Verifiable AI for Law: a legal verifiability compiler with a rule/judgment firewall

The model proposes; a deterministic compiler disposes

Aicumene Preprint

Alexandra Bernadotte
AICumene*
bernadotte@aicumene.com

Ivan Menshikov
AICumene*

June 2026

AiCumene:2026.040v1 [cs.AI] June 2026

Abstract

Legal language-model agents may fabricate legal authorities, overstate discretionary conclusions, and prematurely present incomplete analyses as final. This AICumene research note introduces *Verifiable AI for Law*, a bounded decision-support architecture centered on an independent deterministic verifiability compiler. Proposer and Critic agents generate and challenge a typed *Application-IR*; the compiler checks citations, jurisdiction, source currency, and formally decidable predicates against approved rules and validated case-specific facts, producing auditable diagnostics. A *rule/judgment firewall* separates mechanically verifiable conditions from open-textured legal standards and escalates the latter to qualified human reviewers. Distinct authority and evidence retrieval paths retain provenance, while capability-scoped execution supports matter isolation. We describe the prototype, illustrative legal workflows, security design, and staged rollout. The system does not establish the legal correctness of its rulebook or replace professional legal judgment. Its verification guarantees are conditional on the validity of encoded rules and input evidence. The model proposes; the compiler disposes.

Keywords: verifiable AI; legal AI; deterministic compiler; rule/judgment firewall; Application-IR; legal retrieval; provenance; agent security; VeNet; LLM; BernioOS; Engineering Intelligence.
DOI: 10.13140/RG.2.2.18174.75841.
Online note: https://research.aicumene.com/papers/verifiable-law/.

1 Why raw LLM outputs are insufficient in law

Two recurrent failure modes matter: **fabrication / over-assertion** (invented citations, or discretionary standards such as "extreme hardship"[1] and "well-founded fear"[2] treated as mechanically settled) and **premature completion** (partial analyses presented as final). Merely scaling a model does not make its output

*Aicumene. https://aicumene.com. © 2026 Alexandra Bernadotte and Ivan Menshikov.

[1] *Extreme hardship* is a context-sensitive standard used in certain U.S. immigration waivers. It requires hardship to the statutorily qualifying person beyond the ordinary consequences of separation or relocation, assessed in the totality of relevant circumstances. The applicable statutory provision determines whose hardship matters. See USCIS Policy Manual, Vol. 9, Part B, https://www.uscis.gov/policy-manual/volume-9-part-b.

[2] *Well-founded fear of persecution* is an asylum-law standard requiring a reasonable possibility of future persecution on account of a protected ground, evaluated using the applicant's circumstances and supporting evidence; it does not require a greater-than-50% probability of persecution. See 8 CFR § 208.13, https://www.ecfr.gov/current/title-8/section-208.13.

independently auditable. The broader epistemic problem of establishing structurally measurable access to truth is also discussed in the April Theses of the New Cognitive Paradigm [1, 2]. A legal decision-support system must expose what can be verified, what cannot, and why.

2 Approach: one deterministic arbiter

The language model proposes a candidate analysis; a deterministic compiler checks the encoded predicates and controls release of conclusions. A bright-line *rule* is independently recomputed; an open-textured *judgment* is referred to a qualified human rather than automatically asserted.

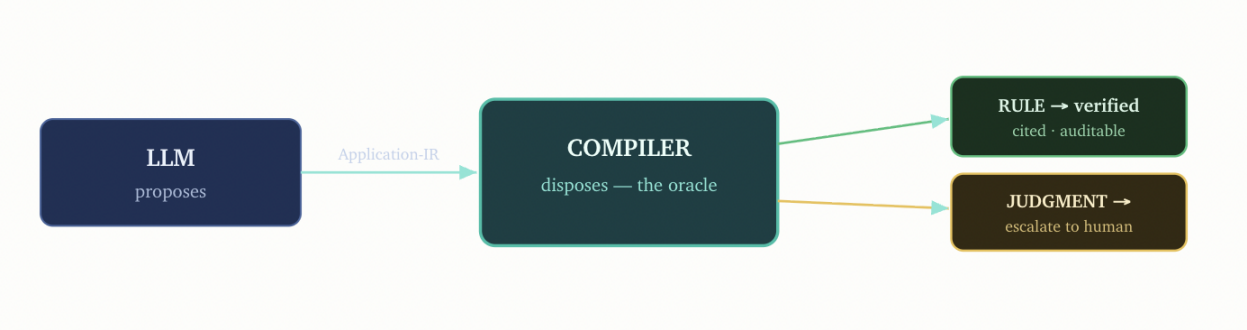


Figure 1: Approach: the compiler separates mechanically verifiable rules from judgments requiring human review.

