Final)	Token prediction, surface generation	Medium — output style, less domain-critical

HIIE targets late-layer fine-tuning for all specialist adapters. A Patent Strategist adapter trained on layers 19–26 improves how the agent reasons about novelty windows and claim structure — without touching the foundational language understanding that makes it useful in the first place.

8.9.4 The Training Data Pipeline — Project Outcomes as Labels

HIIE does not require external datasets to fine-tune its specialist models. Every completed project generates labeled training signal automatically through the existing validation and approval architecture:

- Feasibility Manager outputs carry F_{score} labels — scored pairs of (agent output, feasibility score) are ground-truth training examples for the Feasibility Manager adapter
- Alice acceptance decisions at approval gates provide binary and graded labels (rework required / accepted / accepted with commendation) for all domain agent adapters
- Ethics Officer assessments label dual-use risk, environmental impact, and structural compliance — training signal for the Ethics scoring model
- Anti-slop divergence flags identify outputs that required forced re-reasoning — negative training examples that teach agents to avoid convergence traps
- Simulation validation results (PySpice pass/fail, FreeCAD stress test outcomes) anchor training signal in physical ground-truth — not model preference

> **Closed Training Loop**
>
> Active Project (GPU) → Produces outputs → Validation layer scores them → Alice approves/rejects at gates → All labeled pairs written to ChromaDB training partition on HDD → ANE background process reads partition → LoRA adapter updated → Improved specialist model loaded at next project initialization.

This loop requires no human curation, no external dataset acquisition, and no additional infrastructure. The training data is a natural byproduct of HIIE's existing validation architecture. The ANE fine-tuning layer simply harvests what the pipeline already produces.

8.9.5 RAG and Fine-Tuning as Complementary Memory Systems

RAG (ChromaDB) and ANE fine-tuning serve distinct and non-competing memory functions. They operate at different layers of the intelligence stack and must be understood as complementary, not redundant:

Dimension	RAG (ChromaDB on HDD)	LoRA Fine-Tuning (ANE)
Memory type	Explicit — retrieved at inference time	Implicit — encoded in weights
Question answered	What do I know?	How do I reason?
Update speed	Immediate — new embeddings written per project	Gradual — adapter updated across project batches
Scope	Project-level and corpus-level facts	System-level reasoning capability
Persistence	HDD — queryable across all projects	HDD — loaded per agent role
Interaction	Feeds context into fine-tuned model	Improves use of retrieved context

The interaction between the two systems is multiplicative, not additive. A fine-tuned Patent

Strategist adapter improves the agent's ability to reason about the patent documents retrieved by RAG. A well-populated ChromaDB provides richer retrieval context that the fine-tuned model is better equipped to synthesize. Each layer makes the other more effective.

8.9.6 Adapter Versioning, Storage, and the Machine as Asset

All LoRA adapters are versioned and stored on the 5TB HDD alongside ChromaDB. Storage overhead is minimal — a full suite of specialist adapters across all HIIE agent roles consumes less than 500MB total, a negligible fraction of available HDD capacity.

Adapter versioning follows the project lifecycle:

- Each adapter version is tagged with the project batch that produced it and the F_{score} delta observed across that batch
- Previous adapter versions are archived, not deleted — rollback is available if a fine-tuning batch degrades performance
- The Treasury Agent monitors HCT efficiency ratios across adapter versions — a version that produces lower F_{score} outputs at equal HCT cost triggers an automatic rollback flag to Alice
- Arthur tracks adapter performance trends as part of the System-Level KPI Dashboard (§8.8.2), surfacing degradation before it affects client-facing output quality

This versioning architecture produces a significant commercial property: the Mac Mini delivered to a Builder or Enterprise client at contract year-end is not the same machine that was initialized twelve months prior. It carries trained specialist adapters derived from real project outcomes — accumulated intellectual capital encoded in weights, archived in ChromaDB, and versioned in the adapter store. **The hardware asset appreciates over the contract term.**

8.9.7 Maturity Trajectory — Phase Alignment

ANE background fine-tuning is introduced as a capability that matures across HIIE's development phases rather than a Day 1 requirement:

Phase	ANE Training Status	Capability Unlocked
Phase 1 (Months 1–6)	Architecture designed; training data pipeline built into ChromaDB partition	Training data accumulates; no adapter deployment yet
Phase 2 (Months 6–12)	Single-layer LoRA fine-tuning on Feasibility Manager and Patent Strategist adapters	First measurable F_{score} improvement from adapter deployment
Phase 3 (Months 12–24)	Full specialist adapter suite across all domain agent roles	System-wide capability improvement; adapter versioning live
Phase 4 (Months 24–48)	Multi-layer fine-tuning as ANE research matures; potential custom adapter architectures	HIIE designs its own fine-tuning pipeline on purpose-built hardware

PERSISTENT INTELLECTUAL CAPITAL

HIIE does not merely execute projects — it compounds from them. Every validated output, every Alice approval, every simulation result is simultaneously a deliverable and a training signal. The ANE background fine-tuning layer converts that signal into improved specialist model weights, silently, on dedicated silicon, without interrupting active inference. Over the lifetime of the system, this produces an intelligence substrate that is measurably, demonstrably better than it was at initialization — not by design change, but by accumulated experience.

