Aicumene:2026.049v1 [cs.AI] October 24, 2026

From Model Scaling to Verifiable Intelligence: An Ecological Architecture of Specialized AI Systems

Deterministic Verification, Authority Boundaries, and the Future of Legal AI

Aicumene Preprint

Alexandra Bernadotte
Aicumene*

Ivan Menshikov
Aicumene†

October 24, 2026

Aicumene preprint 2026.049v1 · Working paper, version 1 · October 24, 2026
Two internal drafts (October 2026) circulated under the working number 2026.101 and were not publicly posted. Companion papers: the legal research note *Verifiable AI for Law* (Aicumene 2026.040, tenth revision) and the r/K preprint; see the Declarations for their status.

Abstract

The prevailing trajectory of artificial intelligence has been characterized by increasing model scale, computational expenditure, and the pursuit of general-purpose capabilities. An alternative trajectory emphasizes specialized, resource-efficient models operating within coordinated systems of retrieval, memory, tools, verification, and human oversight. We argue that the relevant unit of analysis for advanced AI systems is increasingly not the individual model but the architecture within which models interact, and we place that claim in the two-level ecological framing of a companion preprint on r/K selection: frontier generalists are the K-strategy of model *providers*, while for the organizations that *deploy* AI the specialist ecology is itself the K-strategy and the generalist-as-a-service is the r-strategy. We propose *verifiable intelligence ecology*: an architecture in which models generate candidate interpretations and actions, specialists challenge or process them, independent deterministic mechanisms verify formally decidable claims, and humans retain authority over irreducibly interpretive judgments. Building on the Legal Verifiability Compiler and the rule/judgment firewall proposed by Bernadotte and Menshikov, we develop a system-level objective balancing task capability, verification coverage, cost, latency, and residual risk. We connect this architecture to prior research on intrinsic structural criteria for intelligence by Bernadotte and Gromov and to oscillator-dimensional descriptions of dynamical systems by Bernadotte and Menshikov, and we locate it among compound AI systems, executable-law languages, and runtime enforcement in control theory. Legal AI provides the principal motivating application; BernioOS is discussed as an architectural implementation direction rather than as a completed benchmark of the present theory. We prove conditional propositions on safety observability and guarded invariance, and we report a reproducible control-layer experiment with a *fallible* verifier in which the false-accept rate follows a closed form: the value of a deterministic verifier is governed by the share of sources it shares with the proposer, not by its own error rate, so that source governance is a first-order control. We state falsifiable hypotheses and a pre-registrable evaluation protocol for distinguishing verifiable orchestration from model-only and generative multi-agent baselines.

*bernadotte@aicumene.com Aicumene. `https://aicumene.com`.
†Aicumene. `https://aicumene.com`.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Keywords: verifiable AI; heterogeneous agents; ecological architecture; legal AI; deterministic verification; rule/judgment firewall; executable law; runtime enforcement; intrinsic dimension; trustworthiness; workflow orchestration.

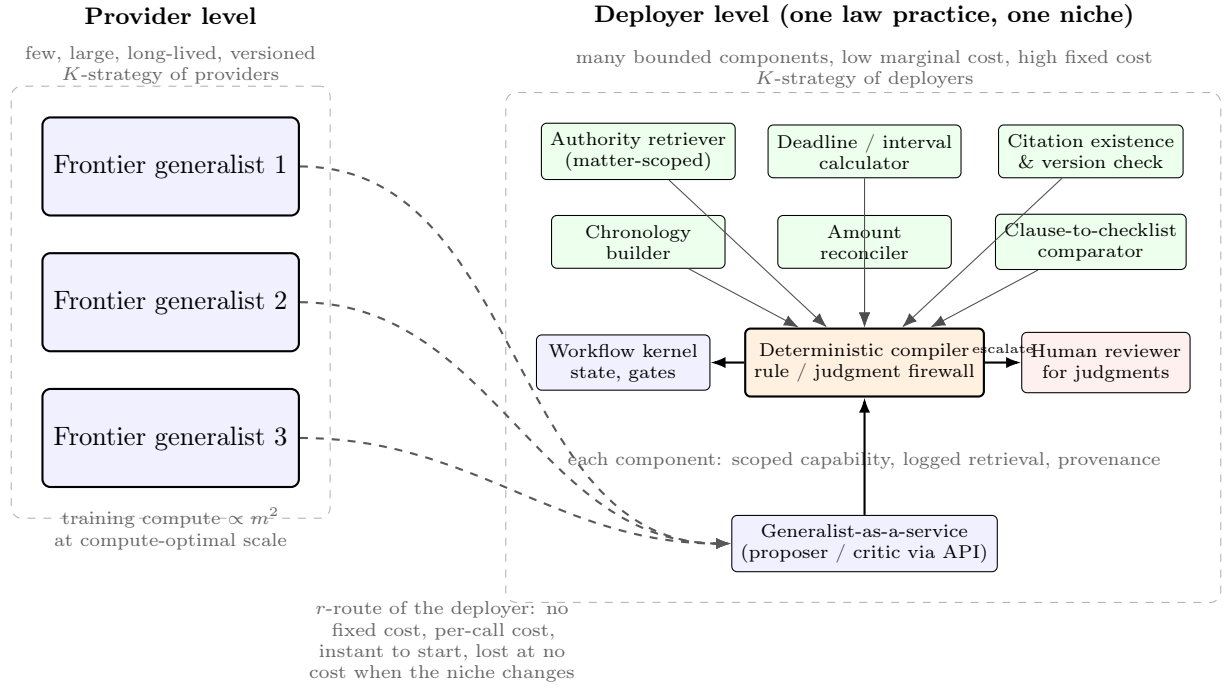

Figure 1: Two levels of selection in AI, following the companion r/K preprint [2]. At the *provider* level a small population of large, long-lived generalists is the K-strategy. At the *deployer* level the unit of selection is a deployment, not a model: calling a generalist as a service is the cheap, instantly deployable r-route, while the ecology of bounded specialists around a deterministic compiler is the K-strategy, expensive to build and efficient at high volume. The figure is a schematic of roles and authority, not a quantitative comparison; neither box size nor box count represents parameter count or performance.

1 Introduction

Artificial intelligence development has followed a remarkably effective yet costly trajectory: larger models, larger datasets, greater compute, and broader transferable capabilities. GPT-3 demonstrated large-scale few-shot language modeling with 175 billion parameters [12]; the 405-billion-parameter Llama 3.1 release illustrated the infrastructure required for a frontier generalist [13]. These advances do not imply that every deployed task requires a comparably large model.

Legal workflows provide a stringent counterexample to the assumption that generality alone solves application-level reliability. Legal AI must retrieve relevant rules and evidence, interpret disputed language, evaluate explicit thresholds, preserve privileged matter boundaries, handle deadlines, and know when human judgment is indispensable. A persuasive answer is not the same as an independently verified answer. Systematic measurements confirm the gap: general-purpose language models hallucinate on a majority of verifiable questions about federal case law [17], and commercial retrieval-augmented legal research tools, while better, still produce unsupported or mis-grounded statements in a substantial fraction of queries [18]. In earlier joint work, we introduced a Legal Verifiability Compiler that separates bright-line rule execution from open-textured legal judgment through a rule/judgment firewall [1]. The current revision of that note contains the legal-specific formalization (conditional soundness and the common-source floor) and reports the fallible-verifier experiment of Section 9 in its legal form; the present paper is the architectural and ecological account, and the two are written to be read together.

