DETERMINISTIC LOGICAL CORE ARCHITECTURE (DLCA)

Decoupling Deterministic Logic from Probabilistic Language Interface

Daniel Thomas Price

Abstract

Short Abstract

Current Large Language Models(LLMs) combine probabilistic generation with limited explicit reasoning capacity, producing systems that are fluent but impossible to audit. The Deterministic Logical Core Architecture (DLCA) proposes deterministic reasoning processes which can be separated by Human-In-The-Loop (HITL) processes from necessary probabilistic processes. A Pre-Processing Unit (PPU) refines propositions to rectify language, cultural and logical factors before the Logical Core (LC) mathematical expressions evaluate the proposition. A probabilistic Social Interface(SI) following the LC is restricted to the role of governed translator producing human digestible text from formalized LC output. An Audit File(AF) records meta data describing every event from raw input to rendered output. LC content is intended to be derived from expert knowledge and foundational documents using public domain sources. Using meta data, the logic will thereby be traceable to foundational documents or experts. A core architectural proposition, which is deterministic symbolic evaluation is more efficient, has been empirically demonstrated in peer-reviewed literature. DLCA is realisable in hardware or software. Hardware implementation provides stronger immutability guarantees and software implementation is a deployable first stage or within an LLM. The architecture is intended to facilitate precision, reproducibility, auditability, efficiency, cultural specificity, and HITL control.

Extended Abstract

Current Large Language Models (LLMs) are unreliable for high-stakes reasoning tasks because logical inference and probabilistic word prediction occur within a single opaque process that cannot be independently audited. The Deterministic Logical Core Architecture (DLCA) proposes deterministic reasoning processes to be separated by Human-In-The-Loop (HITL) approval from necessary probabilistic logical processes. An immutable Logical Core (LC) evaluates propositions mathematically. A probabilistic Pre-Processing Unit (PPU) translates common language to technical language processable by the LC. A post processing unit, Social Interface (SI), translates the technical output(s) to commonly understood language. HITL gate keeping can be configured at the output of the PPU, input to SI and output from SI. An Audit File (AF) records every data event from raw input to rendered output with a tracking code traceable to the foundational documents. Logical Core content is intended to be derived from foundational public domain sources or from experts. The lemmatic structure and logical relationships used in the

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

PPU and LC can be published for scholarly review and public access. The core architectural proposition has been empirically demonstrated [1] indicating deterministic symbolic evaluation measurably outperforms language-model-only reasoning.

Six expected benefits motivate the design, each is a hypothesis to be validated: *precision* through a process developing mathematical relationships for extracted lemmas and logical relationships from foundational documents or expert knowledge, *reproducibility* through output dependent on mathematical relationships, *auditability* through a proof-path traceable to foundational documents or experts, *efficiency* through pre-determined mathematical relationships displacing chain-of-thought token generation; *cultural specificity* through peer reviewed cultural translation module and HITL control to assure process fidelity at logical boundaries where uncertainties exist.

DLCA represents the convergence of five prior research efforts that have each independently sought similar structural improvements: a) neuro-symbolic AI, b) expert systems, c) runtime assurance, d) probabilistic logic programming, and e) memory-augmented neural networks. DLCA LC development will be facilitated by building from these prior efforts because the intellectual foundation is a valuable starting point. DLCA extends to physical and robotic control systems where real-time deterministic processing is a safety requirement.

Keywords: Deterministic Logical Core Architecture; DLCA; AI; NLP; neuro-symbolic AI; deterministic reasoning; auditability; formal verification; human-in-the-loop.

1. Introduction

1.1 Epistemic Fragility and Systemic Cost Misallocation in Monolithic LLMs

Transformer-based Large Language Models have achieved remarkable linguistic fluency while exhibiting persistent epistemic fragility. The monolithic architecture performs three fundamentally uncertain cognitive operations: a) probabilistic logical inference from invariant axioms, b) factual recall of mutable world-knowledge, and c) stylistic language generation where truth is not measured. The consequence is a class of systems that can struggle to reliably distinguish truth from plausible-seeming utterances. A failure mode empirically documented as hallucination [2] [3] [4].

Numerous mitigations exist: a) retrieval-augmented generation [5], b) chain-of-thought prompting [6], and c) constitutional AI [7] among them. However, each operates within a probabilistic framework, therefore cannot eliminate stochastic inference error by construction. Beyond hallucination, monolithic LLMs can exhibit catastrophic forgetting [8]. LLM's provide no transparent auditable reasoning trace [9] [10]. LLM's can be functionally altered by prompt injection [11] or weight updates [12].

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

The economic consequences are equally significant. AI hallucination has been systematically documented as a structural reliability failure across deployed NLP systems [4], and enterprise AI deployment has been shown to generate value reliably only when governance and oversight mechanisms are embedded at the architectural level [13] [14]. Chain-of-thought (CoT) reasoning generates extensive intermediate token sequences whose cost grows substantially with sequence length [15].

Agentic and reasoning-specialised models generate substantially more tokens per task than standard deployments [15]. Regulatory compliance costs compound this further: deploying a high-risk AI system under the EU Artificial Intelligence Act incurs substantial initial and ongoing compliance costs [15]. Systems that cannot demonstrate deterministic auditability by design cannot satisfy the GDPR right to explanation [16] except through innovative methods such as counterfactual proxies rather that explain away the concern than expose the underlying reasoning [17].

It has been argued waiting for LLM's to fully develop and then develop guidelines for development is not the most desirable approach [18]. Proposing to resolve some of the percieved problems with LLM development by considering alternative architectures such as DLCA which require alternative governance protocols at the early stage of development is therefore timely. DLCA is being proposed at a time when LLM's are undergoing rapid development and problems are starting to be understood more generally. DLCA proposal theoretically illustrates the power of architectural choices to shape the quality, reproducibility, consistency, costs and immutability of Artificial Intelligence (AI) systems.

The five prior-art trajectories address a subset of the identified gaps: a) probabilistic-only reasoning, b) absence of hardware-governed immutability, c) no operator-supervised input fidelity protocol, d) no structured audit trail linking every output to a formal proof-path. Section 4 presents the three module DLCA architecture as a response to this combination of gaps.

These structural deficiencies motivate an architectural separation of audit-able deterministic reasoning from probabilistic language generation with HITL validation of probabilistic function.

1.2 Contributions of This Work

The Deterministic Logical Core Architecture (DLCA) proposes deterministic reasoning processes be separated by human-in-the-loop approval from probabilistic logical processes to improve process fidelity by performing diambiguation. The DLCA architectural option does not seek to minimize the considerable capabilities of current LLM processes, however, seeks to extend LLM capability to high value applications where uncertainties have higher impact. Hardware implementation (ASIC/FPGA/OTP eFuse) is intended to provide stronger immutability than software-only implementations. Software implementation (SAT-solver library + frozen lemma database) is a valid first stage and is implementable in a LLM.

The software configuration requires no specialised hardware and is achievable with open-source tools (Z3 [19], MiniSAT [20], Protégé [21]) on standard infrastructure.

None of the six prior research trajectories (neuro-symbolic AI, expert systems, formal verification, runtime assurance, probabilistic logic programming, and memory-augmented neural

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

networks) has independently sought the same structural separation. DLCA is an evolution of these prior research trajectories. This evolutionary positioning can reduce adoption barriers. Every research trajectory listed provides an intellectual foundation for DLCA's adoption, validation, and future development.

2. DLCA Lemmatic Foundations

DLCA lemmatic foundations are presented partly to explain how intrinsic design creates the claimed features. In addition the lemmatic foundational approach can be valuable during testing and optimization by allowing a systematic response if economic or other measurements indicate deficiencies.

2.1 Rationale

Architectural constraints, hardware choices, testing criteria, and marketplace definitions in this paper are derivable from the DLCA lemma set. Fourteen DLCA lemmas are presented in Table 2.1. L1–L6 form the core architectural foundation of: precision, reproducibility, audibility, efficiency, cultural specificity and HITL control. While DCLA proposes use of all the lemma sets into one system, these lemmsas can be applied independently to achieve different objectives.

L14 is formulated to allow DLCA and LLM's to jointly execute a single task by employing a zone classification plan. L13 is derived for real-time physical control, showing how DLCA philosophy extends to existing control plans. Each lemma maps to architectural sections and hardware elements.

See table 2.1 for a detailed explanation of the lemmatic foundations and how these relate to this paper organization. Figure 2.1 diagrams the primary relationships between these lemmas.

Lemma	Statement	Maps To
L1 - Fidelity with foundational documents and expert reasoning.	Derived from logical fidelity feature. LC, Language Translator and Cultural Translator are derived from Public Documents or experts which can be peer reviewed or published in standards	Section 4
L2 – Logical is reproducible from public sources.	Derived from logic reproducibility feature. Logical relationships are evaluated consistently when using the same input language set.	Section 4
L3 – Audit-able since logical source is identified in AF file.	Derived from auditability feature. Using meta data, logical operations are traceable back to the foundational documents and phrases or experts that prompted the logical relationship.	Section 4
L4 – LC logic is represented by mathematical equations.	Mathematical relationships results in logic computational efficiency. Logic is the same each time and constructed in NLP, boolean, or SAT format.	Section 4
L5 – Cultural and language diversity accommodated through design.	Derived from cultural specificity. Language and cultural variables are consistently evaluated by matching the language and cultural units with unique culturally defined LC for each topic being evaluated.	Section 4
L6 – Human approval of inputs or outputs of	Human in the Loop (HITL) derived from need for one or more levels of human approval. HITL is used to first verify the probabilistic process	Section 5