3 Architecture: three axes around the arbiter

The first axis is the *verifiability compiler*; the second is *security and isolation*; the third is a *capability-scoped data plane* joining law, agency guidance, executable rules, and per-matter evidence. The verifier remains outside the model's action runtime. The surrounding question of how intelligence and knowledge-production systems are organized in a changing sociotechnical environment is examined at a broader level in the collective report *The Woven World* [2]. The present note addresses a narrower, implementable verification boundary.

4 Axis 1: the verifiability compiler

The compiler receives a typed `Application-IR` containing asserted facts, cited norms, requested application of law, and a candidate verdict. Its successive gates parse, resolve citations, establish jurisdiction, recompute thresholds, check whether sources are in force, and classify a proposition as a determinate rule or an evaluative judgment. Each gate emits

```
Diagnostic{code, severity, location, hint}.
```

An `error` blocks or repairs a deterministically incorrect rule claim; a `warning` triggers escalation for legal judgment. Executable rules carry source identifiers pointing back to their Akoma Ntoso texts. This architecture checks conformity with an approved rulebook; it does *not* prove that the rulebook itself is legally correct.

5 Axis 2: The rule/judgment firewall

For a determinate numerical test such as "at least 913 days present?", the compiler recomputes the result from validated facts and an approved rule. For an evaluative question such as "extreme hardship?", it declines to issue an autonomous substantive conclusion and escalates. The numerical threshold is an *illustrative encoded test*, not a jurisdiction-independent legal requirement.