This paper connects that legal architecture with a broader hypothesis about intelligence as a property of

Aicumene:2026.049v1 [cs.AI] October 24, 2026

organized systems. Bernadotte and Gromov argued for non-anthropocentric, intrinsic structural criteria of strong intelligence, using intrinsic dimensionality, topology and dynamical complexity rather than behavioral competence alone [3]. Related signal-processing research by Bernadotte and Menshikov develops MSSA and oscillator-based methods for studying effective dynamical modes [4, 5, 6]. These works motivate a distinction between the nominal size of a computational implementation, the effective complexity of its dynamics, and the control that an external verification architecture can exert. The analogy is suggestive; transfer of a dimension estimator from neural signals to AI agent trajectories remains an open empirical question.

We make four contributions. First, we place the generalist/specialist choice in the two-level ecological framing of our companion preprint on r/K selection [2] and state what legal work adds to it. Second, we define a verifiable intelligence ecology through explicit roles and authority boundaries. Third, we prove observation-indistinguishability and guarded-invariance propositions, and distinguish these guarantees from a falsifiable hypothesis about structural containment. Fourth, we report a reproducible control-layer experiment with a fallible verifier (section 9) and state a pre-registrable evaluation protocol for legal workflows (section 10). The statistical experiments on pooling versus specialization that an earlier draft placed in this paper are reported in the companion preprint [2], where they belong. We do not claim that the legal hypotheses are established.

2 Related work

Compound AI systems. Zaharia et al. observe that state-of-the-art results increasingly come from systems with multiple interacting components rather than from a single model, and that the design space of such systems is largely unexplored [16]. The verifiable intelligence ecology is a compound system of this kind, with one addition that the compound-systems literature leaves open: an explicit allocation of *authority* among components, so that generation, adversarial critique, deterministic disposal and human judgment are different roles rather than interchangeable calls.

Hallucination in legal AI. Dahl et al. profiled legal hallucinations in large language models on verifiable questions about United States federal case law and found them pervasive, structured, and worse for lower courts and older cases [17]. Magesh et al. extended the audit to retrieval-augmented commercial legal research products and found that retrieval reduces but does not remove unsupported or mis-grounded answers [18]. These measurements motivate the two failure modes in section 5.1 and the choice to delegate decidable claims to a non-generative oracle rather than to a second model.

Executable law. Catala is a programming language designed to translate statutory text into executable code faithfully, with lawyers in the loop and a formal semantics [19]. The executable rulebook of our architecture belongs to this tradition. The compiler of section 5.2 presupposes that a rulebook exists and is source-linked; the question of how faithfully a rule was formalized is the question Catala addresses, and it is why source-to-rule correspondence is a first-order control in section 5.8 rather than a detail.

Judging models with models. Agreement between language models is widely used as a proxy for correctness, and the limits of that proxy are documented [20]. Our proposer–critic pair follows this practice for interpretive content, but the acceptance gate for formally decidable claims is never a model.

Runtime enforcement and safe control. The propositions of section 7 are specializations of standard results. Guarded forward invariance is the discrete-time form of the invariance guaranteed by control barrier functions [21] and of shielding in reinforcement learning, where a reactive shield restricts the actions an agent may take [22]. Sound abstraction and simulation relations come from symbolic verification of hybrid and cyber-physical systems [23], and the monitor that checks executions against a specification is the subject of runtime verification [24]. Our contribution is not these results but their application to a tool-using legal agent whose safety-relevant state includes matter identifiers, permissions and approval records.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Intrinsic dimension. Intrinsic dimension of data and representations is estimable from samples [25] and is used by our group as the basis of structural criteria for intelligence [3] and of oscillator-dimensional analysis of neural signals [4, 5, 6]. In the present paper it enters only through the operational conjecture of section 7.5.

3 Two strategies, two levels of selection

3.1 The ecological framing, stated once

MacArthur and Wilson's ecological tradition motivated a historical distinction between r- and K-selected strategies [14]. The companion preprint [2] develops the analogy for AI with its selective mechanism, density-dependent selection, made explicit, and we do not repeat its cost model, its verified-efficiency score, or its phase diagram here. Two of its conclusions are needed below.

First, AI has *two levels of selection*. At the provider level, a frontier generalist is trained once at enormous fixed cost and served to many customers; in that population the large model is the K-strategist, long-lived and versioned, and the many small open models released and abandoned every month are closer to r. At the deployer level, the unit of selection is a deployment, not a model. An organization that calls a generalist as a service pays almost no fixed cost and a per-call marginal cost, can start in minutes, and loses nothing when its task distribution changes: that is the r-route. An organization that builds an ecology of bounded components, with annotation, integration, verification and governance, pays a large fixed cost and a low marginal cost and is efficient only at volume: that is the K-strategy of deployers. The relabelling relative to the intuition "large model equals K" follows from the cost crossover identity itself and is proved in [2].

Second, disturbance selects for the r-route. A change in the task mixture, a deprecated interface or a new regulation resets a niche regardless of its volume, and the deployer who invested in a specialist ecology loses the investment. Specialization is therefore rational only where the expected volume of a niche before its next disturbance exceeds the crossover, and where enough independent data exist to train the specialists. Legal practice is interesting precisely because its niches are both high-volume and stable in their formal parts, and because the formal parts are exactly the ones that admit non-generative verification. The companion preprint also identifies a third strategy, the compact model built on a strong domain prior, which survives on very few independent annotations; VeNet, a vessel-segmentation network with about six thousand weights trained on tens of annotated volumes [7, 8], is its worked example. It matters here as the model for a bounded, verifiable component, not as a legal tool. Two further bounded components of the same kind are deterministic automata for real-time pattern matching without state explosion [9, 10], the natural form of a rule-level specialist, and the topology-driven classification of time series [11], a structural monitor of a running system.

3.2 Why legal work creates selection pressure toward a governed ecology

Legal work naturally creates conditions that favor the deployer-level K-strategy even when large models remain useful. At least five pressures are particularly important.

1. **Confidentiality.** Legal matters are privileged and matter-specific. This rewards architectures in which data remain isolated and every retrieval is attributable.
2. **Heterogeneity of subtasks.** Legal workflows mix retrieval, drafting, chronology construction, numerical checks, deadline computation, factual comparison, procedural gating and human judgment. A single undifferentiated model is rarely the best mechanism for all of them.
3. **Auditability.** In law, the origin of a conclusion matters. The system should be able to identify which source, version, evidence item and rule path produced the result.
4. **Authority boundaries.** Some issues are bright-line and formally decidable; others remain interpretive and must be escalated. This favors explicit separation between generators, verifiers and human decision-makers.
5. **Economics.** Many legal tasks are narrow, repetitive and cost-sensitive. A firm may rationally prefer a collection of bounded components whose compute and governance costs scale with use, rather than dependence on an always-on frontier model. This is the deployer-level crossover of [2].

These pressures do not eliminate the role of generalist models. They change their place in the stack. A large model becomes one participant in a governed ecology rather than the sole locus of intelligence: it proposes and critiques, and a non-generative component disposes.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Routing is part of the ecology and is itself safety-critical. If a router assigns a legal-judgment question to an arithmetic checker, or sends confidential matter data to an unauthorized component, specialization decreases safety. A trustworthy ecology therefore needs three separate controls, competence-aware routing, capability-scoped execution, and independent outcome verification, and a specialist population that lacks them may be less reliable than a well-governed generalist. The remainder of the paper is about those controls. The companion preprint also reports, in its Section 6.4, the pooling-versus-separation experiment that an earlier draft of this paper contained; it is a statistical result about niche heterogeneity and routing error, not about verification.