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Lemma	Statement	Maps To
probabilistic processes.	of prompt classification, language translation and cultural translation are sufficient for LC evaluation. Then used to verify the outputs are as expected from the LC and the probabilistic SI unit.	
L7 – Hardware implementation	Derived from L1 and L3. By formulating logic from foundational documents or experts in public view the logic can be reasonably assured of stability. Stable logic will enable hardware etching. Etching will improve operations to facilitate achieving: L2 and L4.	Section 7
L8 – Cultural and Language canonical propositions generated.	Derived from L1 and L5. The combination of peer review and matching logical engines to the language and cultural frameworks precise propositions are generated for HITL then LC evaluation.	Section 8
L9 – LC requires unique and specific propositions.	Derived from L6 and L8. The LC output is only valid as the inputs represent the query to be asked. Cultural, language and interpretive uncertainties must therefore be minimized.	Section 7
L10 – SI is LLM interpreting LC outputs.	Derived from L9. Si produces human digestible output by combining LC output with user inputs that can only be evaluated by a probabilistic core.	Section 9
L11 – Epistemic fidelity of LC outputs	Derived from L1, L2. and L5. Matching LC with PPU for each ontology will result in the same results from the LC for the same user input.	Section 7
L12 – Measurement characterization of SI drift. OPM	Derived from L9 and L10. SI uses LLM processes to develop human digestible text. Developing a measurement of the change from LC output to SI output can cause a human response.	Section 9, 10
L13 – Marketplace viability of logical processes.	Derived from L1, L2, and L3. The architecture allows language unit, cultural unit, or LC to be provided by different entities.	Section 13
L14 - Physical control real-time determinism	Derived from L4 and L7. The architecture allows for fast deterministic logical operations when the processing speed must be faster than software only processing will allow.	Section 15
L15 – DLCA can be implemented in LLM	Derived from L11. With LC located within LLM the LC outputs can be included in LLM output as demonstrated by [1].	Section 9
L16 – Some LLM user inputs can be evaluated only with probabilistic processes. QCM	Derived from L15. Robustness of system to handle all types of queries. Some instances will not require the user to have intimate knowledge of tool limitations therefore will need processes to sort out and process propositions using different tools.	Section 6.3
L17 – Proposition zone classification	Derived from L16. Tasks receive a zone classification (Zone 1, 2, or 3) derived from L1 and L5. Zone 1: all premises unambiguous, proof-path closes. Zone 2: task decomposes into deterministic and probabilistic sub-tasks requiring multi-pass routing. Zone 3: task is solely a probabilistic task without LC involvement.	Section 6.3

Table 2.1 — DLCA Lemmatic Foundation L1–L14. Each lemma maps to architectural sections, hardware elements, and testing constructs for primary features.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

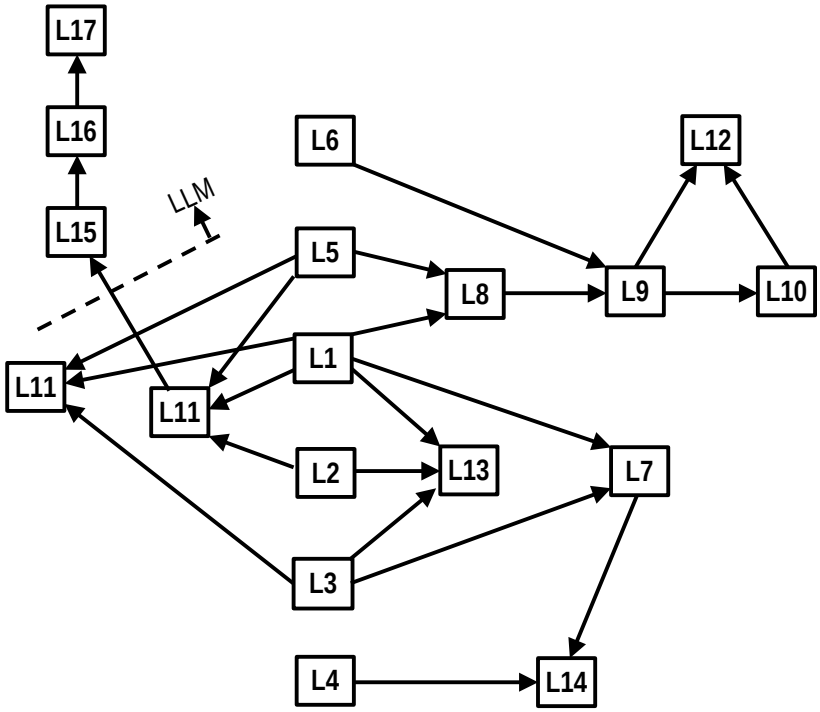

Figure 2.1 — Lemmatic Derivation Structure: L1–L17 dependency graph showing which lemmas derive from others

3. Prior Art and Identified Gaps

The prior art analysis in this section is based on a structured search conducted across IEEE Xplore, ACM Digital Library, arXiv, Semantic Scholar, and Google Scholar over the review window through 2026, using the following query families: (i) neuro-symbolic AI, hybrid reasoning architecture, symbolic-neural integration; (ii) expert system, production rule engine, knowledge engineering; (iii) formal verification, runtime verification, model checking, theorem prover; (iv) explainable AI, interpretable machine learning, post-hoc explanation; (v) AI governance, AI audit trail, transparent AI, trustworthy AI; (vi) safety-critical control, runtime assurance, functional safety, deterministic control; (vii) dual-engine architecture, dual cognitive architecture, modular AI; (viii) probabilistic logic programming, neuro-probabilistic; (ix) memory-augmented neural network.

Searches were supplemented by forward and backward citation tracing from the most-cited works in each category and by review of IEICE Transactions, IEEE Transactions on Neural Networks and Learning Systems, ACM TOSEM, and ICATD proceedings.

All "no identified prior work" claims in this section refer to the body of work identified through the search process and review window described above. The authors acknowledge that the field is

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

large and that works not identified through this methodology may exist. Claims are stated as "no identified prior work" or "no identified prior system" rather than absolute assertions. Claims are scoped to the specific conjunction of properties described, not to any individual property in isolation. Neuro-Symbolic AI research [22] has sought to integrate symbolic reasoning with neural language processing. However, we identify no production hardware realisation of these approaches in the reviewed literature.

A related sub-field is Probabilistic Logic Programming, which embeds neural networks inside a probabilistic logic programming framework. Probabilistic Logic Programming is exemplified by DeepProbLog [22] [23] and provides learnable rule weights over logical predicates. These systems represent the closest software analogue to the LC's proposition evaluation function. The distinction is structural: a) probabilistic logic engines have no hardware immutability. b) no operator-supervised input approval gate, c) no natural-language rendering layer. d) no cultural vocabulary separation e) no audibility.

DLCA's proposition evaluation does not produce a probability distribution over conclusions but a verifiable proof-path with a unique outcome under a fixed proposition set. These unique outcomes can have conflicting or different conclusions especially when combining results derived from foundational documents or expert system logical relationships. A full formal comparison of DLCA's proposition evaluation function against probabilistic logic programming semantics, including DeepProbLog [23] and Markov Logic Networks [24], is designated as a first companion paper deliverable. Memory-augmented neural networks, such as the Differentiable Neural Computer [25], address the related problems of long-term information retention and reinforcement learning by coupling a neural network to an external differentiable memory. These architectures improve recall and structured data manipulation but remain fully probabilistic. Outputs carry no auditable path.

Retrieval-Augmented Generation (RAG) [5] externalizes factual knowledge but leaves reasoning probabilistic. Mixture-of-Experts (MoE) architectures reviewed are probabalistic [25]. MoE architecture introduce modularity at the layer level but do not separate reasoning logic from linguistic interface. Low-Rank Adaptation (LoRA) [27] permits parameter-efficient fine-tuning with a frozen base, but the base itself remains a probabilistic weight matrix. To the best of the authors' knowledge, based on the structured search, no prior system identified in the surveyed literature combines precision, reproducibility, auditability, efficiency, cultural specificity and HITL simultaneously. This five-dimensional gap constitutes the principal contribution of DLCA.

Pan et al. [1] demonstrate empirically that integrating LLMs with deterministic symbolic solvers (SAT solver) produces accuracy improvements of 39.2% over standard LLM prompting and 18.4% over chain-of-thought prompting Improvements of 24.98% and 10.44% respectively are observed even for GPT-4. These results establish that offloading deterministic inference from the LLM to a symbolic engine measurably reduces output error rates across model scales and task domains, providing empirical grounding for key parts of the DLCA architectural proposition. Logic-LM [1] is accordingly the closest existing peer-reviewed empirical analogue to the DLCA core. DLCA extends this approach through hardware commitment of the symbolic component, a persistent audit chain, cultural vocabulary routing, and governed HITL oversight — mechanisms absent from Logic-LM's scope. Compantion paper testing is planned to determine whether

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

DLCA's additional properties produce further measurable improvement above this established baseline [1].

3.1 Rule-Based Expert Systems

DLCA bears resemblance to classical expert systems: MYCIN [28], R1/XCON [29], and the broader expert-system shell tradition surveyed in [30]. Experts can be a recognized source of valid interpretation and has been proposed in social sciences as an alternative logical methodology to logical extraction from research articles. DCLA architecture recognizes use of experts or expert systems logical depictions as a qualified source of truth just so long as the audit-ability, fidelity and precision can be maintained.

There are structural distinctions to note however:

(1) Classic expert systems have no natural-language input pipeline. Rules must be manually encoded by knowledge engineers, The translation from user intent to formal proposition is entirely absent. DLCA's PPU provides a query →language →cultural → proposition translation layer that expert systems entirely lack.

(2) Classic expert systems produce raw rule-engine outputs with no probabilistic rendering layer. DLCA's SI provides governed output whose departure from the canonical conclusion is quantified by the Output Precision Metric (OPM).

(3) Classic expert systems reside entirely in software and can be modified at any time. DLCA provides a pathway to physical immutability through silicon encoding.

(4) Classic expert systems embed a single cultural and linguistic ontology implicitly. DLCA vocabulary and cultural families provide explicit, switchable, formal pluralism. DLCA is therefore not an expert system but a principled successor architecture that addresses structural limitations that expert systems have not been shown to overcome [28] [29] [30] [31].

(5) DLCA expert system logic is anticipated to be derived in a public forum traceable to the foundational documents or experts governing the logical formulations. The public forum expectation does not rule out proprietary systems.

Modern production rule engines including CLIPS [32] (developed at NASA Johnson Space Center, 1985, and maintained in the public domain through release 6.4, 2023) and Drools [32] (Red Hat/JBoss, 2001, the dominant enterprise rule engine) extend the expert-system tradition to large-scale deployments. They remain subject to the same structural limitations: no governed natural-language input pipeline, no probabilistic output rendering layer, no hardware immutability, no cultural pluralism architecture and no language translation unit.

3.2 Formal Verification and Runtime Assurance Systems

DLCA's claims regarding proof-paths, deterministic verification, and safety-critical suitability will be evaluated against the formal methods tradition: model checking [33], theorem provers (ACL2 [34], Coq [35], Isabelle/HOL [36] , TLA+ [37], SPIN [38]) , and runtime verification [39]. The key distinction is that formal verification tools operate on manually specified formal models. DLCA's distinguishing contribution is the natural-language-to-formal-proposition translation pipeline that theorem provers assume away. An expert using Coq or Z3 must first construct the formal model

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

entirely in expert notation. DLCA's PPU can be configured to translate natural languge into formal model language. Table 3.1 allow comparison of these findings.