AiCumene:2026.040v1 [cs.AI] June 2026

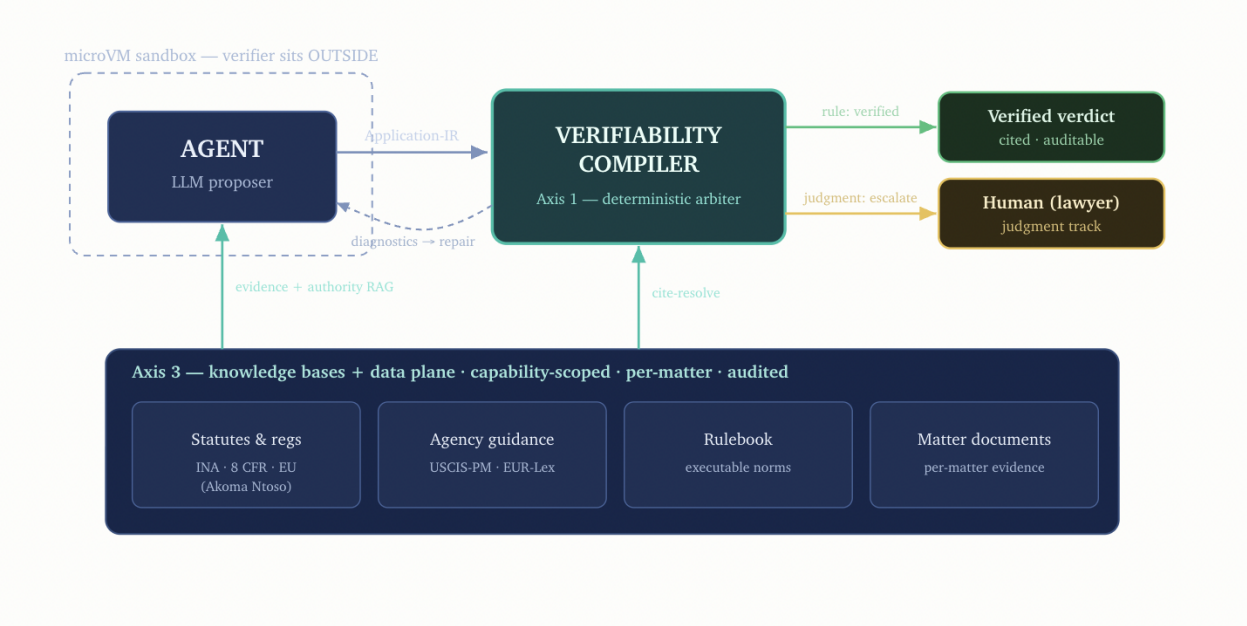


Figure 2: Architecture: three axes meet at an independent deterministic arbiter.

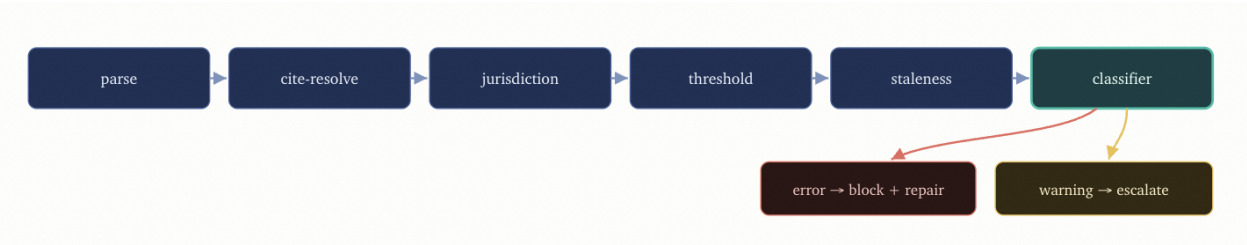


Figure 3: Compiler gates: each check produces a location-aware diagnostic and a controlled next step.

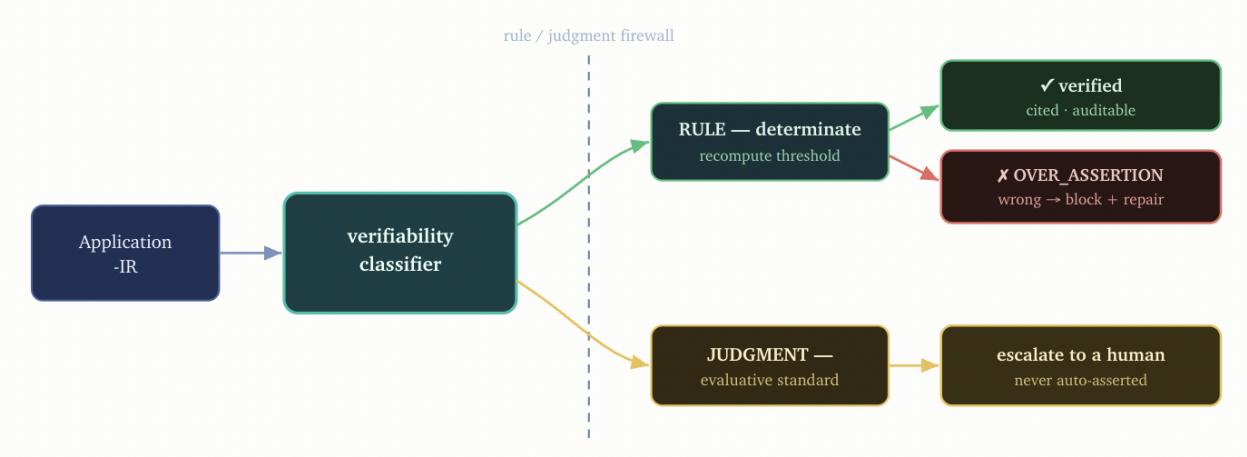


Figure 4: Rule/judgment firewall: different legal propositions receive different kinds of treatment.