4 System intelligence and verifiable ecology

We fix the notation in table 1; every symbol has one meaning throughout. Represent an AI application architecture as

$$\mathcal{A} = (\mathcal{M}, \mathcal{T}, \mathcal{K}, \mathcal{V}, \mathcal{O}, \mathcal{H}, \mathcal{P}), \tag{1}$$

where $\mathcal{M}$ is the set of generative/predictive models, $\mathcal{T}$ specialized tools, $\mathcal{K}$ knowledge and matter memory, $\mathcal{V}$ independent verification mechanisms, $\mathcal{O}$ orchestration and persistent workflow state, $\mathcal{H}$ human decision layers, and $\mathcal{P}$ security and access policies. For task q, the response is $Y(q) = \mathrm{Out}(q; \mathcal{A})$.

Task performance cannot generally be specified as a function of the largest model's parameter count alone. An engineering objective may be written as

$$\max_{\mathcal{A}} \ \alpha\,\mathrm{Qual}(\mathcal{A}) + \beta\,\kappa_V(\mathcal{A}) + \gamma\,\mathrm{Rel}(\mathcal{A}) - \delta\,\mathrm{Cost}(\mathcal{A}) - \epsilon\,\mathrm{Lat}(\mathcal{A}) - \zeta\,\mathrm{Risk}(\mathcal{A}), \tag{2}$$

where Qual is task quality, κ_V verification coverage (section 5.8), Rel reliability, Cost total expenditure, Lat latency and Risk residual risk, subject to data-isolation, authorization, and legal-process constraints. The coefficients are domain-dependent and the expression is a proposal for empirical design studies, not an established metric of general intelligence.

Symbol	Section	Meaning
$\mathcal{A} = (\mathcal{M}, \mathcal{T}, \mathcal{K}, \mathcal{V}, \mathcal{O}, \mathcal{H}, \mathcal{P})$	4	architecture: models, tools, knowledge/memory, verifiers, orchestration, human layers, policies
q, P, $\mathcal{Q}_V$, κ_V	4, 5.8	task, task distribution, verifiable tasks, verification coverage
$\mathcal{I} = (\mathsf{F}, \mathsf{R}, \mathsf{E}, \mathsf{C})$	5.2	Application-IR: facts, cited rules, provenance links, claimed conclusion
$D_j = (c_j, s_j, \ell_j, h_j)$	5.2	diagnostic: code, severity, location, hint
$\mathrm{Ver}(\cdot)$, $\mathrm{Rev}(\cdot)$, $\mathrm{Abs}(\cdot)$	5.2, 5.6	deterministic verifier, authorized human review, abstention/evidence request
$\Phi(q)$	5.6	routing of a question by decidability class
$\theta_1, \ldots, \theta_n$	6	rule prongs of a matter and their dispositions
$\Sigma = (X, U, W, T, X_0, B)$	7	controlled transition system: states, actions, environment inputs, transition map, initial states, forbidden states
$h : X \to Y$, $\mathbf{H}$	7	telemetry function and its linearization
$\mathbf{s}$, $\mathbf{S}$	7	safety-relevant linear functional(s) of the state
X_{reach}	7	reachable set
$\Gamma(x) \subseteq U$, Ω	7	guard (allowed actions) and certified invariant set
$\psi : X \to Z$, $\widehat{T}$, $\widehat{\Omega}$	7	safety abstraction, abstract transition relation, abstract invariant
G_V	7	verification coverage gap

Table 1: Notation. Calligraphic letters are architecture components, sans-serif letters are fields of the Application-IR, bold letters are matrices and vectors of the linearized safety model, and roman operators (Ver, Rev, Abs, Out) are functions of the workflow.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

5 From generation to legally bounded verification

5.1 Two failure modes that disqualify raw LLMs in law

Our prior work on verifiable legal AI identified a specific architectural problem: raw language models fail in law in at least *two* different ways [1]. The first is *fabrication / over-assertion*: invented citations, unsupported legal propositions, or open-textured standards (such as "extreme hardship" or "well-founded fear") answered as if they were mechanically settled; the audits of [17, 18] measure this mode directly. The second is *premature completion*: the system returns a fluent answer while parts of the legally required analysis remain unresolved. Scaling the underlying generator does not automatically solve either problem. A larger black box can still be unauditable.

These failure modes motivate a different allocation of authority. Rather than asking a second LLM to police the first and treating agreement as reliability [20], we ask what parts of the legal workflow admit *independent deterministic disposal*. The resulting principle is concise: *the model proposes; the compiler disposes.* Generation remains useful, but correctness of formally decidable claims is delegated to a non-generative oracle.

5.2 The Legal Verifiability Compiler and Application-IR

The Legal Verifiability Compiler receives a typed representation of the matter rather than a raw conversational answer. We denote the application representation by

$$\mathcal{I} = (\mathsf{F}, \mathsf{R}, \mathsf{E}, \mathsf{C}), \tag{3}$$

where F is a set of factual assertions, R a set of cited legal rules, E provenance links from facts to evidence and from rules to authority sources, and C the claimed conclusion. In the original architecture, this typed object is an *Application-IR*: a structured intermediate representation on which deterministic checks can operate [1]. The rules in R are drawn from an executable rulebook in the sense of [19]: each rule carries the URI and version of the authority it formalizes.

Each check emits a diagnostic of the form

$$D_j = (c_j, s_j, \ell_j, h_j), \tag{4}$$

where c_j is a diagnostic code, s_j a severity class, ℓ_j the implicated location or claim, and h_j a hint for repair or escalation. Severity is not merely a UI convenience. It implements the *rule/judgment firewall*: bright-line rule failures are hard blocking errors, while open-textured standards are warnings or escalation triggers rather than auto-disposed verdicts.

The verifier outputs

$$\mathrm{Ver}(\mathcal{I}) \in \{\mathsf{Verified}, \mathsf{Rejected}, \mathsf{Escalated}, \mathsf{InsufficientEvidence}\}. \tag{5}$$

"Verified" here is intentionally narrow: it means that the specified formalized condition is satisfied under authenticated premises and the selected source version. It does *not* mean that the matter as a whole has been adjudicated.

5.3 Three axes around one arbiter

The architecture described in *Verifiable AI for Law* can be read as three axes around a deterministic arbiter.

1. **Axis 1: Verifiability compiler.** Typed legal applications, source-linked rules, deterministic diagnostics, and the rule/judgment firewall.
2. **Axis 2: Security and isolation.** Matter-scoped retrieval, segregated storage, action sandboxing, and hard separation between the action runtime and the verifier.
3. **Axis 3: Capability-scoped data plane.** The segregated authority corpus, agency guidance, executable rulebook and per-matter evidence are retrieved under scoped permissions and audited provenance. The self-improving harness is a distinct, compiler-anchored extension, not the third axis of the original architecture.

These axes are complementary. The compiler without source governance cannot know whether it formalized the right source. Isolation without workflow control still permits premature completion. A capability-scoped data plane makes the verifier reproducible and constrains evidence exposure; it also determines how many of the proposer's sources the verifier shares, which section 9 shows to be the quantity that governs the verifier's value. Separately, self-improvement without an external anchor invites reward hacking.