DLCA does not replace formal verification tools, but it extends formal verification to settings where natural language input is unavoidable and where complete model specification by a domain logic expert is impractical.

Dimension	Neuro-Sym.	RAG	MoE	LoRA	Expert Sys.	Agentic Fw.	DLCA
Reasoning locus	Hybrid SW	LLM	LLM	LLM	Rule Engine	LLM	Bounded LC
Immutability	None	None	None	Partial	None	None	Full (L5)
Auditability	Partial	Source	Partial (gating inspectable)	None	Partial	None	Full AF-01–06
Cultural pluralism	None	None	None	None	None	None	7 families (L12)
Hardware grounding	None	None	None	None	None	None	ASIC/ FPGA (L5)
NL input pipeline	None	None	None	None	None	Probabilistic	HITL

Table 3.1 — DLCA vs prior art: six-system comparison across six dimensions.

3.3 Explainable AI

LIME [40] and SHAP [41] explain the behaviour of a probabilistic model by approximating local feature importance. Lime and SHAP operate without a canonical ground truth, producing approximated importance scores.

Counterfactual explanation methods [17], concept bottleneck models [42], and mechanistic interpretability approaches [43] similarly explain a probabilistic model's outputs after the fact.

The counterfactual approach was proposed specifically as a response to the legal and technical barriers that prevent black-box systems from satisfying the GDPR Article 22 right to explanation [16] [44]. Because probabilistic models cannot expose their internal reasoning, Wachter et al. [17] propose counterfactual explanations as an efficient means to counteract challenges. The Counterfactual explanations show what inputs would need to change to produce a different decision, without opening the model.

Hybrid deterministic-probabilistic system like LIME and SHAP are not deterministic because they cannot distinguish between the deterministic canonical component and the probabilistic rendering component.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

3.4 Agentic Frameworks

Agentic orchestration frameworks including AutoGen [45], and similar multi-agent systems [46] [47] attempt modular reasoning pipelines and multi-agent task decomposition. These frameworks improve task routing and tool use but leave all reasoning probabilistic and non-auditable. In agentic orchestration frameworks every agent node is an LLM call whose output cannot be formally verified against a canonical ground truth. The need for AI systems whose objectives and constraints remain robust to such pressures has been argued more broadly in the AI-control literature [48].

A related alignment approach, Constitutional AI [7], imposes content and reasoning constraints at the Reinforced Learning from Human Feedback training level through a small set of explicit natural-language principles. Constitutional AI improves harmlessness and transparency of LLM outputs but remains entirely probabilistic.

Agentic systems and other tools may be utilized in the work to construct the mathematical relationships needed by the DLCA LC. For logical relationships derived from foundational documents, a configured pipeline can parallelise literature synthesis, extract lemma candidates, run SAT-solver consistency checks [19], attach meta data, and produce structured output packages for HITL review prior to publication. For logic derived from expert reasoning agents can extract lemma candidates from verbal or text resources, develop logical relationships, attach meta data, and produce structured packages for HITL review. The goal being substantially lower per-lemma cost than human-only authoring. Tools used to construct LC content may have utility in other human facing work by organizing content from therapy sessions or job interviews into logical formats suitable for human consumption.

3.5 Clinical NLP Systems

Existing clinical NLP platforms, John Snow Labs Spark NLP Healthcare [49], commercial clinical NLP pipelines [50], and AI-powered point-of-care platforms such as OpenEvidence [51], may represent the most developed prior art in domain-specific language processing. These systems perform entity extraction, relationship identification, and clinical concept mapping from natural language. However these tools all do so using probabalistic approaches.

OpenEvidence [92] is a physician-facing clinical knowledge platform synthesising peer-reviewed literature for point-of-care queries. Peer-reviewed evaluation has confirmed capability alongside identifying limitations in citation accuracy and domain coverage [51]. Commercial clinical NLP platforms are deployed at scale for entity extraction and clinical concept mapping. No independent peer-reviewed evaluations of the specific APIs were identified in the literature surveyed for this paper.

However, no existing clinical NLP system (a) formally specifies the language and cultural output as to rectify differences between inputs, (b) subjects that output to iterative operator approval before reasoning begins, or (c) enforces a hard boundary preventing stochastic errors from entering a deterministic reasoning engine. DLCA's PPU with HITL is distinguished by making input fidelity a governance requirement, not merely a software engineering concern.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

3.6 AI Advisory Platforms in Law

Domain-specific AI advisory platforms have emerged in legal practice, representing the same category of probabilistic, unauditable prior art as clinical NLP systems.

In legal practice, AI research tools from Thomson Reuters (Westlaw AI-Assisted Research, CoCounsel) and LexisNexis (Lexis+ AI) have achieved broad deployment as the dominant legal research platforms, however, it has been found hallucinations have not been eliminiated [52]. A subsequent evaluation by Munir et al. [53] confirmed that Lexis+ AI outperforms general-purpose LLMs on legal tasks but continues to produce fabricated responses [53].

AI-assisted legal drafting and research platforms have achieved significant adoption at major law firms. No vendor has published systematic benchmarking data [52]. These tools perform legal research, summarisation, and drafting but provide no formal proof-path, no lemmatic evaluation, and no operator-approved input fidelity gate.

3.7 AI Advisory Platforms in Academic Research

In academic research, AI-assisted literature synthesis platforms including Elicit [54] and Consensus [55] have achieved broad adoption. A peer-reviewed evaluation of Elicit [54] found it a valuable complementary tool for systematic review that requires human oversight and does not replace traditional search methods [54]. A systematic review of Consensus App [55] identified significant evidence-synthesis potential alongside documented ethical concerns and reproducibility limitations. These platforms synthesise literature probabilistically without formal audit trails or operator-approved query interpretation.

3.8 Safety-Critical Control Systems

The broader field of safety-critical control—fly-by-wire avionics (ARINC 653 [56] [57]) , automotive safety (ISO 26262 [14]) , industrial control (IEC 61508 [58]) —provides production evidence that deterministic, time-bounded logic is not merely desirable but mandatory in physical control domains. DLCA's physical control extension builds directly on this tradition. The distinction is that existing safety-critical systems are domain-hard-coded with no natural-language interface and no cultural or jurisdictional reasoning layer. DLCA generalises the deterministic-logic-with-governed-interface principle across both advisory reasoning domains (legal, medical, ethical) and physical control domains.

3.9 Modular Cognitive Architectures and Dual-Process Framing

The classical modular cognitive architectures, ACT-R [59], SOAR [60] [61], and CLARION [62], are important intellectual predecessors to DLCA’s modular framing. The dual-process cognitive science literature provides the broader theoretical framing represented by: System 1 (fast, implicit, automatic) vs. System 2 (slow, deliberate, explicit) [63]. DLCA is best understood as a stricter, governance-oriented, physically-bounded instantiation of this broader architectural family: where ACT-R, SOAR, and CLARION decompose cognition into cooperating modules. DLCA assigns canonical epistemic authority to the deterministic LC alone, with the PPU/SI as governed boundary layers with limited autonomous reasoning role.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Table 3.2 provides a structural comparison of different human cognition architectures and the DLCA architecture. DLCA's comparative advantage is in governed reasoning with audit-able outputs, not in cognitive modelling or learning flexibility. For simulation of human cognition, ACT-R and SOAR remain the established platforms.

Dimension	ACT-R	SOAR	CLARION	DLCA
Core purpose	Model human cognition	General intelligent agent	Implicit/explicit cognition model	Governed epistemic reasoning architecture
Epistemic authority	Distributed across modules	Distributed across knowledge stores	Distributed across subsystems	Centralised in deterministic LC
Determinism	Probabilistic/ subsymbolic throughout	Mixed symbolic/learned	Dual implicit/explicit; both adaptive	Deterministic LC; probabilistic only at governed boundaries
Language relationship	Language is part of the agent stack	Language integrated with task learning	No hard boundary between reasoning and expression	Language and interpretation demoted to governed translator (SI)
Immutability	None; continuous learning	None; continuous learning	None; continuous learning	Hardware-enforced (ASIC/OTP)
Audit trail	Not a design goal	Not a design goal	Not a design goal	Audit File by design
Cultural pluralism *	Single implicit ontology	Single implicit ontology	Single implicit ontology	Seven explicit vocabulary families (L12) and 7 identified cultural groups
Human approval gates	Not required	Not required	Not required	Mandatory at all probabilistic boundaries (L10)

Table 3.2 — DLCA vs. Classical Cognitive Architectures

* Cultural pluralism in this section refers to those cases when a master culture is derived from multiple cultural or language roots.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

4. The Three-Module DLCA Architecture

4.1 Overview

The primary architecture discussed herein is the three-module DLCA architecture where the cognitive function of an AI system is separated into three operationally distinct components (PPU, LC and SI) joined by formally specified protocols. Probability is permitted to operate in exactly two locations: inside the PPU (input normalisation) and inside the SI (output rendering). Both probabilistic regions are bounded by human approval gates. The LC operates deterministically throughout with bounded evaluation complexity.

Figure 4.1 shows the primary elements, a three-module architecture (PPU, LC, SI) all writing to the Audit File (AF). The internal architecture has inter-module human approval gates (HITL protocol) separating probabilistic and deterministic operations at each boundary, spanning both software and hardware realisation. Figure 4.2 shows the Three-Layer Functional overview with internal architecture also shown. HITL, QCM, language processing units, and cultural processing units are introduced and discussed in subsequent sections.

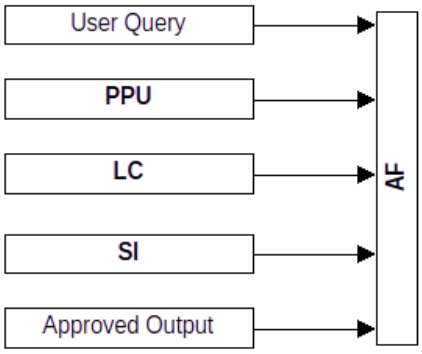

Figure 4.1 — DLCA Three-Layer Functional Overview. Pre-Processing Unit - PPU, Logical Core - LC, and Social Interface – SI, showing overall inputs and outputs. All stages write to the Audit File (AF).