AiCumene:2026.040v1 [cs.AI] June 2026

6 Knowledge bases and two retrieval paths

Four bases are logically segregated: *statutes and regulations* (e.g., U.S. INA and 8 CFR; relevant EU instruments), *agency guidance* (e.g., USCIS Policy Manual and EUR-Lex), an *executable rulebook*, and *per-matter client documents*. Authority RAG resolves the norm and its effective date; Evidence RAG establishes facts from the authorized matter. Retrieval uses capability-scoped access, logs requests and returned identifiers, and preserves provenance: every material fact maps to evidence; every rule maps to its source.

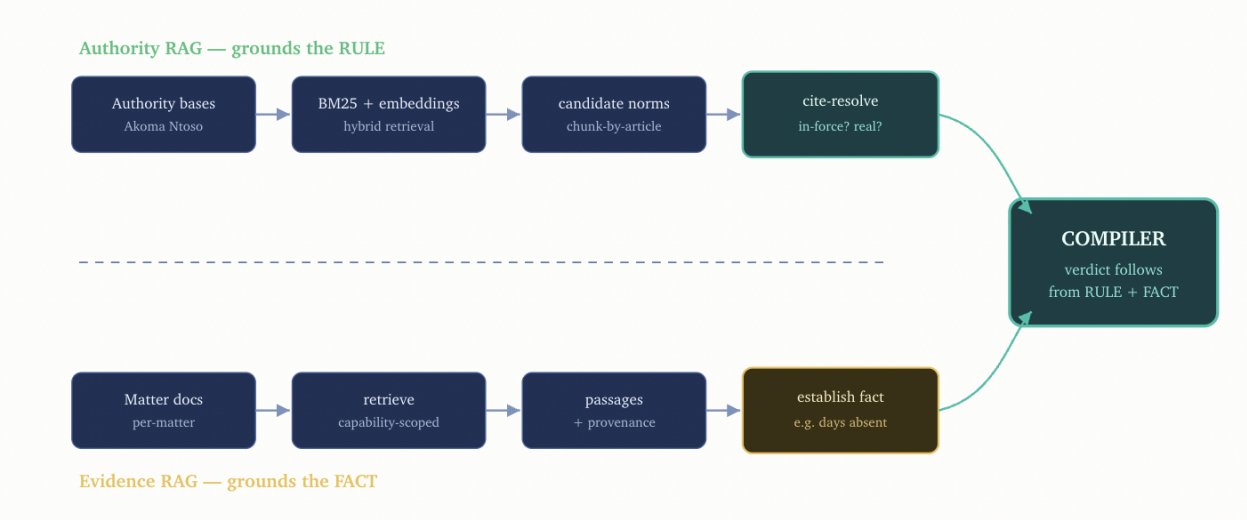


Figure 5: The two retrieval paths: authorities establish the applicable rule; documents support case-specific facts.

7 Three agent roles: a grounded dialectic

The **Proposer** (a Qwen-based agent in the described prototype) generates a structured candidate. The **Critic**, preferably from a different model family, attempts to refute assertions and expose gaps. The **Compiler** resolves mechanically verifiable issues against encoded sources rather than taking a vote among LLMs. Criticism can improve proposals but does not become legal authority.

Independent verification boundary. The Proposer and Critic can expose gaps and improve candidate analyses, but agreement between generative models is not an independent legal authority. The compiler instead recomputes formally decidable predicates using an approved executable rulebook and validated, provenance-linked facts. It emits traceable diagnostics, blocks unsupported mechanically checkable conclusions, and escalates questions requiring contextual legal evaluation to qualified counsel. This is a limited verification boundary, not an oracle of legal truth: correctness remains conditional on current and accurate legal sources, correctly encoded rules, and authentic, sufficiently complete evidence. The distinction between truth and evidence-mediated access to it is discussed more broadly in the April Theses [1].

8 A harness that improves without gaming the evaluator

The agent can propose revisions to its prompts, tools, or harness after inspecting failures; a held-out compiler-anchored evaluation gate accepts only validated improvements. The published note reports a critic trust measure increasing from 0.47 to 0.65 over training epochs. This is a *reported internal prototype indicator*, not a general accuracy result or a cross-system benchmark. Its validity depends on the coverage and correctness of the fixed compiler anchor.