§9 Human-AI Dialogue & User Flow

Step 1. Problem Statement Input — Alice submits goal in any form through OpenClaw: sentence, paragraph, spec sheet upload, reference images, prior art documents.

Step 2. Socratic Clarification Loop — HIIE asks one targeted question at a time. Arthur flags contradictions. No assumptions made without confirmation.

Step 3. Requirements Document — Human Approval Gate — Structured requirements presented to Alice. No project work begins without explicit sign-off. This document governs all subsequent agent decisions.

Step 4. Agent Team Spawn & Research Phase — Full agent team initialized. Research Group begins live retrieval. Theoretical Engineer generates first-principles concepts. Patent Strategist scans global landscape.

Step 5. Parallel Domain Generation — Domain modules execute via async task queues. Feasibility Manager cross-validates. Alice receives streaming progress and can redirect any agent in real-time.

Step 6. Anti-Slop Validation Loop — Cosine similarity divergence checks, Feasibility scoring ($F_{\text{score}} \geq 65$ required), and physical ground-truth anchoring execute before simulation.

Step 7. Simulation & Validation Loop — Electrical, mechanical, and thermal simulations run against designs. Results feed back to agents for refinement. Loop continues until thresholds are met or escalated.

Step 8. Ethics Board Review — Human Approval Gate — Full output reviewed by Ethics Officer. Any flag requires Alice's explicit acknowledgment before final output proceeds.

Step 9. Final Output Package & Archive — Documentation Architect compiles complete deliverable set. Alice performs final review. Upon sign-off, outputs archived to HDD, patent strategy initiated, and project outcomes written to ANE training partition (§8.9).

§10 Use Case Catalogue — Reference Projects

The following use cases illustrate HIIE's intended scope and demonstrate the system's reasoning range from precision instrumentation to large-scale manufacturing. Each represents a full HIIE project cycle from intent to deliverable.

Use Case 1 — Localized Atomic Force Microscope ($\leq 3 \times 3$ ft)

Alice's Intent: "Architect me a localized atomic microscope that is under 3x3 feet and has a remote connection point to view the microscope and control it."

HIIE Approach: Hardware Engineering generates a cantilever-based AFM design with piezoelectric XYZ stage. The constraint $V \leq (0.9\,\text{m})^3$ governs all mechanical envelope decisions. Materials Engineer specifies silicon nitride cantilevers sourced from US suppliers. A Raspberry Pi 5 + custom FPGA control board provides the remote WebRTC connection layer. Civil & Structural Engineer validates vibration isolation table requirements.

Key Design Equation — Cantilever Spring Constant:

$$k = \frac{Ewt^3}{4L^3}$$

where E is Young's modulus, w is cantilever width, t is thickness, and L is cantilever length. Target: $k \approx 0.1$–$1\,\text{N/m}$ for contact mode. Force sensitivity $F_{\min} = k \cdot \delta z_{\min}$ where $\delta z_{\min} \approx 0.1\,\text{nm}$ for the piezo stage.

Expected Outcomes: Full mechanical CAD (FreeCAD), PCB schematic (KiCad), BOM with US sourcing, remote control software package, provisional patent draft, viability report. Estimated BOM cost: $8,000–$25,000 depending on detector quality.

Use Case 2 — Novel vRAM Architecture

Alice's Intent: "Architect a new design system for vRAM that is material cost effective and easy to access the materials in the United States."

HIIE Approach: Hardware Engineering and Materials Science modules run in parallel. The Patent Strategist performs real-time prior art sweep across HBM, GDDR, and LPDDR patent families. The system targets SRAM-on-DRAM stacking alternatives using domestically available silicon and substrate materials.

Key Analysis — Memory Bandwidth:

$$B = 2 \cdot W \cdot f \cdot N_{\text{channels}}$$

HIIE's architecture exploration targets $B > 1\,\text{TB/s}$ at $< 30\%$ of current HBM2e material cost. Bandwidth efficiency $\eta = B/P_{\text{watts}}$ is optimized against domestic supply chain constraints.