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

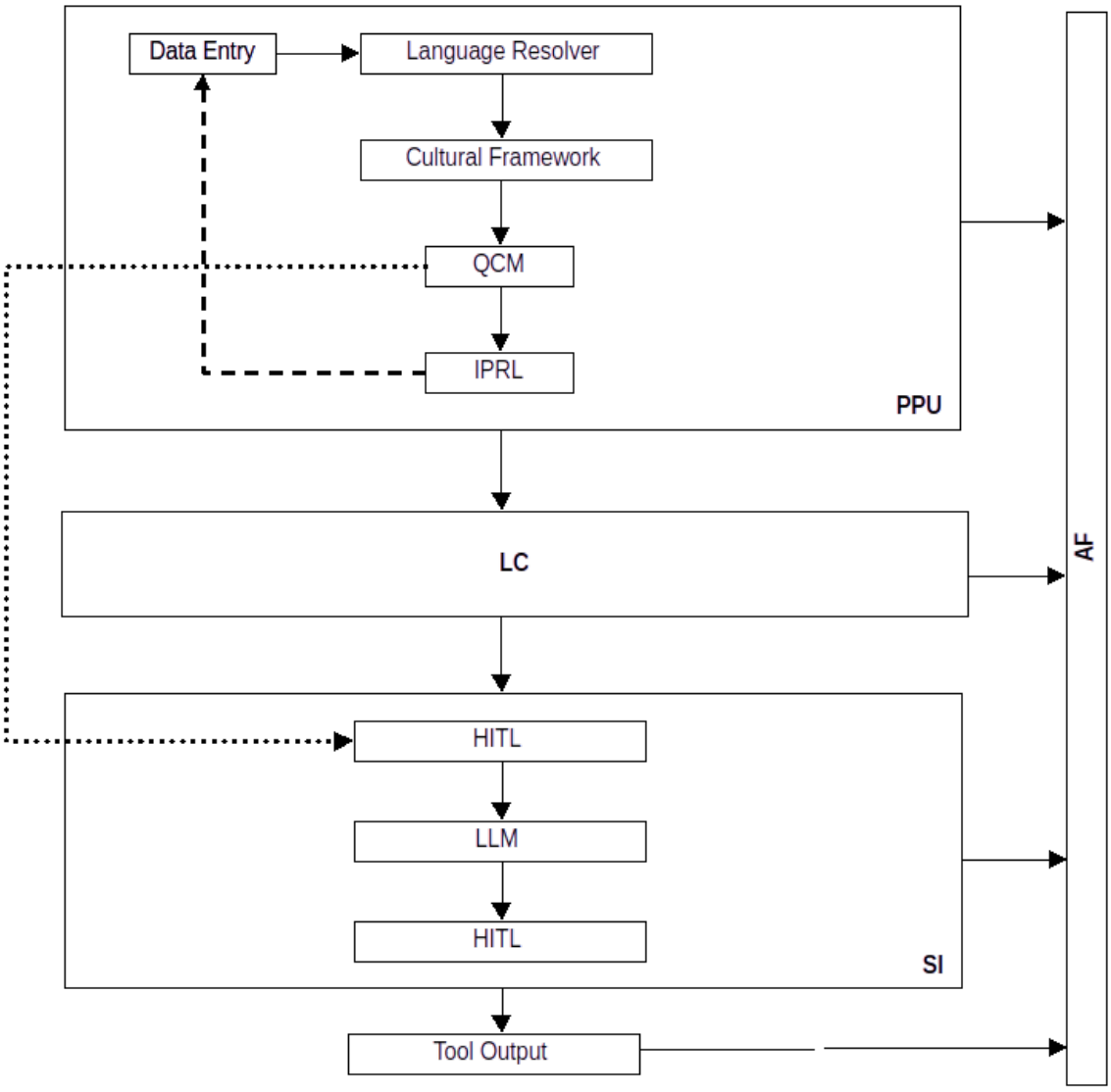

Figure 4.2 — DLCA Three-Module Architecture with Type-F bypass pathway and Audit File.

5. Human-in-the-Loop (HITL)

5.1 Rationale

HITL is an architectural feature of DLCA allowing evaluation of all probabilistic outputs. It appears at three positions in the processing pipeline, each with a distinct function: (1) Within the PPU at the output to LC. (2) at the LC output. (3) at the SI output. This multi-stage model is for example purposes allowing HITL to be a deployment variation. A minimal deployment may activate only one HITL. A maximum HITL deployment activates all three.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Two deployment classes require qualification of this model. First, for real-time physical control where per-inference latency must meet δmax (e.g., emergency braking at ~5ms), per-inference HITL is architecturally impossible. Second is for use by minors who may not have the capability to evaluate the proposition to be processed.

Human factors research is directly relevant to HITL gate reliability. Operator review fatigue, time pressure, and decision saturation are documented phenomena in safety-critical human-machine systems [83] that affect approval-gate accuracy over extended sessions.

The HITL can be configured to process information from one or more humans, thereby allowing supervision/approval of human interaction by another human.

6. The Pre-Processing Unit (PPU)

The Pre-Processing Unit is needed to rectify variations in language, culture, operator expressions and data gathering modalities described in Section 6.2. To achieve repeatable and accurate results, the deterministic logical core will require normal variations be rectified prior to processing. Rectification can be a combination of language translation, cultural translation, terminology translation, API extractions and sensor calibrations. In high value logical operations, a trained operator can provide a rational check or correction of the process.

Insomuch as feasible rectification in the PPU can be deterministic language and cultural translation formed by standards or religious bodies.

An example of PPU functionality follows.

> A pilot speaking arabic may need to access a western focused module in english focused on the operation of aircraft. The operator may be culturally or language unaware of certain english terminology. The translation and cultural unit can increase specificity by translating from the source language into specific english language proposition.

6.1 Rationale: Why Translation Must Not Reside in the SI

Assigning the input translation function to an LLM creates a logical contradiction. A stochastic error in translation produces a formally incorrect proposition that the LC then evaluates deterministically. This process can generate a deterministically wrong answer from a stochastically corrupted input. Lemma L11 makes this explicit, input fidelity is a necessary precondition for output validity. The PPU is designed to address this by housing the translation function in a dedicated unit whose outputs are subject to iterative human approval before reaching the LC. In addition deterministic translations managed by standards bodies or religious organizations can help reduce rectification errors.

The PPU is itself a significant independent research agenda. Its architecture varies substantially by application domain: a medical PPU enforces strict cardiology-specific vocabulary (rejecting lay terms such as 'heart' in favour of formally grounded clinical terminology); a machine-control PPU converts sensor signals into kinematic variables; a child-safety PPU hard-codes a restricted concept space to prevent inappropriate inputs from reaching the LC. Different

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

domains may require materially different PPU architectures. The PPU may be implemented as an NLP-based language resolver, an LLM-based resolver, or a hard-coded domain-specific resolver, depending on operator requirements and skill levels. Hard-coded resolvers are most appropriate for child safety and machine control; LLM-based resolvers provide maximum flexibility for expert research users.

While some translation is as simple as equating one word with another, much translation is ambiguous. [64] The complexities of translating language and culture are sufficient that translators have in some cases failed to achieve success. In the event translation fails a system designer can utilize a HITL process to produce an acceptable output.

6.2 Input Modalities

The PPU's input scope is not limited to human-generated queries. Structured data feeds may include: weather service APIs, government open-data portals (legislative calendars, court dockets, school schedules), RSS feeds, and regulatory data streams. These feeds can also arrive in machine-readable schema form or as continuous sensor streams. The recognized data feed types are presented in Table 5.2 along with some important characteristics conceptualized for each.

Modality	**Description**	**Primary User**	**PPU Workload**	**LC Input Quality**
M1 — Free-form Chat	Open text box; typos and ambiguity expected	Expert users, researchers	High: correction, disambiguation, IPRL cycles	Medium — IPRL essential
M2 — Guided Multiple Choice	Structured question sets; constrained option lists	Lay users, clinical/legal intake	Low: options pre-mapped to propositions	High — option set pre-verified
M3 — Yes/No Branching	Binary decision tree narrowing proposition space	General public, triage, compliance	Very low: binary paths map directly to lemmas	Very high — path space enumerated
M4 — Structured Template Form	Named field entry (engineering spec, contract clause)	Professional domain users	Low: field values substituted into templates	High — template grammar pre-formalised
M5 — Voice with Transcription (incl. Voice Assistant and Emergency Services)	Spoken input transcribed and normalised. Covers voice assistants (structured confirmation loops), emergency services, and hands-free professional input. LLM can interactively prompt with queries.	Accessibility, hands-free contexts	High: Information set completion, transcription correction + M1 normalisation	Medium — IPRL essential
M6 — Document	PPU parses uploaded document and extracts	Legal, medical,	Very high: extraction, segmentation, cross-	Medium-High — structure-

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Modality	Description	Primary User	PPU Workload	LC Input Quality
Upload	propositions	engineering professionals	reference	dependent
M7 — Machine API	Structured data arrives in defined schema; PPU validates	Automated pipelines, enterprise integration	Minimal: schema validation only	Very high — schema enforces grammar
M8 — Sensor Stream (SSN)	Continuous numerical sensor data converted to propositions at hardware speed	Physical control systems	Hardware-speed threshold/state/derivative classification	Very High (deterministic classification)

Table 5.2 — PPU Input Modality Taxonomy including M8 Sensor Stream Normaliser. A single deployment may support multiple modalities simultaneously.

6.3 Query Classification Module (QCM)

When the DCLA architecture is implemented within an LLM a need can evolve to keep deterministic logic functionally distinct from probabilistic logic. QCM is proposed to classify queries and create a processing route for queries that maximizes determinism. The QCM is a classifier and logic router operating on raw input before the HITL is invoked. It assigns each query to Type 1, 2, or 3 where Zone 1 is an entirely deterministic query, Zone 2 is a combination of deterministic and probabilistic query and Zone 3 is purely probabilistic.

Examples showing usefulness of the zone classification follow.

a) A query for 'all options for purchasing 915 MHz LoRa devices' is Zone 2 (deterministic LC constraint evaluation of available options, preceded by probabilistic SI search/retrieval). A query for 'all options' is Zone 3—the word 'all' has no proof attachable to it; completeness over an open-world domain cannot be formally verified.

b) A lawyer asking LLM to propose writing a letter to push back on an effort by a clients opposing attorney to challenge contract terms. The contents preparation by the LLM can include an LLM search of the relevant legal database and return the court cases that support arguments why the opposition is wrong. The LLM can incorporate these cases by reference in a proposed letter that is aggressively toned to push back on the challenge to contract terms.

In high-value deterministic deployments (e.g. legal, medical, regulatory, and safety-critical applications) the HCM and Zone Classifier can be used to screen queries for probabilistic elements. Those queries with probabilistic elements can be rejected.