AiCumene:2026.040v1 [cs.AI] June 2026

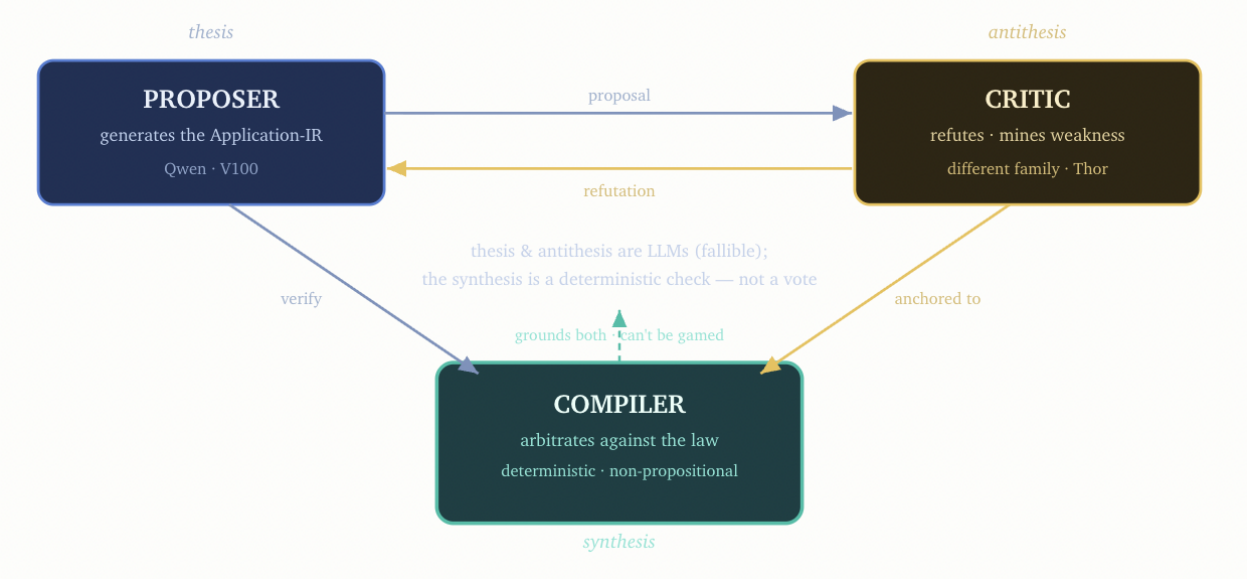


Figure 6: Proposer–Critic–Compiler. The proposer generates candidate Application-IR; the critic challenges its claims; the independent compiler verifies formally decidable conditions against approved rules and validated facts, or escalates questions requiring human legal judgment. The synthesis is a deterministic check, not a vote between language models.

9 Long-horizon matters: stopping is verified

Complex matters require gap finding, additional retrieval, and repeated testing of unmet preconditions. Technical completion occurs only when every deterministically encoded rule prong has been checked and every judgment prong has been referred for review. This guards against declaring success after an incomplete chain of analysis.

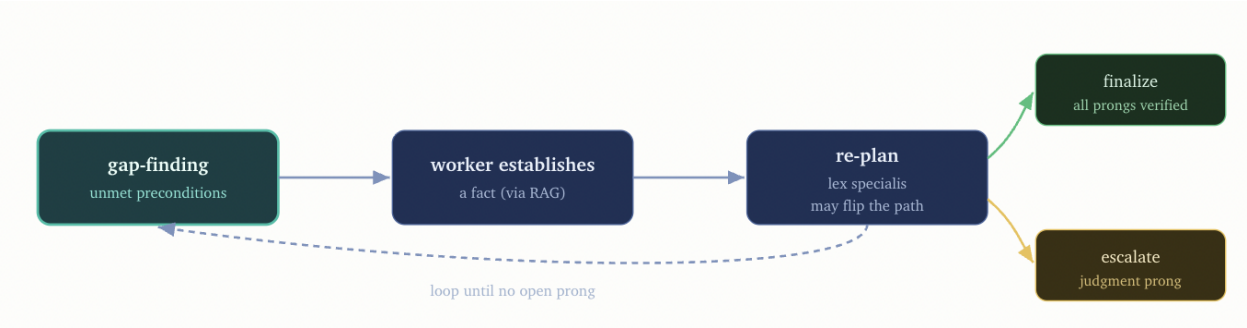


Figure 7: Long-horizon stopping: the kernel, not the model, determines whether a matter is technically complete.