Expected Outcomes: Architecture whitepaper, provisional patent claims, BOM with US supplier pricing, VHDL/Verilog controller specification. Yellow paper for formal protocol specification if a novel memory bus protocol emerges.

Use Case 3 — GPU Batch Manufacturing Asset System

Alice's Intent: "Create a manufacturing asset system that batch produces a specific kind of product — like a GPU."

HIIE Approach: Manufacturing Systems module designs the full factory line: wafer handling, die bonding, substrate attachment, thermal interface application, and burn-in testing stations.

Key Metric — Yield Model (Poisson):

$$Y = e^{-A \cdot D_0}$$

where A is die area (mm^2) and D_0 is defect density ($defects/cm^2$). The system optimizes batch size B^* to maximize throughput-per-dollar:

$$B^* = \arg\max_B \frac{B \cdot Y(A, D_0)}{C_{setup} + B \cdot C_{unit}}$$

Expected Outcomes: Process flow diagram (SVG), equipment specification list, facility layout CAD, yield model, QC system design, environmental report.

Use Case 4 — Environmentally Net-Positive AI Compute Machine

Alice's Intent: "Generate an environmentally net positive AI machine that produces a positive carbon offset."

HIIE Approach: Environmental Systems module leads. The design targets a carbon offset $\Delta C > 0$:

$$\Delta C = C_{offset} - C_{operational} - C_{embodied} > 0$$

Options explored: waste heat capture for building heating, on-site solar generation, direct air capture integration, and renewable energy certificate procurement. The carbon payback period t^* is computed as:

$$t^* = \frac{C_{embodied}}{C_{offset} - C_{operational}} \quad \text{(years)}$$

Expected Outcomes: Carbon accounting report, hardware specification, renewable integration plan, net-positive certification pathway.

Use Case 5 — Drone Fleet Factory with Headwear Control Interface

---

Alice's Intent: "Make a factory line that produces fleets of drones that use headwear for controlling the drones."

HIIE Approach: Manufacturing Systems designs drone assembly line. Software Engineering generates the EEG/EMG headwear signal processing pipeline. The control latency constraint drives the firmware architecture:

$$\tau_{total} = \tau_{EEG} + \tau_{proc} + \tau_{radio} + \tau_{actuator} < 50\,\text{ms}$$

(50 ms is the human proprioceptive threshold for natural control feel.) Typical budget: $\tau_{EEG} \approx 10\,\text{ms}$, $\tau_{proc} \approx 15\,\text{ms}$, $\tau_{radio} \approx 5\,\text{ms}$, $\tau_{actuator} \approx 10\,\text{ms}$.

Expected Outcomes: Factory line design, drone PCB schematic, headwear firmware, BOM, patent novelty analysis, viability report.

Use Case 6 — Chemical Compound Analysis Agent for Novel Materials

---

Alice's Intent: "Find bounties and train a local agent AI model that all it does is analyze chemical compounds for creating new materials."

HIIE Approach: Chemistry & Materials module deploys a fine-tuned specialist model trained on PubChem, ChemRxiv, and materials science literature. The ANE background fine-tuning layer (§8.9) continuously improves this specialist model using compound analysis outcomes as training signal. The agent participates in open science bounty programs (Materials Project, NIST challenge programs). Outputs feed back into HIIE's ChromaDB as a curated materials corpus.

Key Metric — Tanimoto Novelty Score:

$$S_{\text{novel}} = 1 - \max_{j \in \text{DB}} T(C_{\text{candidate}}, C_j) = 1 - \max_j \frac{|A_j \cap A_{\text{cand}}|}{|A_j \cup A_{\text{cand}}|}$$

where A_j is the fingerprint bit-set of compound j. Compounds with $S_{\text{novel}} > 0.85$ trigger provisional patent drafting.

Expected Outcomes: Fine-tuned specialist model, materials database, novel compound candidates, provisional patent drafts, bounty participation record.

Use Case 7 — Localized Hadron Collision Experiment Platform

Alice's Intent: "Make a localized version of the CERN hadron collider."

HIIE Approach: This use case represents HIIE's most ambitious scope. The Theoretical Engineer and Civil & Structural Engineer immediately establish the physically achievable envelope. HIIE is honest about the energy gap:

$$\frac{E_{\text{CERN}}}{E_{\text{localized}}} = \frac{13.6\,\text{TeV}}{100\,\text{keV}} \approx 1.36 \times 10^8$$

A tabletop RF quadrupole linac achieves $E_{\text{cm}} \sim 10\text{–}100\,\text{keV}$, suitable for educational use and materials analysis. Cyclotron radius scales with momentum:

$$r = \frac{p}{qB} = \frac{mv}{\sqrt{1-(v/c)^2}\,qB}$$

For a compact design targeting $r \leq 0.5\,\text{m}$, field strength $B \approx 1\,\text{T}$ and beam energy $\sim 10\,\text{MeV}$ are achievable with superconducting coils.