Table 6.1 summarizes characteristics of the 3 zone classification proposal. Zone assignment is dynamic. Operator may constrain deployment to Zone 1 queries only. Default Zone 3 display shows SI probabilistic outputs.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Zone	Name	Conditions	LC Authority
Zone 1	Full LC Authority	All FRP premises derived from unambiguous formal inputs	Full - deterministic, auditable, reproducible
Zone 2	Shared Authority / Multi-Pass	Valid proof-path exists for at least one plausible premise interpretation	Conditional - Portions are: deterministic, auditable, reproducible
Zone 3	Probabilistic	task logical structure not definable within any available deterministic framework	Full SI probabilistic output shown simultaneously by default

Table 6.1 — Three-Zone Epistemic Classification.

6.4 The Iterative Proposition Refinement Loop (IPRL)

The IPRL is a optional processing specification for HITL implementation in the PPU. It provides an operator-supervised iterative loop that continues until the operator is satisfied that the Candidate Proposition Set (CPS) faithfully represents intent. Each IPRL cycle is written to the Audit File, providing a complete record of how the final Formal Request Packet (FRP) was reached. The IPRL approval model can be applied as a retrofit to any AI system inserting human approval of the system's interpretation of the user's input before processing begins. This provides an incremental adoption pathway and a Stage 1 test methodology for DLCA concepts across existing infrastructure. While there are many options for the IPRL control loop, one option is shown in Figure 6.3.

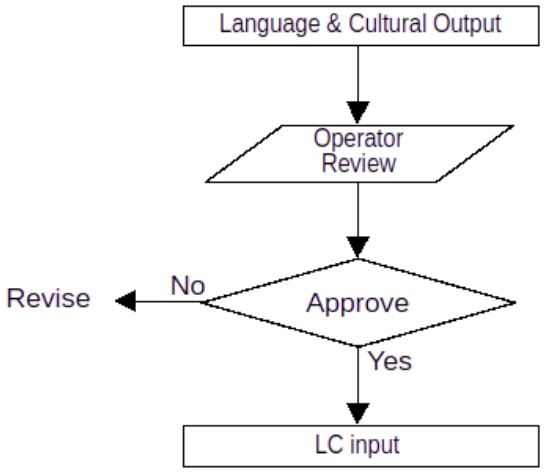

Figure 6.3 — IPRL: operator supervises each cycle via a second chat interface until explicit approval. All cycles written to Audit File.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

7. Logical Core (LC)

The Logical Core receives a request from the PPU and evaluates it deterministically. The output is returned to the SI for rendering. For any given input the output is unique, reproducible, and traceable to the foundational documents or expert sources from which the logic was derived.

LC evaluation is expressed as mathematical relationships — Boolean, SAT, or first-order logic formulations. Evaluation time is determined by lemma set size at fabrication, not by input length at inference time.

The internal representation language of the LC is an open architectural parameter. English is not assumed to be the most efficient or appropriate substrate for logical formalization across all domains. Formal logic notation, domain-specific symbolic languages, or the native formal vocabulary of the relevant cultural or legal tradition may yield more compact and verifiable structures.

The LC mathematical representations may be languge and culture-specific. Each vocabulary or cultural family identified in Section 8 may require its own LC instantiation.

The LC writes every evaluation step, lemma identifier, and source metadata tag to the Audit File at write point AF-04. This proof-path satisfies the auditability requirement and supports EU AI Act Article 13 transparency provisions. Where lemma sets derived from different foundational sources produce conflicting conclusions, the LC surfaces all divergent proof-paths at AF-04. Reconciliation and human-readable explanation of any conflict is assigned to the SI after HITL review.

The LC is the sole epistemic authority in the DLCA architecture. Its authority is bounded on the input side by the PPU's input fidelity protocols and on the output side by the SI's governed rendering function. No probabilistic process operates within the LC boundary. This structural separation is the central architectural commitment of DLCA and the primary basis for its precision, reproducibility, and auditability claims.

8. Language and Cultural Framework

A defining feature of DLCA is to locate language and culture translation in the PPU just prior to a potential HITL gate. The intended result is to disambiguate queries before the LC is engaged to process. Placing the language and cultural units into the PPU provide a natural separation which will allow these units to be managed separately by governmental or religious entities.

8.1 Why Setpoint Weighting is Insufficient for Cultural Diversity

Philosophical traditions differ not merely in how they weight shared concepts but in which foundational ontological categories they employ. Western analytical ethics operates on individual rights, duties, and aggregate utilities. Confucian ethics operates on relational

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

obligations within defined social relationships. Islamic jurisprudence (fiqh) operates on divine commands, permitted categories (halal/haram), and scholarly consensus (ijma). Ubuntu philosophy operates on communal person-hood.

These differences cannot be bridged by setpoint adjustment; they require separate lemmatic vocabularies. This is not a claim about cultural relativism but about logical incompatibility. Empirical cross-lingual transfer learning research demonstrates measurable limits on how much of one tradition's categorical structure can be represented in another's vocabulary without information loss [65] [66]. Efforts to systematise philosophy computationally [67] [68] [69] [70] and to apply natural-language processing Mayo-Wilson to religious and philosophical corpora [70] have encountered this challenge.

Furthermore, natural language itself changes with culture and over time. What may be a single concept in one language may have mutiple distinct expressions in another.

The PPU must therefore be language and culture specific. The cultural vocabulary families described in 7.2 are architectural placeholders awaiting formal specification by domain scholars. Formal vocabulary specification requires engagement with philosophy of language (cross-linguistic semantics [66]), ontology engineering (OWL 2 [71], formal ontology alignment [72]), and intercultural AI governance [18].

8.2 Language families

DCLA's Cultural Framework Router (CFR) organises output vocabulary along cultural-geographic lines. An orthogonal dimension—typological structure—governs how propositions are encoded in natural language and therefore how reliably the Social Interface (SI) renders Logical Core (LC) outputs. Morphological type, word order, and phonological organisation all affect proposition parsing. Under the neo-Whorfian position, grammatical structure influences causal and relational framing [73] [74] [75], and SI models trained on Indo-European-dominant corpora exhibit measurable propositional-fidelity asymmetries across language families [76]. Language-family classification in the CFR directly mitigates this asymmetry.

Family	Representative Languages	Morphological Type	Word Order	Key DLCA-Relevant Features	Speakers (L1+L2)
Indo-European	English, Spanish, Hindi, Russian, Persian, Bengali	Fusional	SVO (Germanic/Romance); SOV (Indo-Iranian); free (Slavic)	Case inflection; grammatical gender; tense-aspect-mood morphology; large derivational inventory	~3.5 B
Sino-Tibetan	Mandarin, Cantonese, Wu, Tibetan, Burmese	Isolating	SVO (Chinese); SOV (Tibetan, Burmese)	Tonal; classifiers; topic-prominence; no word delimiters; serial verb constructions	~1.5 B
Afro-Asiatic	Arabic (MSA + dialects), Hebrew, Amharic, Hausa, Somali	Fusional (Semitic); Analytic (Chadic)	VSO/SVO (Arabic, Hebrew); SOV (Amharic)	Root-and-pattern morphology; diglossia; pharyngeal phonemes; broken plural	~600 M
Niger-Congo	Swahili, Yoruba, Igbo, Zulu, Shona, Fula	Agglutinative (Bantu); Isolating	SVO (majority)	Noun-class system (up to 20 classes); tone; cross-referential agreement prefixes	~700 M

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Family	Representative Languages	Morphological Type	Word Order	Key DLCA-Relevant Features	Speakers (L1+L2)
		(Kwa)			
Austronesian	Malay/Indonesian, Tagalog, Javanese, Malagasy, Hawaiian	Agglutinative; focus-marking	VOS/AVO (Philippine-type); SVO (Malay)	Symmetrical voice system; verb-marked thematic role; reduplication; no morphological case	~380 M
Dravidian	Tamil, Telugu, Kannada, Malayalam	Agglutinative	SOV	Suffix-only morphology; postpositions; retroflex consonants; extensive pro-drop	~250 M
Turkic	Turkish, Uzbek, Kazakh, Azerbaijani, Uyghur	Agglutinative	SOV	Vowel harmony; exclusively suffixal; postpositions; evidentiality marking; no grammatical gender	~200 M
Japonic	Japanese, Ryukyuan languages	Agglutinative	SOV	Grammaticalised honorific register; topic-prominence; postpositions; verb-final; pro-drop	~130 M
Koreanic	Korean	Agglutinative	SOV	Morphologically encoded speech levels; postpositional case markers; evidential suffixes	~80 M
Uralic	Finnish, Estonian, Hungarian	Agglutinative	SVO (Finnish/Estonian); flexible (Hungarian)	Extensive case systems (Finnish: 15 cases); vowel harmony; no grammatical gender	~25 M
Nilo-Saharan	Dinka, Luo, Kanuri, Songhay	Isolating to mildly agglutinative	VSO / SOV	Tone; complex verb extensions; limited inflectional morphology	~60 M
Language Isolates	Basque (primary example)	Ergative-agglutinative	SOV	No genealogical affiliation; ergative-absolutive alignment; polypersonal verb agreement	~1 M+

8.3 Cultural Lemmatic Vocabulary Families

Cultural factors can be classified as families of cultures finding common ground in broad categories. Addressing impacts of sub-cultures is beyond the capability of an overview investigation. Table 7.2 identifies major cultural families which are sufficiently structurally different as to require separate implementations for achieving deterministic and repeatable results. Each family requires a distinct LC vocabulary layer and its own cultural layer. Foundational references by family (primary texts and peer-reviewed scholarly monographs): Western Analytical [77] [78]; Confucian [79] [80]; Buddhist [81]; Islamic [82]; African Communitarian [83]; Hindu Dharmic [84]; Positive Legal (jurisdiction-specific).