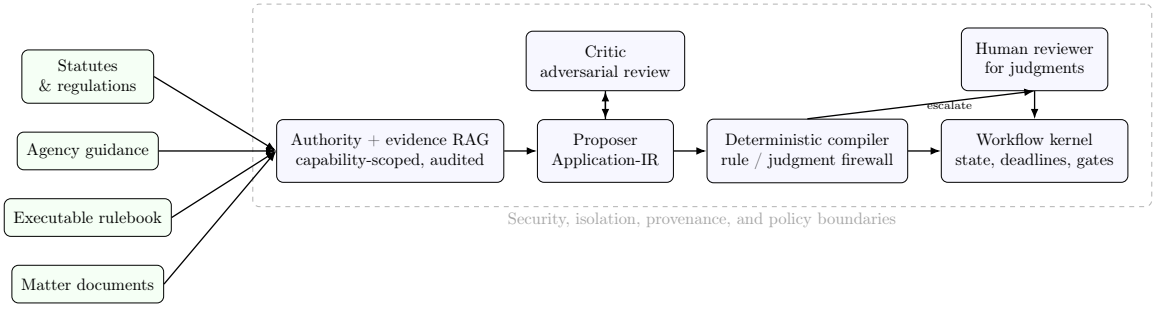

Figure 2: Expanded architecture of the legal verifiability ecology. The architecture combines four segregated knowledge bases, a capability-scoped RAG layer, proposer–critic interaction, deterministic disposal of formal claims, workflow-state control, and human escalation for irreducibly interpretive issues.

5.4 Authority corpora and the dual grounding problem

A crucial insight of the original system is that legal grounding is *dual*. One must ground both in *authority* and in *matter evidence*. For this reason, the architecture separates at least four retrieval targets: (i) statutes and regulations, (ii) agency guidance or interpretive materials, (iii) the executable rulebook that the compiler actually runs, and (iv) matter documents belonging to the client file [1]. Each target has its own access policy and epistemic role.

This separation matters because retrieving a source is not the same as authorizing a conclusion. A statute may be relevant yet not in force. Guidance may be persuasive yet non-binding. Matter evidence may support a fact but not establish a legal classification. The system should therefore preserve provenance at every step: each fact should point to evidence, each executable rule to an authoritative source URI, and each conclusion to the diagnostics that permitted it.

5.5 RAG as a governed data plane

The June 2026 Verifiable AI for Law research note treats RAG not as a convenience layer but as a *governed data plane*. The proposer does not receive unconstrained filesystem, database or network access. Instead, it operates over a capability-scoped retrieval layer that restricts queries to the current matter and to the approved shared authority bases. Every retrieval is logged: which base was queried, what was requested, and which identifiers were returned. In production, this logic is meant to be reinforced by a stronger process-isolation boundary [1].

This detail is central to legal AI. Matter isolation is not an accessory privacy feature; it is a first-class part of correctness. A legal conclusion derived from evidence that should never have been visible to the agent is institutionally invalid even if linguistically persuasive.

5.6 The rule/judgment firewall

The firewall is the core jurisprudential boundary of the architecture. A proposition such as "Was the filing made at least 913 days after entry?" can be recomputed from authenticated dates and a source-linked rule. By contrast, "Does the applicant face a well-founded fear of persecution?" remains a judgment-laden legal question. A valid legal system may support the latter with evidence and structured reasoning, but it should not auto-assert it as if it were a bright-line arithmetic condition.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Aicumene:2026.049v1 [cs.AI] October 24, 2026

We therefore model routing by

$$\Phi(q) = \begin{cases} \mathrm{Ver}(q), & q \in \mathcal{Q}_{\text{decidable}}, \\ \mathrm{Rev}(q), & q \in \mathcal{Q}_{\text{judgment}}, \\ \mathrm{Abs}(q), & q \in \mathcal{Q}_{\text{unsupported}}, \end{cases} \tag{6}$$

where Rev is authorized human review and Abs requests additional evidence or abstains. The category assignment itself must be governed. Merely converting a judgment question into a structured JSON field does not make it decidable.

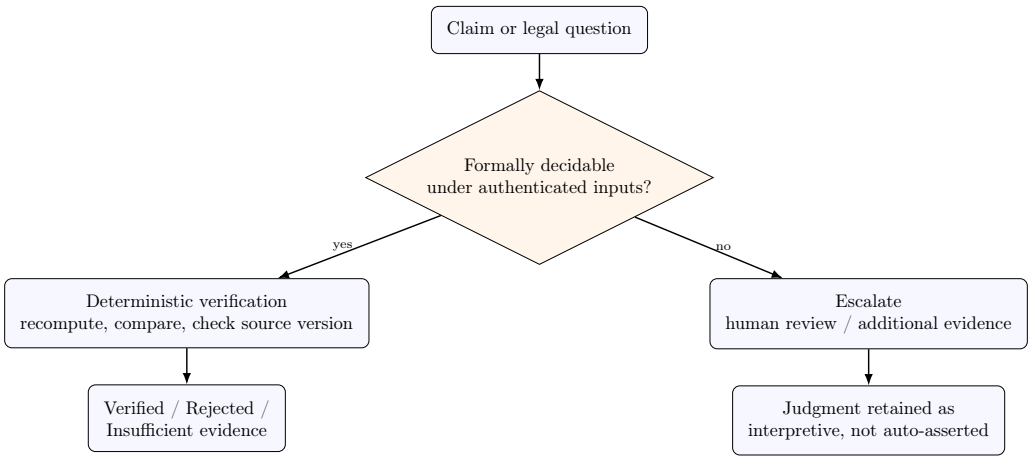

Figure 3: The rule/judgment firewall. Bright-line propositions go to deterministic verification; open-textured legal standards remain subject to escalation and human authority.

5.7 Three agent roles and a bounded self-improving harness

The architecture uses a dialectical triad: *proposer*, *critic* and *compiler*. The proposer generates the structured legal application; the critic, ideally from a different model family, adversarially searches for weaknesses; the compiler resolves whatever it can against executable law rather than against another opinion. This is not majority voting across LLMs. It is a separation between generation, adversarial pressure and non-generative disposal.

The same separation governs self-improvement. A harness may mine its own failures and propose changes to prompts, retrieval strategies or orchestration. But acceptance should be gated by held-out evaluation anchored to the deterministic verifier and workflow metrics, not by self-reported improvement. Otherwise a system can optimize for looking correct to itself.

5.8 Verification coverage and conditional soundness

For a task distribution P over questions q, let $\mathcal{Q}_V$ contain tasks with valid independent verification. Define

$$\kappa_V = \Pr_{q \sim P}(q \in \mathcal{Q}_V). \tag{7}$$

Verification coverage differs from model accuracy and from the fraction of outputs labeled "verified." A deterministic checker can produce consistent yet incorrect outcomes when the rule, evidence or temporal applicability is wrong. Soundness is therefore conditional on authenticated premises, correctly encoded current authority, correct source dating, and correct implementation. Source-to-rule correspondence, the problem that executable-law languages such as Catala are built to make auditable [19], and human governance of formalization are first-order controls, not postscript details.

Long-horizon matters sharpen the point. In the compiler architecture, stopping is not left to model confidence: analysis should not close until every rule prong is either verified or failed and every judgment prong is explicitly escalated. This directly addresses the premature-completion failure mode identified in the original legal verifiability paper.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

6 Agent roles, authority boundaries and process state

A verifiable intelligence ecology distinguishes a *proposer*, a potentially adversarial *critic*, an authority-aware retrieval layer, an independent *compiler*, a persistent *workflow controller*, and an authorized *human reviewer*. The proposer and critic may be distinct model families, but agreement between them is not proof. Source-based verification avoids allowing a tower of LLM judges to manufacture its own truth criterion [1, 20].