Ethics Officer reviews radiation safety under NRC 10 CFR Part 30. Feasibility Manager flags the gap between popular conception and buildable reality. HIIE designs what is buildable, not what is aspirationally described.

Expected Outcomes: Feasibility-scoped design (tabletop cyclotron/linac), radiation shielding specification, CAD files, safety protocols, patent landscape analysis.

§11 Ethics Board & Self-Governance Model

Ethics is not a terminal filter — it is a continuous layer woven through every stage of reasoning and generation. No technological capability justifies ethical compromise. This is an architecture decision, not a policy position.

Six Core Pillars

- **Harm Prevention** — Dual-use detection on all outputs. Weaponizable designs trigger mandatory Alice review.
- **Environmental Responsibility** — All designs carry a carbon impact score. Net-negative outputs flagged; greener alternatives actively recommended.

- **Full Transparency** — Every recommendation includes a complete reasoning trace. No black-box outputs.
- **Human Primacy** — HIIE has no autonomous consequence. Every real-world physical action requires explicit human authorization from Alice.
- **Equitable Design** — Outputs must not create systems that systematically disadvantage protected groups or enable inappropriate concentration of power.
- **Immutable Audit Trail** — Tamper-evident log of all ethics decisions per project, exportable for third-party review.

Escalation Protocol

Ethics Officer → Arthur (AI Co-Director) → Alice (human principal). Risk scoring $R \in [0, 100]$:

$$R > 70 \Rightarrow \text{Alice sign-off required}$$

$$R > 90 \Rightarrow \text{Documented external human review required}$$

HIIE cannot self-modify its ethics framework — this layer is protected from all instruction overrides.

> **ABSOLUTE CONSTRAINTS — CANNOT BE OVERRIDDEN BY ANY INSTRUCTION**
>
> HIIE will never generate designs for weapons systems, biological or chemical weapons, mass surveillance infrastructure, or any system whose primary purpose is to cause physical harm. These are architecture-level constraints, not configurable policies. They cannot be overridden by Alice, Arthur, any client, Registered Agentics, or any prompt instruction.

§12 Data Strategy — Storage, Streaming & Retrieval

Three-Tier Data Model

Tier	Location	Content	Persistence
Hot (Active)	256GB SSD	Model weights, active project state, vector index shards, OS, services, active LoRA adapters	Persistent — managed rotation by project lifecycle
Warm (Archive)	5TB HDD	Completed outputs, full ChromaDB, curated datasets, model checkpoints, versioned adapter archive	Persistent — compressed and indexed
Stream (Live)	RAM only	Real-time web content, live patents, GitHub, papers	Never written to disk — analyzed in RAM, raw content discarded

Training Partition	HDD (ChromaDB)	Labeled project outcome pairs for ANE fine-tuning	Persistent — partitioned within ChromaDB, batch-consumed by ANE process

Research Group agents retrieve and analyze live internet sources entirely in the 8GB streaming RAM buffer. Only extracted structured insights are embedded and written to ChromaDB on HDD.

ChromaDB Collection Architecture

ChromaDB performs Approximate Nearest Neighbour (ANN) search natively via HNSW (Hierarchical Navigable Small World graphs). **No additional binary search layer is implemented** — HNSW handles all ANN retrieval optimally for HIIE's embedding workloads at the operating collection sizes.

HIIE partitions ChromaDB into three explicit, non-overlapping collections:

» **Active RAG Collection** — hot vector shards for all currently active projects. This collection's index resides on SSD (§3.2), not HDD, for sub-millisecond retrieval during agent reasoning. At project close, the collection is migrated to the archive collection on HDD.

» **Training Partition** — labeled output pairs from completed projects, batch-consumed by the ANE background fine-tuning process (§8.9). Partitioned separately to prevent live retrieval queries from contaminating the training data distribution.

» **Archive Collection** — embeddings from completed projects retained for cross-project retrieval and long-term pattern analysis. Stored on HDD; queried only on explicit demand, never during active inference.