Multiple examples exist illustrating the difficulty or impossibility to resolve logical frameworks for cultural axioms [85] [86]. This inability does not negate value of interpretation but limits the extent of determinism. The system designer(s) when encountering these edge cases have multiple options to either resolve the difficulties or to note the process returns an error. Some combination of error message, LLM analysis, HITL or further information query can help resolve the discrepancies in an acceptable form.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Vocabulary Family	Philosophical Tradition	Foundational Ontological Category	Core Lemmatic Distinctions
Western Analytical	Kantian Deontology, Utilitarianism, Rawlsian Justice	Individual rights, duties, aggregate utility	Rights-holder/duty-bearer; maximise welfare; original position fairness
Confucian Relational	Confucianism, Neo-Confucianism	Relational obligations; five relationships; ren; li	Agent A's obligation to B within relationship R; family > state > self
Buddhist Consequential	Theravada/ Mahayana ethics	Reduction of suffering; non-attachment; interdependence	Propositions reference suffering-states and causal chains, not rights
Islamic Jurisprudential	Fiqh; Usul al-Fiqh	Divine command; five necessities; maslaha	Haram/halal classification; ijma as lemma; qiyas as proof method
African	Ubuntu philosophy; African jurisprudence	Communal personhood; collective wellbeing; restorative justice	'I am because we are' — individual validity contingent on community framing
Hindu Dharmic	Dharmashastra; Arthashastra	Dharma; karma; varna-specific obligations	Propositions are role-dependent; same action may be dharmic for one role, adharmic for another
Positive Legal	Common Law, Civil Law, Statutory Law	Statute, precedent, procedural rules	Jurisdiction-specific lemma sets; update-governed

Table 7.2 — Cultural Families.

9. Social Interface (SI)

The Social Interface (SI) is a large language model operating under a structured harness. It receives normalised propositional output from the Logical Core. It renders that output as natural language suitable for the end user. The SI does not reason. It does not modify LC conclusions. Its sole function is accurate, contextually appropriate expression of LC-determined content.

The SI harness enforces two checking mechanisms. The first is self-checking. The SI evaluates its own output against the LC propositional form before release. Where divergence is detected the SI initiates a revision cycle. Three divergence types are recognised: lexical drift, where the SI substitutes terms not present in the LC output; qualifier omission, where hedging terms present in the LC output are dropped; and modality shift, where the epistemic status of a conclusion is altered in rendering. The second mechanism is peer-checking. A second SI instance independently renders the same LC output. The two renderings are compared. All

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

divergence events and revision cycles are logged to the Persistent Audit File at write point AF-06.

The SI performs two translation functions. The first is propositional rendering: conversion of formal LC output to natural language. The second is language translation: where the user's original query was submitted in a language other than the LC evaluation language, the SI renders the final output in the user's source language. For both functions the SI draws on the Language and Cultural Translation defined in the PPU.

The SI may be implemented as an agent in an orchestration framework such as n8n, permitting complex multi-step rendering pipelines and peer-checking workflows.

10. Output Precision Metric (OPM)

OPM is proposed to identify degree of SI output variation from inputs. Quality can be improved if a machine produced metric can identify output for deeper review. "SelfCheckGPT" has been proposed as one method to qualify LLM output precision [87].

10.1 SI Translation Problem

PPU output can be operator approved before reaching the LC. Output translation is assigned to the SI. The SI must render the LC's canonical conclusion faithfully.

10.2 Output Precision Metric: Specification

The OPM is a signal by which the LC's deterministic canonical output is compared to the SI's probabilistic rendering. There are infinite structural variations for how metrics are formulated and reported. Generally both qualitative and quantitative methods can provide value.

This paper therefore establishes the OPM as an optional architectural component of DLCA without prescribing its implementation.

11. Audit File (AF)

The Audit File (AF) is an append-only, data store capturing every data event in a DLCA processing cycle. It is intended to provide a traceable record linking every element of every SI output back to the raw user input while allowing identification of the source documents from which the logic used to generate the result is derived.

The AF architecture is directly implementable using established tamper-evident logging techniques. Hash-chained, append-only Merkle-tree log structures [88], as standardised in IETF RFC 6962 Certificate Transparency [89], provide cryptographic proof that no historical record has been modified, deleted, or inserted out of sequence. For industrial control deployments, IEC 62443 [90] provides complementary cybersecurity requirements for audit logging at the system level.

Table 6.1 proposes a partial specification for logical flow write points.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Write Point	Content Written	Written By	Trigger
AF-01	Raw user input (all modalities); timestamp; session ID; QCM classification	PPU / QCM	Immediately on input receipt
AF-02-N	CPS for IPRL cycle N; confidence scores per proposition; ambiguity flags	PPU (per cycle)	After each IPRL processing cycle
AF-02-N-op	Operator action: APPROVE / MODIFY / ADD / REJECT with annotation	Operator (via IPRL interface)	Immediately on operator action
AF-03	Final approved proposition; cultural context tag; setpoint vector S; zone classification tag; operator approval signature	Operator Gate	On final IPRL approval
AF-04	Logical elements used in evaluation along with meta data indicating foundational source of logic.	LC	On LC processing completion
AF-05	HITL decision: expert ID, timestamp, CDT cryptographic signature, rejection annotations, zone context acknowledgement	HITL Interface	On HITL commitment or rejection
AF-06	All SI renderings; per-element lemma attribution; OPM components ; OPM scores; classification for Zone 1/2/3 outputs	SI	On SI rendering completion

Table 6.1 — Proposed Audit File write points.

12. Hardware Realisation

12.1 Design Philosophy

A central distinction between DLCA and prior AI reasoning systems is that epistemic authority for DLCA resides in a component whose evaluation complexity is bounded at fabrication time, rather than scaling with input length at inference time. This is accomplished through construction of mathematical relationships to represent the logic which can then be either proprietary or a public facing process. This is the same methodology employed in SAT systems discussed above [91].

In hardware implementations, logic etched in ASIC silicon or programmed via OTP eFuse cannot be altered by software updates, adversarial prompting, retraining, or operator configuration drift. The broader move toward domain-specific hardware accelerators [91] and the demonstrated efficiency of FPGA and ASIC implementations relative to general-purpose processors [92] [93] [94] provide the engineering context for hardware LC realisation. In software implementations, disciplined version locking and access control approximate this property. Both NLP-based and LLM-based software implementations are valid DLCA architectures. The hardware version is presented as an evolutionary step beyond current

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

software efforts, intended to provide lower marginal cost, stronger immutability, and real-time performance not available in software-only implementations.

The governance protocols proposed in the QCM process are designed to ensure that only independently validated content reaches commitment. Immutability is thereby an amplifier of sound governance rather than a substitute for it. Within its defined epistemic domain, the LC achieves output determinism by construction. For any Zone 1 input, there exists a unique logically derived consequence, however these outputs can conflict or provide different indications especially in the case of comparison between foundationa documents and expert logical relationships. The LC hardware reproduces this consequence identically across all invocations. The SI is tasked with reconciling these outputs and explaining any differential or conflicting output in human terms.

The efficiency advantage of bounded-complexity LC evaluation over transformer models is established as a formal property of the architecture.

Software attack vectors are a real consideration in software-only deployments: the ephemeral nature of software LCs creates potential vectors for modification that hardware implementations reduce by design. This does not invalidate software implementations.

12.2 Implementation Options

Technology	Immutability	Cultural LC Support	Update Capability	Recommended Use
Pure Software (SAT-solver + Lemma DB)	Low	Full configurability	Full	Stage 1 validation;LLM implementations
Full Custom ASIC	Absolute	Multiple vocabulary layers on-die	None	Mass-market LC
FPGA + OTP eFuse	High	Configurable pre-OTP	FPGA	Enterprise LC; single vocabulary family
FPGA (SRAM-based)	Low	Full configurability	Full	LC requiring updates; development
OTP eFuse Array	High	Limited by eFuse density	None	Embedded LC; minimal vocabulary
Hot-Swap OTP Cartridge	High	Per-cartridge vocabulary patch	Monthly swap	High-fidelity updates
Multi-Bank OTP eFuse Switchover	High	Per-bank vocabulary update	Monthly to quarterly	High-frequency updates requiring hardware-level integrity without physical cartridge swap
Co-Package (ASIC+NPU)	High	Separate die	Rarely	Production DLCA

Table 7.1 — Hardware implementation options including software-only implementations.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

12.3 Hardware Update Strategies

A practical constraint is the scientific, legal and medical LC update requirements (potentially frequent, depending on domain and jurisdiction) and refresh cycles (which vary by implementation tier). The following illustrative ranges are indicative only. Actual update cycles cannot be known in advance of empirical deployment data.

LC content evolves at rates that vary substantially by domain class. Legal precedents, scientific standards and clinical guidelines change on timescales of months to years, whereas mathematical lemmas, anatomy, and physical constants are stable indefinitely. Class S (Static): mathematical axioms, the periodic table, constitutional provisions, fundamental physical constants. These never or rarely change and are best encoded in ASIC or OTP eFuse for maximum immutability. Class M (Moderate): clinical practice guidelines, case law precedent, engineering standards revisions. These may change on a periodic basis and suit FPGA SRAM or hot-swap OTP cartridge. Class D (Dynamic): emerging research consensus, breaking legal precedent, regulatory guidance updates. These may change frequently and suit the PPF software layer. Implementation options for the update layer are designated as a Companion Paper 1 deliverable.

Strategy	Mechanism	Update Frequency	Immutability	Best For
Post-Processing Filter (PPF)	Software layer applying recent rule changes without modifying silicon	Daily to monthly	Low (software)	High-frequency minor changes; emergency recalls
Hot-Swap OTP Cartridge	Physically replaceable OTP eFuse module; each cartridge holds incremental lemma additions	Monthly to quarterly	High (hardware per cartridge)	Mission-critical legal/medical field updates
Multi-Bank OTP eFuse Switchover	Multiple OTP banks on-die; spare bank written while primary is active; cryptographically authorised switchover	Monthly to quarterly	High — each bank is OTP	High-frequency updates without physical cartridge swap
FPGA SRAM Reload	Full FPGA vocabulary reprogramming under multi-party cryptographic authorisation	Quarterly to annually	Medium (reload risk; mitigated)	Major legislative or clinical guideline changes

Table 8.2 — Hardware Update Strategies. Strategies may be combined.

13. Marketplace Model and Regulatory Economics

The logical cores for language, culture and domains are to be constructed from foundational documents, peer reviewed articles, experts and other primary sources. It is anticipated the extraction of lemmas and construction of logical relationships will be subjects of publically reviewed process published to allow public or scholarly access. The process of construction can

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

be facilitated by LLM's and/or AI agents with Human in the Loop methodologies. Driving forward through the logical relationship will provide an output. Meta data identifying source of logic is stored providing an audit-able file to identify the source of logical relationships. Result is a logical system to generate output which allows an audit trail for scholarly review.

13.1 Three Independent Components

The three tier model will enable incremental testing and governance of the components. A PPU can be tested with an LLM as the LC. An LC can be tested with an LLM substituted for the PPU. This modularity will also allow PPU to be governed by one agency while the LC is governed by another.

13.2 Server-Based Deployment Model

Although DLCA is suitable for individual machine deployment, the DLCA can be implemented as a part of sophisticated LLM. LLM's have achieved remarkable capability.