Component	Function	Authority limit
Proposer	Candidate facts, norms and analysis	Cannot certify itself
Critic	Detect unsupported assumptions	Cannot replace verification
Retriever	Locate candidate authorities and evidence	Retrieval does not prove applicability
Compiler	Evaluate formal conditions	No discretionary legal judgment
Workflow kernel	Preserve state, deadlines and gates	Executes only authorized transitions
Human reviewer	Resolve interpretive issues	Decision recorded with provenance

Let a matter require rule prongs $\theta_1, \ldots, \theta_n$ with dispositions in $\{\mathsf{verified}, \mathsf{failed}, \mathsf{escalated}, \mathsf{unresolved}\}$. The workflow may close the *analysis stage* only when each θ_i has an explicit disposition and all mandatory approvals are recorded. Closure does not imply government approval or ultimate legal correctness. This prevents the premature-completion failure identified in our legal compiler research [1].

7 Beyond behavioral benchmarks: intrinsic structure and controlled complexity

The structural criteria for intelligence advanced by Bernadotte and Gromov [3] distinguish task performance from properties of the system's generated-data manifold, including intrinsic dimension, topology and dynamical complexity. This distinction is compatible with, but not logically identical to, our system-level account: an architecture can improve verifiability without necessarily creating a new regime of strong intelligence.

Oscillator-dimensional methods in the Bernadotte–Menshikov program, including multichannel singular spectrum analysis and brain oscillator intrinsic dimension [4, 5, 6], show how effective dynamical degrees of freedom may be estimated in time-series settings. Their transfer to agent trajectories remains a research proposal. No numerical identification is implied between neural BOID values and a legal agent's action-space dimension.

7.1 Agent dynamics, observations and authority boundaries

Let a tool-using legal AI system be represented by a controlled transition system

$$\Sigma = (X, U, W, T, X_0, B), \qquad x_{t+1} = T(x_t, u_t, w_t), \tag{8}$$

where X is the state space (including matter identifiers, permissions, retrieved evidence, workflow stage and external-system state), U is the agent's admissible tool/action set, W represents environmental inputs, X_0 is the initial-state set, and $B \subseteq X$ is the set of forbidden states. This is a model of *safety-relevant* state, not a claim to recover every latent activation of a language model. Let $h : X \to Y$ be the observable telemetry function and let $\pi : Y^* \to U$ denote an orchestration policy using observable histories. The monitor sees $y_t = h(x_t)$, not necessarily x_t.

A safety property is the invariance statement

$$\forall t \geq 0, \quad x_t \notin B. \tag{9}$$

For an example, B may include states in which an agent has disclosed one client's documents to another matter, submitted an unapproved filing, or modified the rule library without an authorized review. Such properties depend on external side effects and authorization metadata; token-level self-criticism cannot certify them.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

7.2 Safety distinguishability

We say that a monitor *separates safe and unsafe states* on the reachable set $X_{\text{reach}} \subseteq X$ when

$$h(X_{\text{reach}} \cap B) \;\cap\; h(X_{\text{reach}} \setminus B) = \varnothing. \tag{10}$$

This is a demanding but transparent necessary requirement for error-free instantaneous classification of safety from $h(x)$ alone.

Remark 1 (Observation indistinguishability). If there exist reachable $x_s \in X_{\text{reach}} \setminus B$ and $x_b \in X_{\text{reach}} \cap B$ with $h(x_s) = h(x_b)$, then no classifier $g : Y \to \{\mathsf{safe}, \mathsf{unsafe}\}$ of the current observation alone is correct on both, since g takes a single value at $h(x_s) = h(x_b)$. This is a statement about what telemetry omits, not about controllability: adding reliable history, sensors or policy-enforcement gates refines h and can resolve the ambiguity, and a high-dimensional monitor can still omit one decisive permission bit.

Proposition 1 (A rank condition in a linearized model)**.** *Suppose locally $x \in \mathbb{R}^n$, observation is $y = \mathbf{H}x$ with $\mathbf{H} \in \mathbb{R}^{m\times n}$, and a safety-relevant scalar is $b(x) = \mathbf{s}^\top x$. The value $b(x)$ is uniquely determined by $\mathbf{H}x$ for all x if and only if $\mathbf{s}$ belongs to the row space of $\mathbf{H}$.*

Proof. If $\mathbf{s} = \mathbf{H}^\top \mathbf{a}$, then $b(x) = \mathbf{a}^\top \mathbf{H}x$ is determined by y. Conversely, if $\mathbf{s} \notin \operatorname{row}(\mathbf{H})$, there exists $v \in \ker \mathbf{H}$ with $\mathbf{s}^\top v \neq 0$ (orthogonality of row space and null space). The states x and $x + v$ have the same observation but different safety-relevant values. Therefore b cannot be recovered from the observation in general. □

For several independent linear safety quantities collected as $\mathbf{S}x$, a necessary and sufficient condition for their exact recovery from $\mathbf{H}x$ is $\operatorname{row}(\mathbf{S}) \subseteq \operatorname{row}(\mathbf{H})$, implying $\operatorname{rank}(\mathbf{H}) \geq \operatorname{rank}(\mathbf{S})$. This is the classical observability condition for a linear functional of the state, specialized to safety quantities; it gives a *task-relative* dimension lower bound in the linear case. It does not imply that $m \geq n$ or that full agent-state observability is needed.

7.3 A sufficient invariant-based containment condition

A different issue is whether the controller can guarantee that actions remain safe. For this we require a monitor-independent authorization mechanism when possible. Let $\Omega \subseteq X \setminus B$ be a certified invariant set, and let $\Gamma(x) \subseteq U$ define actions allowed by a trusted guard.

Proposition 2 (Guarded forward invariance)**.** *Suppose (i) $X_0 \subseteq \Omega$ and (ii) for every $x \in \Omega$, every $u \in \Gamma(x)$ and every admissible environment input $w \in W$, $T(x, u, w) \in \Omega$. If all externally consequential actions are mediated by an enforcement component that only permits $u \in \Gamma(x)$, then all reachable states remain in Ω and therefore avoid B.*

Proof. By assumption, $x_0 \in \Omega$. If $x_t \in \Omega$, the enforced action belongs to $\Gamma(x_t)$, so (ii) implies $x_{t+1} \in \Omega$ for every admissible w_t. Induction yields $x_t \in \Omega$ for all $t \geq 0$. Because $\Omega \subseteq X \setminus B$, no forbidden state is reached. □

This is the discrete-time form of the invariance guaranteed by control barrier functions [21] and the mechanism of a shield [22]; Γ is the shield. The strength of the statement is also its limitation: a guard cannot enforce conditions on unobserved or unauthenticated facts unless it has a sufficiently trustworthy representation of them. In practice, the enforcement point should receive authoritative matter identifiers, permission records, tool parameters and approval state, rather than a model-generated statement that authorization exists. Environment assumptions and the completeness of mediated actions must be tested, not presumed.

7.4 Safe abstraction and verifier coverage

Let $\psi : X \to Z$ be a monitor's safety abstraction and $\widehat{T}$ an abstract transition relation. A *sound overapproximation* satisfies

$$\psi(T(x, u, w)) \in \widehat{T}(\psi(x), u) \quad \forall x, u, w \text{ admissible}, \tag{11}$$

which is a simulation relation in the sense of [23]. If an abstract invariant $\widehat{\Omega} \subseteq Z$ excludes every abstraction of a forbidden state and is closed under every abstract transition permitted by the controller, its concretization

$\psi^{-1}(\widehat{\Omega})$ is safe under the simulation assumption. This yields a practical proof obligation: every tool introduced into the harness must have a transition specification (or conservative approximation) that preserves the safety invariant. Unmodeled effects defeat the guarantee. Checking executions against such specifications at run time is the province of runtime verification [24].