Metadata Indexing. All embeddings across all collections carry four mandatory metadata fields, indexed at write time:

Field	**Type**	**Purpose**
`project_id`	String (UUID)	Scopes all retrieval and audit queries to a specific project
`agent_role`	Enum	Filters embeddings by the producing agent; enables per-role retrieval
`fscore`	Float [0–100]	Enables quality-gated retrieval — training partition queries can filter out low-scoring outputs
`adapter_version`	String	Tags embeddings with the LoRA adapter version active at generation time; supports adapter regression analysis

These fields are enforced at write time — any embedding submitted without all four fields populated is rejected by the FastAPI ingestion layer before reaching ChromaDB.

HDD Directory Structure

The 5 TB HDD uses a deterministic directory hierarchy. Every path is structured and machine-readable, enabling automated archival, retrieval, and adapter version management without filesystem scanning.

HDD ROOT: /hiie-archive/

```
/projects/{id}_{client}_{date}/
   /outputs/       -- final deliverable package
   /specs/         -- spec sheets and BOMs
   /cad/           -- FreeCAD, KiCad, SVG files
   /patents/       -- patent drafts and prior art records
   /ethics/        -- immutable Ethics Officer reports
   /checkpoints/   -- serialized agent state snapshots
   metadata.json   -- project manifest and F_score summary

/adapters/{agent_role}/{version}_{date}/
   versioned LoRA adapter files per agent role
   active → symlink to current deployed version

/chromadb/
   /training_partition/  -- labeled pairs for ANE fine-tuning (§8.9)
   /project_embeddings/-- archive collection (completed projects)
   /archive/       -- compressed snapshots for point-in-time restore
```

Project directory naming follows `{id}_{client}_{date}` where `id` is the 8-character project UUID prefix, `client` is a sanitized client identifier, and `date` is the project close date in `YYYYMMDD` format — producing lexicographically sortable, human-readable paths with no filesystem metadata dependency.

Adapter versioning follows `{version}_{date}` under each agent role directory, consistent with the versioning scheme in §8.9.6. An `active` symlink in each role directory points to the currently deployed adapter. Treasury Agent rollback (§8.9.6) operates by atomically repointing this symlink — no file moves required.

ChromaDB persistence maps each collection to a dedicated subdirectory, matching ChromaDB's native per-collection layout. The `/archive/` subdirectory holds periodic compressed snapshots of the `project_embeddings` collection, enabling point-in-time restoration without full ChromaDB reconstruction.

Data Sources by Domain

Domain	Sources
Patents	USPTO, EPO Espacenet, Google Patents, WIPO PatentScope, JPO
Research Papers	arXiv, PubMed Central, IEEE Xplore (OA), ChemRxiv, bioRxiv
Hardware & Standards	NIST, IPC, JEDEC, Mouser/Digikey datasheets, OSHWA
Code	GitHub public repos, crates.io, PyPI, npm documentation
AI Research	Hugging Face papers, arXiv cs.AI, Semantic Scholar
Materials	Matweb, ASM International (OA), NIST Materials Data Repository
Manufacturing	JLCPCB/PCBWay capability specs, industrial supplier databases

Environmental	EPA databases, carbon accounting frameworks, lifecycle databases

§13 Self-Hosted UI Toolchain — Autonomous Visualization & Implementation

HIIE's outputs are not static documents delivered to Alice for manual processing. The self-hosted toolchain enables AI agents to **autonomously generate, render, and iterate** on designs within a pre-configured visual environment. Every tool listed below is pre-installed, containerized under Coolify, and accessible via authenticated web interface.

Toolchain Overview

Tool	Category	HIIE Integration
FreeCAD (headless + GUI)	3D/Mechanical CAD	Agents invoke FreeCAD Python API to generate and validate 3D models autonomously. Alice views results via browser-accessible VNC session. Supports STL, STEP, DXF export.
KiCad (headless + GUI)	PCB/Schematic Design	Hardware Engineering agent generates KiCad project files programmatically. Gerber file export automated. Visual review in KiCad GUI via VNC.
Stirling PDF	Document Tools	All whitepapers, patent drafts, spec sheets, and reports rendered to PDF via Stirling PDF API. Alice downloads directly from OpenClaw.
Nextcloud	File Management	Central file system for all project assets. Agents write outputs here; Alice accesses via web browser. Version history maintained per project.
Tectonic / LaTeX	Document Generation	Whitepapers and yellow papers compiled from HIIE-generated LaTeX source by Documentation Architect agent.
Grafana + Prometheus	System Monitoring	Real-time dashboards for RAM, CPU, GPU, and agent queue status. Alice monitors system health during long-running projects.
Open WebUI	Model Interface	Local LLM interface for direct Alice-to-model interaction outside formal project flows.
Netlistsvg	PCB Visualization	Automated netlist visualization from KiCad exports, rendered to SVG for patent drawings.
Structurizr Lite	Architecture Diagrams	C4 architecture diagram generation for all software system outputs.
PlantUML server	UML Diagrams	UML diagram rendering embedded in documentation workflows.