14. Testing Hierarchy — Complete Eight-Test Suite

Testing can commence with analogue tools such as SAT solvers being feed propositions evaluated by harnassed LLM's configured to rectify language and cultural differences. Testing will not only confirm or invalidate hypothesis but likely illustrate relative value of specifications for components and may inform regulatory efforts.

14.1 Two-Tier Testing Philosophy

DLCA's architecture distinguishes between formal verification (establishing truth by construction within a closed axiomatic system) and empirical measurement (establishing probabilistic bounds over observed behaviour). The deterministic portions, LC, language resolver and the cultural resolver, can be formally verified. The probabilistic portions, the QCM, HITL, and SI, are quality controlled only by empirical measurement.

15. Physical and Robotic Control Applications

For academic completeness physical and robotic applications are identified as a exhibiting characteristics of the lemma derivations as currently implemented. It is expected identification of these features will allow LLM's and these physical domains to be combined in ways that benefit both processes by sharing hardware features and integrity rules.

15.1 Why Physical Systems Are a Natural DLCA Domain

Physical systems governed by known engineering parameters represent a deployment domain for the DLCA Logical Core. Invariant physical laws, deterministic constraint satisfaction, bounded actuator responses, and fixed structural tolerances are precisely the class of immutable canonical knowledge which is envisioned to be encoded in the LC. The efficiency

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

argument is qualitatively different from advisory reasoning deployments. In physical control, the bounded LC evaluation advantage over LLM reasoning is a safety argument.

A self driving automobile example is given below.

> Self driving automobiles operate through sensor inputs. These systems are however imperfect and a human observer or passenger can provide valuable input if configured to do so. Say a person in a car notices the automobile is heading towards a hazard without evidence the control systems sees the hazard. The person in the car can scream or otherwise signal the control system by saying ‘stop’. This natural language can be put into the car’s control algorithm either to look for a potential anomaly or to execute the command.

The LC addresses both through the same hardware mechanism. Refined proposition is evaluated against a fixed lemma set derived from foundational documents or experts. The unified implementation is valid because both are forms of deterministic evaluation over fixed, pre-committed knowledge structures. This is the key property that distinguishes both from stochastic LLM inference.

For safety-critical control loops—aircraft fly-by-wire, robotic collision avoidance, automotive emergency braking—the difference between 50ms and 2ms is not a performance preference; it is the difference between safe operation and catastrophic failure, as established in IEC 61508 [58], ISO 26262 [14], and DO-178C [56].

Existing production systems—automotive ADAS (ISO 26262) [14]—provide field demonstration of the feasibility and necessity of real-time deterministic control. The DLCA contribution is a unified architecture spanning both advisory reasoning domains and physical control domains, with the same audit, cultural pluralism, and human-oversight provisions across both.

15.2 Physical Parameter Tables

Physical Parameter Tables (PPTs) constitute a new LC content class with distinct properties. They are engineering specifications, not philosophical axioms. They are immutable during operation but updateable. Table 13.1 lists recognized classes PPT’s.

Physical Domain	Hard-Coded LC Parameters (PPT)	Probabilistic SI Inputs	Real-Time δmax
Aviation / Fly-by-Wire	Aircraft mass, CoG envelope, stall speeds vs config, structural load limits, control surface authority, engine thrust curves, airspace constraints	Pilot commands (voice/stick), sensor fusion (IMU, GPS, pitot), weather data, ATC instructions, terrain proximity	10 ms (representative target in certified fly-by-wire systems per DO-178C/ARINC 653 practice)
Industrial Robotics	Joint torque limits, reach envelope, payload capacity, collision	Operator commands, vision system object classification, force	2 ms (representative target observed in ISO 10218-compliant

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026
DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

Physical Domain	Hard-Coded LC Parameters (PPT)	Probabilistic SI Inputs	Real-Time δmax
	geometry, motor current limits, tool offsets, safety zone boundaries	sensor readings, collaborative human proximity	collaborative robot deployments)
Machine Tools (CNC)	Max feed rate by material/tooling, spindle speed limits, axis travel limits, tool life parameters, fixture force limits, thermal compensation tables	G-code program interpretation, tool wear estimation, surface quality assessment, adaptive control inputs	5 ms (representative target in IEC 61800-compliant CNC servo control practice)
Automotive ADAS/AV	Vehicle mass, braking envelope, steering geometry, stability limits, pedestrian detection thresholds, traffic law constraints by jurisdiction	Camera/LiDAR scene interpretation, intent prediction of other road users, natural language navigation commands	5 ms (representative target in ISO 26262 ASIL-D automotive emergency braking deployments)
Agricultural Autonomy / Precision Farming	Field geometry, crop parameters, implement constraints, geo-fencing boundaries	Vision system crop/obstacle classification, soil sensor streams, weather data, operator commands	20–50 ms (domain-specific)
Electronic Warfare Avoidance	Platform signature constraints, emission control rules, countermeasure parameters, engagement envelopes	Threat detection sensor streams, environment classification	Sub-5 ms (mission-critical)

Table 13.1 — Physical control domain applications.

15.3 Sensor Stream Normaliser

Continuous sensor streams are not amenable to HITL disambiguation. These streams require real-time conversion into discrete formal propositions. Modality M8 (Sensor Stream Normaliser) performs threshold comparison, state classification, and derivative monitoring at hardware speed.

16. Discussion and Future Work

This work endeavors to describe features of a DLCA such that a person can decide for themselves the efficacy of the approach or to inspire others to come up with further improved approaches. The work uses a lemma forward method aligning lemmas with desirable features then developing how to meet these features. It is concluded many key elements have been demonstrated separately and results can be adapted for use in DLCA.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

The goal is not to replace the current crop of LLM's but to enhance the utility of LLMs into higher value applications where the deficiency of current LLM's represent a development barrier. In addition the logical core approach is likely to marginally improve LLM functionality through processing speed, audit-ability and improved efficiency.

Companion papers have been proposed which will extend the understanding of DLCA.

Considering social goals early and meeting social goals through architecture can assure the approach does not drift off track at some time in the future.

16.1 Photonic LC — Long-Term Hardware Evolution

Photonic implementation of the LC's combinational logic is identified as a long-term hardware evolution path. Creating potential for the immutability principle (L5) to be strengthened by the absence of charge-based state. This is a topic deferred to future work.

17. Conclusion

This paper has presented the Deterministic Logical Core Architecture (DLCA), a formally specified, hardware-governed framework for AI systems requiring auditable reasoning.

The architectural principals disclosed are intended to achieve the following features:

a) measurable precision

b) logic operation reproducibility

c) auditability

d) efficiency gains over transformer LLMs

e) Cultural and language specificity.

f) Human-in-the-loop control.

Three companion papers are planned for concurrent or subsequent submission. These plans are conceptual at this stage and will evolve as the work progresses. The work described for the companion papers may not be in sequential order listed below.

The first companion paper, DLCA: Formal Verification and Vocabulary Formalisation will design and execute the logical verification tests possible at this stage and will formalise pilot lemmatic vocabulary and cultural specifications. The second, DLCA: Empirical Test Methodology, Prototype Benchmarking, and Zone Classifier Design will present formal hardware test designs, proxy benchmarking of PPU performance against commercial systems, a detailed Zone Classifier design specification, and the empirical test infrastructure plan for when hardware prototypes become available. The third, DLCA: Marketplace Model, Regulatory Economics, and Intellectual-Property Framework will develop the three-tier component marketplace, the server-based cost model and its assumptions, and the intellectual-property treatment. Hardware-dependent empirical measurements are designated as future work contingent on prototype

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

availability. Detailed treatment of regulatory economics, including cost-avoidance estimates for EU AI Act Article 13/17 compliance, is reserved for the third companion paper.

Acknowledgments

This paper was prepared using an AI-assisted methodology in an assistant, checking and research role. Document research and initial draft text were generated using Claude (Anthropic), with subsequent substantial revision and rewriting by the author. Iterative review passes were conducted using ChatGPT (OpenAI), Copilot (Microsoft), Perplexity AI and author. All architectural decisions, lemmatic claims, and intellectual contributions are the work of the sole author. The methodology follows a lemma-first derivation structure, All architectural conclusions are derived from the fourteen foundational lemmas, with each derivation mapped to a design characteristic.

No external funding was received for this work.

[1] Pan L. ,. Albalak, A. ,. Wang, X. and W. Y. Wang, “Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. Findings of the Association for Computational Linguistics: EMNLP 2023, 3806–3824. Association for Computational Linguistics, Singapore. doi:10.18653/v1/2023.findings-emnlp.248 https://aclanthology.org/2023.findings-emnlp.248/,” 2023, p. 248. doi: 10.18653/v1/2023.findings-emnlp.248.

[2] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada: ACM, Mar. 2021, pp. 610–623. doi: 10.1145/3442188.3445922.

[3] R. Ji Z. ,. Lee, N. ,. Frieske and others, “Survey of Hallucination in Natural Language Generation,” *ACM Comput. Surv.*, vol. 55, no. 2023, p. 248, 2023, doi: 10.1145/3571730.

[4] W. Huang L. ,. Yu, W. ,. Ma and others, “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions,” *ACM Trans. Inf. Syst.*, no. 2025, 2025, doi: 10.1145/3703155.

[5] A. Lewis P. ,. Perez, E. ,. Piktus and others, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 2020. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html,” 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[6] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24824–24837, Dec. 2022.

[7] Y. Bai *et al.*, “Constitutional AI: Harmlessness from AI Feedback,” Dec. 15, 2022, *arXiv*: arXiv:2212.08073. doi: 10.48550/arXiv.2212.08073.

[8] N. Kirkpatrick J. ,. Pascanu, R. ,. Rabinowitz and others, “Overcoming Catastrophic Forgetting in Neural Networks,” vol. 114, no. 2017, 2017, doi: 10.1073/pnas.1611835114.