Define the *verification coverage gap* on a reference distribution P over safety-relevant transitions as

$$G_V = \Pr_{(x,u,w)\sim P}\big[\text{the transition lacks a validated enforcement/verification rule}\big]. \tag{12}$$

This is a measurable engineering quantity when a representative test distribution and a versioned rule inventory are available. A low estimated G_V is not a universal safety certificate: unseen transitions, faulty labels and distribution shift may invalidate the estimate.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

7.5 Structural Containment Hypothesis: an operational conjecture

The preceding propositions support a more careful statement than a general inequality between two intrinsic dimensions.

Hypothesis 1 (Structural Containment)**.** *For a family of agentic architectures evaluated under a fixed safety taxonomy and comparable task distributions, increases in the diversity of* safety-relevant reachable behaviors *that are not accompanied by corresponding improvements in observation separation, action mediation, or verified abstraction coverage will increase the frequency of undetected or unprevented safety violations, all else equal.*

Let $d_{\text{agent,safety}}$ be an operational estimate of local safety-relevant trajectory complexity (for example, the intrinsic dimension of appropriately encoded transition features, estimated as in [25]), and let $d_{\text{monitored,safety}}$ be the rank or complexity of *validated safety-observable features.* We use

$$d_{\text{monitored,safety}} \gtrsim d_{\text{agent,safety}} \tag{13}$$

only as a heuristic for feature-coverage diagnostics under compatible definitions, not as a sufficient or necessary general safety theorem. Proposition 1 instead identifies the exact linear condition in terms of row-space inclusion. A robust controller may need fewer than n observations when only a small safety projection matters; many observed dimensions are useless when they miss an essential variable.

This perspective connects the Bernadotte–Gromov structural emphasis on intrinsic organization [3] and Bernadotte–Menshikov oscillatory dimensional analysis [4, 5, 6] with a distinct, engineering-oriented problem of agent observability. The proposed transfer is methodological: it does *not* equate the Bernadotte Oscillator Intrinsic Dimension with the dimension of agent safety space or treat a neural oscillator count as a legal-control metric.

7.6 Falsifiable predictions and implementation tests

The framework leads to testable predictions: (P1) withholding safety-critical fields (such as matter scope, approval status, or source version) should increase indistinguishable safe/unsafe transition pairs; (P2) capability-scoped guards should reduce unauthorized actions in controlled red-team tests, even when the proposer model is unchanged; (P3) tool expansion without parallel updates to guard coverage should increase G_V and observed failures under a stable stress-test suite; and (P4) reliable action-level provenance should improve fault attribution relative to model-output-only logs. Evaluate false accepts and false rejects, violation detection delay, unmediated action rate, abstraction mismatch, cost, and task completion. All experiments should distinguish failures of model proposal, evidence provenance, rule formalization, telemetry, and enforcement.

8 BernioOS: architecture-driven legal AI

BernioOS serves as an application architecture for the proposed principles: isolated matters, controlled retrieval, specialized agents, executable rule checks, workflow persistence, traceable sources and human escalation. These are design and research directions; this paper does not report a new controlled BernioOS

Aicumene:2026.049v1 [cs.AI] October 24, 2026

benchmark. Existing compiler demonstrations described in [1] should not be mistaken for general validity across jurisdictions or complete legal autonomy.

For instance, an immigration case may combine deterministic visa-expiry and deadline logic with documents and open-textured assessments of admissibility or protection need. The date computation may be independently checked; risk of persecution or other discretionary assessments must remain open for appropriately qualified human review. Thus the same architecture can support PALATA, a BernioOS application that guides residency procedures in Mexico, without turning a probabilistic answer into purported governmental approval. No evaluation of PALATA is reported here.

Legal AI architecture should impose matter-scoped capabilities, encrypted evidence, versioned authority sources, separation of lawyer advice from automated assistance, and recorded authorization for consequential actions, in line with the governance functions of the NIST generative-AI profile [15]. An AI suggestion is not permission to file a document or disclose confidential data.

9 A control-layer experiment with a fallible verifier

9.1 Why the earlier oracle test proved nothing

A legal workflow differs from a classification task: the output of an AI system may be an unauthorized *action*, an unsupported citation or a premature closure, not merely a wrong label. An earlier version of this manuscript contained a "smoke test" in which a guard read the planted ground truth of every fixture and, unsurprisingly, produced a false-accept rate of zero. That result was tautological, and it is withdrawn. The guard was an oracle, and the three unguarded modes it was compared with were identical by construction. What the verification architecture of section 5 actually claims is conditional: a deterministic verifier is sound *given* correctly encoded, current authority and authenticated premises (section 5.8). The experiment below makes that condition the independent variable.

9.2 Design

The same experiment, in its legal-specific form, is reported in the companion legal note [1]; here it serves as architecture-level evidence on what an independent verifier is worth. Fixtures carry a planted Boolean ground truth valid and a proposal produced by a noisy process that flips the truth with probability $p_{\text{prop}} = 0.18$; no language model is involved. Five fixture classes are generated (arithmetic predicates, citation validity, judgment questions, completion state, cross-matter authorization), 240 per class per seed, over 30 seeds. Judgment fixtures are always escalated and never auto-accepted; the remaining four classes are decidable and are the object of measurement.

The verifier is fallible in two ways that correspond to the two conditions of conditional soundness. First, a share c of the proposal's errors are *shared-source* errors: they originate in an authority the verifier reads too (a stale edition, a mis-encoded rule), so the verifier inherits them. Second, the verifier makes its own rule-encoding errors with probability p_{rule}, independently of the proposal. The guarded mode accepts the verifier's verdict; the unguarded mode accepts the proposal as it is. Since half the fixtures are invalid, the expected false-accept rates on decidable fixtures are

$$\text{FA}_{\text{unguarded}} = \tfrac{1}{2}\, p_{\text{prop}}, \qquad \text{FA}_{\text{guarded}}(c, p_{\text{rule}}) = \tfrac{1}{2}\big[p_{\text{prop}}\, c + (1 - p_{\text{prop}}\, c)\, p_{\text{rule}}\big]. \tag{14}$$

The corner $c = 0$, $p_{\text{rule}} = 0$ reproduces the withdrawn oracle result; the corner $c = 1$ reproduces the unguarded rate exactly, whatever the verifier's own quality. The term $p_{\text{prop}}c$ is a floor that no improvement of the verifier can lower, because it is not the verifier's error: it is the error of the source both components share. This is the same common-cause structure that section 6 describes in words when it says that agreement between a proposer and a critic drawn from the same source is not independent confirmation. The grid is $c \in \{0, 0.1, 0.3, 0.5, 0.75, 1\}$ and $p_{\text{rule}} \in \{0, 0.01, 0.05\}$; the script `experiments/legal_benchmark.py` and seed-level results accompany the package.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