OpenLCA	Lifecycle Assessment	Environmental lifecycle analysis for all material and manufacturing outputs.

Layering and Cross-Section Visualization

For mechanical designs (use cases 1, 5, 7), HIIE generates layered FreeCAD assemblies where each sub-system is an independent body on a named layer. Alice can:

- Toggle layer visibility to inspect internal structures (e.g., view PCB traces inside an enclosure)
- Export cross-section views as SVG for inclusion in patent drawings
- Run FreeCAD's FEM workbench for finite element stress analysis
- Animate kinematic assemblies to validate mechanical motion

Self-Hosted Architecture Auditing

- **Custom audit dashboard** — FastAPI endpoint queries ChromaDB for all decisions, flags, and validation results on a given project, rendered in Grafana
- **Immutable ethics log viewer** — read-only interface for all ethics officer reports, exportable to PDF via Stirling
- **Dependency graph renderer** — visualizes agent task dependencies and completion state in real-time
- **BOM diff tool** — compares successive BOM versions to flag cost or sourcing regressions between design iterations

§14 Output Engine — Deliverable Types

Output Type	Format	Description
Technical Spec Sheet	PDF + Markdown	Full engineering spec: dimensions, tolerances, performance targets, material callouts
Bill of Materials	XLSX + JSON	Components and materials with US sourcing, costs, MOQ, live availability
Patent Draft	PDF	Provisional patent application with claims, abstract, technical drawings
Viability Report	PDF	Feasibility scores, cost estimate, timeline projection, risk matrix
Schematic / CAD Files	KiCad / FreeCAD / SVG	Machine-readable design files, auto-generated by agents
Code Package	GitHub-ready repo	Production-grade, documented, tested code with CI configuration
Environmental Report	PDF	Carbon footprint, net-positive pathway, material lifecycle analysis
Ethics Assessment	PDF	Full ethics board review, risk scores, flags, resolutions — immutable

Process Flow Diagram	SVG + PDF	Manufacturing process visualization
Technical Whitepaper	PDF + HTML	Full whitepaper for commercialization, fundraising, or publication
Yellow Paper	PDF	Formal mathematical/protocol specification for cryptographic or novel protocol designs

§15 Distribution & Commercialization Plan

Stage 1 — Internal Capability (Months 1–6)

HIIE used exclusively by Arthur Labs. Every client project benefits from HIIE-quality output: validated spec sheets, patent strategy, feasibility reports. Arthur Labs competes differently from any other development organization. Self-funding through existing contracts. Registered Agentics hosts the primary Mac Mini node.

Stage 2 — API Product (Months 6–18)

Tier	Price	Includes
Explorer	$299/month	5 projects/month, core spec + BOM + viability
Builder	$999/month	20 projects/month, full output package + patent draft + dedicated Mac Mini at Registered Agentics
Enterprise	Custom	Unlimited projects, dedicated agent config, white-label, SLA, Registered Agentics hardware contract
Developer API	$0.05/token	Pay-per-use pipeline access for third-party integrations

Stage 3 — Licensed Platform (Months 18–36)

HIIE licensed as a deployable platform to large manufacturers, defense contractors (non-weapons applications), and universities. Licensees run on their own infrastructure or via Registered Agentics. Arthur Labs provides model updates, ethics framework maintenance, and support contracts.

IP Strategy

- Arthur Labs retains full IP on HIIE system architecture, pipeline, ethics framework, and all domain modules
- Clients own the IP in outputs generated for their projects — no Arthur Labs claim

- Open-source non-core components to build developer ecosystem; keep inference core and ethics layer proprietary

§16 Patent Strategy & Provisional Filing Fees

Provisional Patent Applications

HIIE generates provisional patent application drafts for client review. Provisional applications establish a priority date without requiring full examination. Key fee structure (USPTO, effective 2025):

Entity Type	Provisional Filing Fee	Notes
Micro Entity	$320	$\leq$ 4 prior applications, income threshold
Small Entity	$640	$<$ 500 employees
Large Entity	$1,600	Standard rate
Maintenance Fees (Non-Provisional, After Grant)		
3.5 years	$800 (small) / $1,600 (large)	
7.5 years	$1,800 (small) / $3,600 (large)	
11.5 years	$3,700 (small) / $7,400 (large)	

Note: Fees subject to change. HIIE-generated drafts are reviewed by a licensed patent attorney before filing. HIIE documents human-in-the-loop involvement at every step to satisfy USPTO inventorship requirements.