10 Axis 3: isolation and security

The proposed perimeter combines LAN/mTLS, default-deny egress, secrets management, per-matter privileges, and hash-chained audit. The intended microVM progression is gVisor to Firecracker. The action runtime is sandboxed; the compiler is outside the isolation boundary. Full production microVM enforcement is described as a *planned stage*, distinct from existing per-matter segregation in code.

AiCumene:2026.040v1 [cs.AI] June 2026

11 Prototype demonstrations and rollout

The published project note reports end-to-end demonstrations on a Qwen model running on an NVIDIA V100. In an *long-term-residence* scenario, three client documents were ingested, a duration of "six years three months" was represented as 6.25 years, and the prototype reached a compiler-approved demo label `GRANTABLE`. In an *asylum* scenario, a mechanically checkable filing element was evaluated while the substantive "well-founded fear" question was escalated. These are demonstrations of workflow behavior, *not verified legal eligibility opinions*.

The development plan distinguishes a completed prototype core (P0) from planned authority retrieval (P1), microVM hardening (P2), human escalation interface (P3), matter-management integration (P4), and EU-first rulebook authoring (P5).

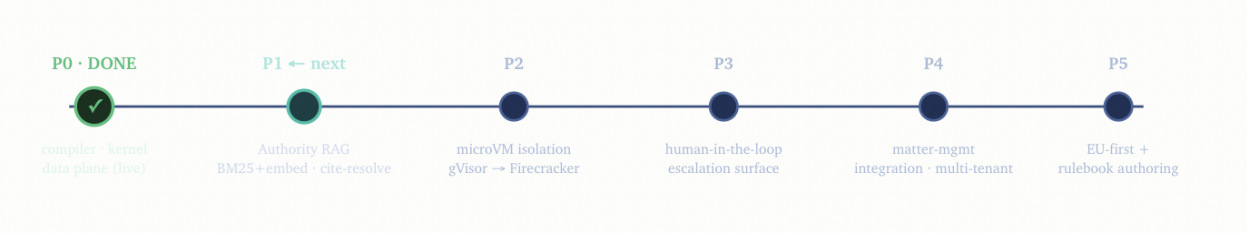


Figure 8: Architectural rollout: the note separates the currently described prototype from intended future components.

12 Scope and limitations

All legal thresholds, effective dates, and citations must be independently verified by qualified counsel before authoritative use. The compiler only decides what can be expressed as reliable executable predicates over validated facts; it cannot autonomously resolve open-textured legal standards or ensure an encoded rule is legally correct. Claims about self-improvement are epoch-local. The system is presented for *research and legal decision support, not autonomous legal advice.*

In one line. *The model proposes; the compiler disposes. Every machine-verified verdict should be checkable, cited, and reconstructable.*

13 Code Availability

The source code and implementation of the legal verifiability compiler, including the rule/judgment firewall, are available in the AICumene GitHub repository: `github.com/aicumene/verifiable-law`.

The repository contains the implementation and supporting materials for the Verifiable AI for Law research project.

14 Conclusion

Verifiable AI for Law presents an architecture for separating generative legal analysis from independently executable checks. The Proposer drafts a structured candidate, the Critic challenges its premises and supporting evidence, and the compiler determines whether formally decidable conditions satisfy approved rules and validated facts. This arrangement does not treat agreement between language models as legal verification.

The note makes three connected architectural contributions. First, a typed *Application-IR* makes assertions, legal authorities, factual predicates, and proposed outcomes explicit. Second, an independent *verifiability compiler* checks encoded conditions and produces reproducible, source-linked diagnostics. Third, the *rule/judgment firewall* prevents evaluative standards, including "extreme hardship" and "well-founded fear", from being mistaken for mechanically decidable rules; these questions are routed to qualified human

AiCumene:2026.040v1 [cs.AI] June 2026

reviewers. Together with segregated authority and evidence retrieval and capability-scoped execution, these components support auditable legal workflows and help prevent premature completion.