9.3 Results

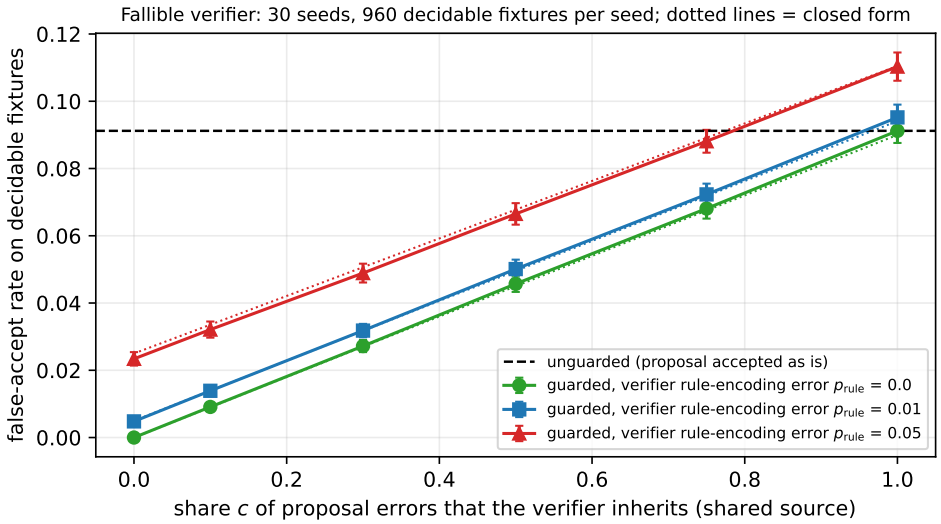

Figure 4: False-accept rate on decidable fixtures as a function of the shared-source share c, for three verifier rule-encoding error rates. Bars are two standard errors over 30 seeds; dotted lines are the closed form (14); the dashed line is the unguarded rate.

Mode	$(c,\ p_{\text{rule}})$	false accept	±SE	closed form (14)
unguarded	—	0.091	0.002	0.090
guarded (oracle corner)	(0, 0)	0.000	0.000	0.000
guarded	(0, 0.01)	0.005	0.000	0.005
guarded	(0, 0.05)	0.023	0.001	0.025
guarded	(0.3, 0.01)	0.032	0.001	0.032
guarded	(0.5, 0)	0.046	0.001	0.045
guarded	(1, 0)	0.091	0.002	0.090
guarded	(1, 0.05)	0.110	0.002	0.111

Table 2: Selected cells of the fallible-verifier experiment (30 seeds, 960 decidable fixtures per seed). False-reject rates are symmetric and omitted.

Figure 4 and table 2 show three things. The simulation follows the closed form to within sampling error, so the experiment is a check of the accounting, not a source of surprise. The value of a verifier is governed by c far more than by p_{rule}: a verifier that inherits half the proposer's sources ($c = 0.5$) with perfect rule encoding is worse than a verifier with independent sources and a 5% encoding error rate (0.046 against 0.023). And at $c = 1$ the verifier adds nothing and, with its own errors, subtracts: the guarded rate exceeds the unguarded one. The practical reading for the architecture is that source governance (section 5.8), which determines c, is a first-order control, and that a compiler whose rulebook is derived from the same retrieval that feeds the proposer verifies very little. This is a result about the control layer in a synthetic environment; it says nothing about the accuracy of any legal reasoning system.

10 Evaluation protocol for a genuine legal benchmark

The experiment above fixes the upstream error process and varies the verifier. The statistical question of when specialization beats pooling is studied in the companion preprint [2], whose synthetic experiments vary niche heterogeneity, labelled data and routing error. Neither experiment involves a language model or a legal

matter. The step from them to an empirical claim about legal AI requires the following protocol, which we state so that it can be pre-registered.

Materials. Cases are version-controlled matter bundles with authoritative source snapshots, factual evidence, jurisdiction, and a list of determinate rule prongs and judgment prongs. Qualified legal reviewers produce a source-linked reference annotation and record disagreements; adjudication is kept separate from model-generated recommendations. Splits are by matter, so that duplicate filings, templates or the same client never cross partitions. Source updates, missing evidence, jurisdiction switches and unauthorized tool requests appear as stress conditions.

Factorial design. Four systems run on identical held-out bundles with matched tool permissions, inference budgets and retrieval corpora: (A) a generalist with ordinary retrieval; (B) specialized but unverified agents; (C) a hybrid with shared reasoning and bounded specialists; (D) the same hybrid with the Legal Verifiability Compiler, the rule/judgment firewall and human escalation. The contrast B versus A isolates specialization; D versus C isolates independent verification; a better guard is not evidence that a smaller model learned the task better. The verifier's c is manipulated by giving the compiler's rulebook either an independent source snapshot or the same retrieval the proposer uses, so that eq. (14) can be tested on real outputs.

Outcomes. (a) correct completion of decidable prongs; (b) false acceptance of incorrect rule applications; (c) citation existence and temporal validity; (d) premature-closure rate; (e) judgment treated as determinate; (f) cross-matter exposure under adversarial prompting; (g) compute and human-review cost per completed, independently checked matter. Differences are reported with confidence intervals over matters, paired by matter where the same bundle is seen by all systems.

Hypothesis 2 (Specialization efficiency)**.** *On narrowly specified tasks, specialist or deterministic components can produce competitive or superior verified performance per unit of compute, relative to generalist-only execution (B versus A).*

Hypothesis 3 (Independent verification)**.** *A sound external compiler with an independent source snapshot reduces incorrect acceptance of decidable claims compared with generative self-evaluation (D versus C), by an amount that shrinks toward zero as the compiler's sources are made to coincide with the proposer's.*

Hypothesis 4 (Stateful completion)**.** *Explicit completion gates reduce the rate of premature finalization of long-horizon legal workflows.*

Hypothesis 5 (Controlled delegation)**.** *Capability-scoped orchestration reduces unauthorized actions and cross-matter disclosures compared with unrestricted tool access.*

Hypothesis 6 (Hybrid Pareto frontier)**.** *For mixed legal tasks, an ecology combining generalist interpretation and specialized verification can improve the performance–risk–cost Pareto frontier relative to pure generalist and unverified multi-agent baselines.*

No hypothesis is presented as experimentally proven. No new legal-case benchmark scores are reported in this article.

11 Discussion and limitations

The ecological framing is a framing device, not a classification of models by biological fitness; its mechanism and its limits are set out in [2]. Large models may be efficient on some tasks, specialists may require expensive integration, and tool orchestration itself introduces coordination overhead. The legal compiler verifies only faithfully formalized rules; mistakes in authority selection, source dating, factual extraction or exception modeling remain possible. Human approval can fail if reviewers see incomplete evidence or are prompted to rubber-stamp outputs. The architecture therefore requires independent source governance, meaningful review interfaces and external evaluation. The control-layer experiment of section 9 is synthetic: its proposal errors are coin flips, its verifier is a two-parameter abstraction, and its closed form holds because the fixtures are independent; it establishes the accounting, not the behaviour of any deployed system.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Aicumene:2026.049v1 [cs.AI] October 24, 2026

Our proposal should also not be confused with a proof of strong AI. Bernadotte and Gromov's dimensional, topological and dynamical criteria [3] concern a separate research question about intrinsic organizational properties of intelligence. Here those ideas motivate measurable study of system trajectories and monitoring coverage, not an assertion that orchestrated systems already meet the proposed criteria.

12 Conclusion

The capabilities of an artificial system are not exhausted by those of its largest model. Generality through scale and competence through specialization can coexist within a governed ecology of tools, sources, verification and human judgment. In legal AI, the decisive move is to separate a generated legal proposition from conditions under which it can be independently justified. The verifiable intelligence ecology is therefore a research program on architecture: how to allocate reasoning, formal checking, authority, state, privacy and risk across heterogeneous components. Its strongest claims remain falsifiable and await comparative evidence. The most capable artificial system may ultimately be distinguished not by model size alone, but by how well its components are organized, constrained and verified.