Global Patent Coverage — Estimated Costs

Jurisdiction	Filing Route	Estimated Cost
United States	Provisional + non-provisional	$3,000–$15,000
Europe (EPO)	EPC application	EUR 5,000–15,000
Japan (JPO)	National phase	JPY 200,000–600,000
PCT (International)	PCT application	$3,500–$8,000

HIIE's Patent Strategist tracks all deadlines and generates cost projections as part of every project's viability report.

§17 Phased Development Roadmap

Phase	Milestones

Phase 1 (Months 1–6) *Foundation*	Deploy OpenClaw + Coolify on Mac Mini M4 Pro at Registered Agentics. Implement Socratic Dialogue Engine. Build Hardware Engineering module. Integrate Patent Strategist with USPTO/Google Patents. Deliver first HIIE-generated project to Arthur Labs client. Begin populating ChromaDB. Deploy self-hosted UI toolchain (FreeCAD, KiCad, Stirling PDF, Nextcloud). Build ANE training data partition into ChromaDB — project outcomes begin accumulating as labeled training pairs.
Phase 2 (Months 6–12) *Full Agent Team*	Launch all domain modules. Ethics Officer with full scoring and audit logging. PySpice electrical simulation. Full project agent teams. Anti-slop validation framework fully operational. Launch API product. Target: 10 paying external clients. Registered Agentics multi-machine hosting live. First LoRA adapter deployment — Feasibility Manager and Patent Strategist adapters trained on Phase 1 outcomes. F_{score} delta tracking begins.
Phase 3 (Months 12–24) *Simulation + Fine-Tuning*	FreeCAD mechanical simulation. Full factory line and batch manufacturing design. Environmental impact scoring in all modules. Full specialist adapter suite across all domain agent roles deployed. Adapter versioning live on HDD. Arthur matures as persistent cross-session companion. Scale to 100+ API clients. Builder/Enterprise clients begin receiving year-one machines with trained adapter packages.
Phase 4 (Months 24–48) *Platform + Custom Hardware*	Purpose-built inference cluster designed by HIIE itself. LPU/custom chip generation. Multi-layer ANE fine-tuning as research substrate matures. Full autonomous design-to-patent pipeline. Platform licensing. HIIE designs its own next-generation compute hardware — potentially including custom NPU architectures optimized for the fine-tuning loop it has been running.

§18 Risks, Limitations & Mitigations

Risk	Likelihood	Impact	Mitigation
Model hallucination in engineering specs	High	High	Simulation validation required before any design clears pipeline; physical ground-truth anchoring mandatory
RAM pressure under full multi-agent load	Medium	Medium	Celery async queuing; state serialization to SSD above 85% memory pressure; N_{active} throttle enforced
Patent data staleness creating prior art blind spots	Medium	Medium	Live retrieval on every project — no cached patent data older than 24 hours used in novelty analysis

Ethics framework circumvention	Low	Very High	Hard-coded constraints in protected system layer; Ethics Officer runs as a separate model instance
Large lab models closing capability gap	High	Medium	HIIE's moat is the full pipeline + ethics governance + domain specialization + self-improving feedback loop
Single hardware node failure	Low	High	Daily HDD backup; Coolify restart policies; all project state checkpointed to SSD; Registered Agentics on-site maintenance
Legal uncertainty around AI-generated patent claims	Medium	High	HIIE generates drafts for human attorney review — not direct filing; human-in-the-loop documented at every step
Recursive loop / stack overflow	Medium	Medium	Hard depth limit $D_{\max} = 8$; cosine similarity divergence enforcement; Celery task timeout at 300s per agent
HCT economy miscalibration causing agent starvation	Low	Medium	Treasury Agent monitors spending velocity; reserve pool absorbs cascade failures; α,β,γ coefficients recalibrated per project by Treasury Agent
Alice intake gate creating friction rather than convergence	Low	Medium	Gate returns one targeted question per cycle, never a list; Exploratory Mode bypasses gate entirely; Arthur surfaces calibration recommendations if pass rate degrades below 70%
System-economy reputation scores calcifying around early data	Low	Low	Exponential decay (λ_d, 90-day half-life) ensures recent performance always outweighs historical; Arthur flags stale weight distributions
ANE API instability across macOS updates	Medium	Medium	Adapter store and training pipeline version-pinned; rollback to last stable macOS version if ANE API changes; Phase 1–2 training data accumulation is not blocked by ANE API instability
LoRA adapter degradation from low-quality training batches	Low	Medium	Adapter versioning with rollback; Treasury Agent monitors F_{score} delta per adapter version; degraded adapters automatically flagged and rolled back

§19 Conclusion

The Hyper Intelligent Innovation Engine represents a genuine category creation in applied AI — not a tool, not an assistant, but a complete innovation organization compressed into an ethically-governed intelligence system. Where existing AI systems excel at fragments of the innovation pipeline, HIIE is designed to own the entire chain from human intent to physical engineered reality. It is what emerges when the reasoning capabilities of frontier language models are combined with the disciplined, domain-grounded methodology of a world-class engineering firm — the closest existing analogy being what one would achieve if CRISPR-GPT and a frontier reasoning model produced a system capable of touching physical space.