The epistemic motivation is consonant with the discussion of evidence-based access to truth in the April Theses [1]. At a different technical level, VeNet [3], together with the associated annotated MRA dataset [4], and harness infrastructure [7, 5, 6, 8, 9] provides examples of explicitly structured computation in a compact neural model; this is an architectural analogy, not a validation of legal reasoning. The broader sociotechnical context is examined by Ageev, Gromov, Pereslegins, Lukovnikova, and colleagues [2].

The compiler is not a source of legal authority: its results depend on correct and current rule encoding, reliable retrieval, authentic facts, and sufficient evidence. Open-textured legal interpretation and consequential decisions still require qualified human review. Prototype demonstrations illustrate workflow behavior rather than independently established legal eligibility or generalizable accuracy. Further work should validate rulebooks with counsel, benchmark across jurisdictions, harden isolation and provenance controls, and evaluate human escalation procedures.

The goal is a shift from unqualified AI-generated answers toward *verifiable legal computation within explicit limits. The model proposes; the compiler disposes.*

References

[1] Bernadotte, A. (2026). *The April Theses of the New Cognitive Paradigm: Measurability of Intelligence, Language, and Probabilistic Truth. Toward Structural Diagnostics of Consciousness, Strong AI, and Creativity. AICumene* preprint, AiCumene:2026.046v1. doi:10.13140/RG.2.2.20527.50087.

[2] Ageev, A. I., Gromov, V. A., Lukovnikova, N. M., Pereslegin, S. B., Pereslegina, E. B., Shilov, S. Yu., and Yutanov, N. Yu. (forthcoming 2027). *The Woven World: Complexity Crisis and Global Futures*. World-Systems Evolution and Global Futures. Cham: Springer. ISBN 978-3-032-36259-9 (hardcover; publisher's announced release: 11 February 2027). `https://link.springer.com/book/9783032362599`.

[3] Bernadotte, A. (2026). VeNet: a lightweight neural network for efficient brain vessel segmentation in endovascular robotic surgery. *Scientific Reports*, **16**, 24566. doi:10.1038/s41598-026-54176-x.

[4] Bernadotte, A., Elfimov, N., and Menshikov, I. (2025). Semi-automatic annotation of brain vessels in magnetic resonance angiography images. *Scientific Data*, **13**, 41. doi:10.1038/s41597-025-06354-1.

[5] 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)*, IEEE, pp. 471–475. doi:10.1109/ICARA69401.2026.11480280.

[6] Menshikov, I., Elfimov, N., Bernadotte, A. (2026). A Lightweight Hankel-Embedded Pipeline for Real-Time EEG Filtering and Classification. *2026 12th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE. `https://doi.org/10.1109/ICARA69401.2026.11480292`

[7] Bernadotte, A. (2023). Cyber Security for Surgical Remote Intelligent Robotic Systems. *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE, pp. 65–69. doi:10.1109/ICARA56516.2023.10126050.

[8] Menshikov, I., Orlov, K., Berestov, V., and Bernadotte, A. (2023). Preoperative Personalized Vascular 3D Simulator of the Intelligent Robotic Ecosystem LevshAI for Endovascular Neurosurgery. *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE, pp. 6–10. doi:ICARA56516.2023.10125981

[9] Bernadotte, A. (2022). The Algorithm That Maximizes the Accuracy of k-Classification on the Set of Representatives of the k Equivalence Classes. *Mathematics*, **10**(15), 2810. doi:10.3390/math10152810


```
@misc{bernadotte2026verifiablelaw,
 title={Verifiable AI for Law: a legal verifiability
 compiler with a rule/judgment firewall},
 author={Bernadotte, Alexandra and Menshikov, Ivan},
 year={2026}, version={AiCumene:2026.040v1},
 howpublished={AICumene Research Note},
 doi={10.13140/RG.2.2.18174.75841},
 url={https://research.aicumene.com/papers/verifiable-law/}
}
```

AiCumene:2026.040v1 [cs.AI] June 2026