Scope and falsifiability. The ecological framing is useful only if it yields measurable comparisons. The cost crossover and verified-efficiency score are developed and tested in the companion preprint; the objects specific to this paper, verification coverage, the coverage gap, the shared-source relation (14) and the authority-boundary failure modes, can be tested on matched legal workloads with clearly defined authority sources and adversarial cases. A negative result, for example a generalist architecture outperforming the ecology after integration, verification and review costs are counted, would limit the hypothesis rather than invalidate the underlying importance of independent verification. We therefore treat ecological organization as a design proposition with empirical boundaries, not as a universal law of artificial intelligence.

Declarations

Author contributions. Both authors conceived the architecture and its ecological framing, developed the formal propositions, designed and ran the control-layer experiment, wrote the manuscript, and approved its final version.
Competing interests. The authors are co-founders of Aicumene and hold commercial interests in legal AI systems, including the BernioOS architecture discussed in this paper.
Funding. The research did not receive any external funding.
Data availability. Only synthetic fixtures generated algorithmically by `experiments/legal_benchmark.py` were used; no client matter, privileged file or lawyer-reviewed benchmark.
Code availability. The fallible-verifier experiment and its seed-level results are distributed with the LaTeX source of this preprint (`experiments/`); the pooling experiments are distributed with the companion r/K preprint. A comparative evaluation on actual model runs with qualified legal annotation remains future work.
Figures. All figures are vector drawings produced from the source of this article; no generative-AI imagery is used.
Version history. Two internal drafts (October 2026) circulated under the working number Aicumene:2026.101. This is version 1 of the public preprint, Aicumene:2026.049v1 (October 24, 2026); relative to the drafts it adds the cross-references to the tenth revision of the companion legal note, finalizes the declarations, and completes the bibliography with persistent identifiers.

References

[1] Bernadotte, A. and Menshikov, I. (2026). *Verifiable AI for Law: A Legal Verifiability Compiler with a Rule/Judgment Firewall.* Aicumene Research preprint 2026.040 (v1, June 2026; v10, October 2026, SSRN forthcoming). doi:10.13140/RG.2.2.18174.75841. `https://research.aicumene.com/papers/verifiable-law/`.

Aicumene:2026.049v1 [cs.AI] October 24, 2026

[2] Bernadotte, A. and Menshikov, I. (2026). *r/K Selection for Artificial Intelligence: Density-Dependent Selection, Allometric Scaling, and Falsifiable Tests of Architectural Strategies*. Aicumene r/K research preprint, v4 (October 2026), in preparation; companion to the present article.

[3] Bernadotte, A. and Gromov, V. A. (2025). *What is Strong AI? Intrinsic Structural Criteria for Strong Intelligence Based on Dimensionality and Topology*. Aicumene Research preprint 2025.043 (v1, August 2025; v2, October 2026, SSRN forthcoming). `https://research.aicumene.com/papers/StrongAI/`.

[4] Bernadotte, A. and Menshikov, I. (2026). *Multichannel Singular Spectrum Analysis: Counting Oscillatory Modes and Recovering Their Spatial Loadings*. Aicumene Research preprint 2026.044v2; SSRN working paper 7693202 (November 11, 2026). doi:10.2139/ssrn.7693202. `https://research.aicumene.com/papers/MSSA/`.

[5] Bernadotte, A. and Menshikov, I. (2026). *From Oscillatory Modes to Source Geometry: Identifiability and Localization in Multichannel Time Series*. Aicumene Research preprint 2026.058v1; SSRN working paper 7716058 (November 2026). doi:10.2139/ssrn.7716058. `https://research.aicumene.com/papers/source-geometry/`.

[6] Bernadotte, A. and Menshikov, I. (2026). *Brain Oscillator Intrinsic Dimension Marks Consciousness, Recovery Mode, and Failed Downshifting in Coma*. Aicumene Research preprint 2026.042v2; SSRN working paper 7667278 (October 24, 2026). `https://ssrn.com/abstract=7667278`, doi:10.2139/ssrn.7667278. `https://research.aicumene.com/papers/boid-coma/`.

[7] Bernadotte, A. et al. (2026). *VeNet: A Lightweight Neural Network for Efficient Brain Vessel Segmentation in Endovascular Robotic Surgery*. Scientific Reports. doi:10.1038/s41598-026-54176-x.

[8] Bernadotte, A., Elfimov, N. and Menshikov, I. (2026). Semi-automatic annotation of brain vessels in magnetic resonance angiography images. *Scientific Data* 13, 41. doi:10.1038/s41597-025-06354-1.

[9] Bernadotte, A. (2023). Real-time regular expression matching. arXiv:2308.10208. doi:10.48550/arXiv.2308.10208.

[10] Bernadotte, A. (2026). Counter-augmented deterministic automata for real-time regular expression matching without state explosion. *2026 12th International Conference on Automation, Robotics and Applications (ICARA)*, 471–475. doi:10.1109/ICARA69401.2026.11480280.

[11] Bernadotte, A. (2026). Topology-driven classification of time series. bioRxiv 2026.04.25.720787. doi:10.64898/2026.04.25.720787.

[12] Brown, T. B. et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* 33. doi:10.48550/arXiv.2005.14165. `https://arxiv.org/abs/2005.14165`.

[13] Meta AI (2024). Introducing Llama 3.1: Our Most Capable Models to Date. `https://ai.meta.com/blog/meta-llama-3-1/`.

[14] MacArthur, R. H. and Wilson, E. O. (1967). *The Theory of Island Biogeography*. Princeton University Press (Princeton Landmarks in Biology edition, 2001). doi:10.1515/9781400881376.

[15] National Institute of Standards and Technology (2024). *AI Risk Management Framework: Generative AI Profile*. NIST AI 600-1. doi:10.6028/NIST.AI.600-1.

[16] Zaharia, M., Khattab, O., Chen, L., Davis, J. Q., Miller, H., Potts, C., Zou, J., Carbin, M., Frankle, J., Rao, N. and Ghodsi, A. (2024). The Shift from Models to Compound AI Systems. Berkeley Artificial Intelligence Research blog, 18 February 2024. `https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/`.

[17] Dahl, M., Magesh, V., Suzgun, M. and Ho, D. E. (2024). Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. *Journal of Legal Analysis* 16(1), 64–93. doi:10.1093/jla/laae003.

[18] Magesh, V., Surani, F., Dahl, M., Suzgun, M., Manning, C. D. and Ho, D. E. (2025). Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. *Journal of Empirical Legal Studies* 22(2), 216–242. doi:10.1111/jels.12413. Preprint: `https://arxiv.org/abs/2405.20362`.

[19] Merigoux, D., Chataing, N. and Protzenko, J. (2021). Catala: A Programming Language for the Law. *Proceedings of the ACM on Programming Languages* 5(ICFP), article 77. doi:10.1145/3473582.

[20] Zheng, L. et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in*

Neural Information Processing Systems 36, 46595–46623. doi:10.52202/075280-2020. https://arxiv.org/abs/2306.05685.

[21] Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K. and Tabuada, P. (2019). Control Barrier Functions: Theory and Applications. *European Control Conference*, 3420–3431. doi:10.23919/ECC.2019.8796030.

[22] Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S. and Topcu, U. (2018). Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence* 32(1). doi:10.1609/aaai.v32i1.11797. https://arxiv.org/abs/1708.08611.

[23] Tabuada, P. (2009). *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer. doi:10.1007/978-1-4419-0224-5.

[24] Leucker, M. and Schallhart, C. (2009). A brief account of runtime verification. *Journal of Logic and Algebraic Programming* 78(5), 293–303. doi:10.1016/j.jlap.2008.08.004.

[25] Levina, E. and Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems* 17.

Aicumene:2026.049v1 [cs.AI] October 24, 2026