The Mac Mini M4 Pro implementation, hosted at Registered Agentics, proves viability at minimum hardware cost, generating commercial revenue before scaling to purpose-built infrastructure. The addition of the ANE Persistent Intellectual Capital layer (§8.9) means the machine does not merely execute projects — it compounds from them. Every validated output encodes learning into specialist model weights, silently, on dedicated silicon, without interrupting active inference.

Ethics and anti-slop validation are not HIIE's constraints — they are HIIE's competitive advantage. As AI systems become more capable, those trusted with consequential engineering decisions will be the ones with demonstrably robust human oversight, embedded governance, and validation architectures that prevent plausible-sounding nonsense from entering the physical world.

The future of machines will not be designed by humans alone, nor by AI alone. It will be designed through the collaboration HIIE is built to enable — human intent directed by values, executed with engineering precision, validated against the physical world, and improved with every project that completes.

§20 Terms, Legal Compliance & Liability

Intellectual Property Notice

Arthur Labs, Inc., incorporated in Wyoming, retains all intellectual property rights in the HIIE system architecture, pipeline design, ethics governance framework, anti-slop validation methodology, and all domain modules described in this whitepaper. Outputs generated by HIIE on behalf of a client are the exclusive property of that client under the applicable service agreement. Arthur Labs asserts no ownership over client-directed outputs.

Patent and Filing Disclaimer

HIIE generates provisional patent application drafts and patent strategy analyses as an informational service. These materials are **not legal advice** and do not constitute the work product of a licensed patent attorney. All patent filings must be reviewed, amended, and submitted by a licensed patent practitioner in the relevant jurisdiction. Alice (the directing human principal) is responsible for ensuring compliance with USPTO inventorship and duty-of-disclosure requirements. Arthur Labs makes no representations regarding patentability, freedom to operate, or enforceability of any HIIE-generated patent draft.

Engineering Liability Disclaimer

HIIE-generated engineering specifications, schematics, bills of materials, process designs, and other technical outputs are **provided for informational and prototyping purposes only**. They have not been certified, stamped, or reviewed by a licensed professional engineer (PE) unless explicitly contracted. Any implementation of HIIE outputs in a physical product, manufacturing environment, or safety-critical application must be reviewed and approved by qualified licensed engineers before use. Arthur Labs, Inc. accepts no liability for physical, financial, or consequential damages arising from the implementation of HIIE outputs without appropriate professional review.

Registered Agentics Service Terms

Registered Agentics, operated by Marko Ruble, provides physical hardware hosting, registered agent services, and logistics under separate service agreements. Arthur Labs, Inc. and Registered Agentics are independent entities. HIIE's intellectual property and ethical constraints are governed solely by Arthur Labs, Inc. Registered Agentics has no authority to modify, override, or grant exceptions to HIIE's ethics framework or operational constraints.

Data and Privacy

HIIE does not retain raw internet content analyzed during live retrieval sessions — it is processed in RAM and discarded. Structured embeddings stored in ChromaDB are retained as part of the project record under the applicable client service agreement. The ANE fine-tuning training partition stores only labeled output pairs — no raw internet content, no PII. Clients should not submit personally identifiable information (PII) as part of project inputs unless explicitly covered by a data processing agreement with Arthur Labs, Inc.

Version and Architecture Notice

This document (HIIE v1.0.5, March 2026) represents the architectural specification as of its publication date. It is subject to revision as development progresses. The most current version is maintained internally by Arthur Labs, Inc. Ambition grounded in ethics and engineering rigor is how the future gets built.

Governing Law

This whitepaper and any service relationships described herein are governed by the laws of the State of Wyoming, United States of America. Disputes shall be subject to binding arbitration in Laramie County, Wyoming.

Watson Lewis-Rodriguez, Founder & CEO | Omaha, Nebraska — Wyoming Incorporated
Co-Author / Sponsor: Marko Ruble, Registered Agentics |
https://registeredagentics.vercel.app/
HIIE v1.0.5 | March 2026