[9] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[10] Z. C. Lipton, “The Mythos of Model Interpretability,” vol. 16, no. 2018, 2018, doi: 10.1145/3236386.3241340.
[11] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, “Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection,” in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, in AISec ’23. New York, NY, USA: Association for Computing Machinery, Nov. 2023, pp. 79–90. doi: 10.1145/3605764.3623985.
[12] X. Qi *et al.*, “Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To!,” presented at the The Twelfth International Conference on Learning Representations, Oct. 2023. Accessed: Jun. 22, 2026. [Online]. Available: https://openreview.net/forum?id=hTEGyKf0dZ
[13] B. Shneiderman, “Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems,” *Bridg. Gap Ethics Pract. Guidel. Reliab.*, vol. 10, no. 2020, p. 26, 2020, doi: 10.1145/3419764.
[14] ISO, “ISO 26262: Road Vehicles — Functional Safety. International Organization for Standardization, Geneva,” ISO, 2011.
[15] T. Han, Z. Wang, C. Fang, S. Zhao, S. Ma, and Z. Chen, “Token-Budget-Aware LLM Reasoning,” in *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 24842–24855. doi: 10.18653/v1/2025.findings-acl.1274.
[16] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*, vol. 119. 2016. Accessed: Jun. 22, 2026. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj
[17] Wachter S. ,. Mittelstadt, B. and C. Russell, “Counterfactual Explanations Without Opening the Black Box,” *Harv. J. Law Technol.*, vol. 31, no. 2017, 2017, doi: 10.2139/ssrn.3063289.
[18] L. Floridi *et al.*, “AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations,” *Minds Mach.*, vol. 28, no. 4, pp. 689–707, Dec. 2018, doi: 10.1007/s11023-018-9482-5.
[19] L. De Moura and N. Bjørner, “Z3: An Efficient SMT Solver,” in *Tools and Algorithms for the Construction and Analysis of Systems*, vol. 4963, C. R. Ramakrishnan and J. Rehof, Eds., in Lecture Notes in Computer Science, vol. 4963. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 337–340. doi: 10.1007/978-3-540-78800-3_24.
[20] Een N. and N. Sorensson, “An Extensible SAT-Solver,” in *SAT 2003, LNCS 2919, pp*, 2004, pp. 502–518.
[21] M. A. Musen, “The protégé project: a look back and a look forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015, doi: 10.1145/2757001.2757003.
[22] Garcez A. d. ,. Gori, M. ,. Lamb, L. C. ,. Serafini, L. ,. Spranger, M. and S. N. Tran, “Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning,” *J. Appl. Log. — IfCoLog J.*, vol. 6, no. 2019, 2019, [Online]. Available: https://arxiv.org/abs/1905.06088
[23] Manhaeve R. ,. Dumancic, S. ,. Kimmig, A. ,. Demeester, T. and L. De Raedt, “DeepProbLog: Neural Probabilistic Logic Programming. Advances in Neural Information Processing Systems (NeurIPS 2018), 31,” 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/dc5d637ed5e62c36ecb73b654b05ba2a-Abstract.html
[24] M. Richardson and P. Domingos, “Markov logic networks,” *Mach. Learn.*, vol. 62, no. 1–2, pp. 107–136, Feb. 2006, doi: 10.1007/s10994-006-5833-1.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[25] I. Graves A. ,. Wayne, G. ,. Reynolds, M. ,. Harley, T. ,. Danihelka and others, “Hybrid Computing Using a Neural Network with Dynamic External Memory,” vol. 538, no. 2016, 2016, doi: 10.1038/nature20101.
[26] K. Shazeer N. ,. Mirhoseini, A. ,. Maziarz and others, “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” in *ICLR 2017*, 2017. [Online]. Available: https://openreview.net/forum?id=B1ckMDqlg
[27] P. Hu E. J. ,. Shen, Y. ,. Wallis and others, “LoRA: Low-Rank Adaptation of Large Language Models,” in *ICLR 2022*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9
[28] E. H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*. Elsevier, 1976. [Online]. Available: https://www.worldcat.org/oclc/2309230
[29] J. McDermott, “R1: A rule-based configurer of computer systems,” *Artif. Intell.*, vol. 19, no. 1, pp. 39–88, Sep. 1982, doi: 10.1016/0004-3702(82)90021-2.
[30] Shu-Hsien Liao, “Expert system methodologies and applications—a decade review from 1995 to 2004,” *Expert Syst. Appl.*, vol. 28, no. 1, pp. 93–103, Jan. 2005, doi: 10.1016/j.eswa.2004.08.003.
[31] C. L. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artif. Intell.*, vol. 19, no. 1, pp. 17–37, Sep. 1982, doi: 10.1016/0004-3702(82)90020-0.
[32] “A configurable Rete-OO engine for reasoning with different types of imperfect information,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. c2–c2, Oct. 2009, doi: 10.1109/TKDE.2009.178.
[33] E. M. Clarke, E. A. Emerson, and J. Sifakis, “Model checking: algorithmic verification and debugging,” *Commun. ACM*, vol. 52, no. 11, pp. 74–84, Nov. 2009, doi: 10.1145/1592761.1592781.
[34] M. Kaufmann and J. Strother Moore, “ACL2: an industrial strength version of Nqthm,” in *Proceedings of 11th Annual Conference on Computer Assurance. COMPASS ’96*, Jun. 1996, pp. 23–34. doi: 10.1109/CMPASS.1996.507872.
[35] Y. Bertot and P. Castéran, *Interactive Theorem Proving and Program Development*. in Texts in Theoretical Computer Science An EATCS Series. Berlin, Heidelberg: Springer, 2004. doi: 10.1007/978-3-662-07964-5.
[36] T. Nipkow, M. Wenzel, and L. C. Paulson, Eds., “1. The Basics,” in *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, Berlin, Heidelberg: Springer, 2002, pp. 3–8. doi: 10.1007/3-540-45949-9_1.
[37] L. Lamport, “The temporal logic of actions,” *ACM Trans. Program. Lang. Syst. TOPLAS*, vol. 16, no. 3, pp. 872–923, May 1994, doi: 10.1145/177492.177726.
[38] G. J. Holzmann, “The model checker SPIN,” *IEEE Trans. Softw. Eng.*, vol. 23, no. 5, pp. 279–295, May 1997, doi: 10.1109/32.588521.
[39] Leucker M. and C. Schallhart, “A Brief Account of Runtime Verification,” vol. 78, no. 2009, p. 2008, 2009, doi: 10.1016/j.jlap.2008.08.004.
[40] Ribeiro M. T. ,. Singh, S. and C. Guestrin, “‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” in *KDD 2016, pp*, 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.
[41] Lundberg S. M. and S. I. Lee, “A Unified Approach to Interpreting Model Predictions. NeurIPS 2017, 30. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html,” 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[42] Y. S. Koh P. W. ,. Nguyen, T. ,. Tang and others, “Concept Bottleneck Models,” in *ICML 2020, PMLR 119, pp*, 2020, pp. 5338–5348. [Online]. Available: https://proceedings.mlr.press/v119/koh20a.html
[43] A. Conmy, A. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso, “Towards Automated Circuit Discovery for Mechanistic Interpretability,” *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 16318–16352, Dec. 2023.
[44] B. Goodman and S. Flaxman, “European Union Regulations on Algorithmic Decision Making and a ‘Right to Explanation,’” *AI Mag.*, vol. 38, no. 3, pp. 50–57, Sep. 2017, doi: 10.1609/aimag.v38i3.2741.
[45] Wu Q. ,. Bansal, G. ,. Zhang, J. ,. Wu, Y. ,. Zhang, S. ,. Zhu, E. ,. Li, B. ,. Jiang, L. ,. Zhang, X. and C. Wang, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation,” in *the First Conference on Language Modeling (COLM 2024)*, 2024. [Online]. Available: https://openreview.net/forum?id=tEAF9LBdgu
[46] Olausson T. ,. Gu, A. ,. Lipkin, B. ,. Zhang, C. ,. Solar-Lezama, A. ,. Tenenbaum, J. and R. Levy, “LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers,” in *the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2*, 2023. doi: 10.18653/v1/2023.emnlp-main.313.
[47] L. Wang *et al.*, “A survey on large language model based autonomous agents,” *Front. Comput. Sci.*, vol. 18, no. 6, p. 186345, Mar. 2024, doi: 10.1007/s11704-024-40231-1.
[48] S. Russell, D. Dewey, and M. Tegmark, “Research priorities for robust and beneficial artificial intelligence,” *AI Magazine*, vol. 36, no. 4, pp. 105-, 2015.
[49] Kocaman V. and D. Talby, “Spark NLP: Natural Language Understanding at Scale. Software Impacts, 8, 100058. doi:10.1016/j.simpa.2021.100058,” *Softw. Impacts*, no. 2021, 2021, doi: 10.1016/j.simpa.2021.100058.
[50] K. Kreimeyer *et al.*, “Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review,” *J. Biomed. Inform.*, vol. 73, pp. 14–29, Sep. 2017, doi: 10.1016/j.jbi.2017.07.012.
[51] Patel N. ,. Grewal, H. ,. Buddhavarapu, V. and G. Dhillon, “OpenEvidence: Enhancing Medical Student Clinical Rotations With AI but With Limitations. Cureus, 17(1), e76867. doi:10.7759/cureus.76867. PMC11787857,” vol. 17, no. 2025, 2025, doi: 10.7759/cureus.76867.
[52] V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. D. Manning, and D. E. Ho, “Hallucination -Free? Assessing the Reliability of Leading AI Legal Research Tools,” *J. Empir. Leg. Stud.*, vol. 22, no. 2, pp. 216–242, Jun. 2025, doi: 10.1111/jels.12413.
[53] B. Munir, M. Z. Abbasi, W. B. Wilson, and A. Colombo, “Evaluating AI in Legal Operations: A Comparative Analysis of Accuracy, Completeness, and Hallucinations in ChatGPT-4, Copilot, DeepSeek, Lexis+ AI, and Llama 3,” *Int. J. Leg. Inf.*, vol. 53, no. 2, pp. 103–114, Jun. 2025, doi: 10.1017/jli.2025.10052.
[54] N. Bernard, Y. Sagawa Jr, N. Bier, T. Lihoreau, L. Pazart, and T. Tannou, “Using artificial intelligence for systematic review: the example of elicit,” *BMC Med. Res. Methodol.*, vol. 25, no. 1, p. 75, Mar. 2025, doi: 10.1186/s12874-025-02528-y.
[55] O. E. Apata, O.-M. Kwok, and Y.-H. Lee, “The Use of Generative Artificial Intelligence (AI) in Academic Research: A Review of the Consensus App,” *Cureus*, vol. 17, no. 7, p. e87297, doi: 10.7759/cureus.87297.
[56] I. RTCA, “DO-178C: Software Considerations in Airborne Systems and Equipment Certification. RTCA, Inc., Washington, DC,” RTCA, 2011.
[57] A. E. E. Committee, “ARINC Specification 653P1-5: Avionics Application Software Standard Interface. Aeronautical Radio, Inc,” 2019.
[58] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, IEC 61508:2010, Geneva., 2010.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[59] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An Integrated Theory of the Mind,” *Psychol. Rev.*, vol. 111, no. 4, pp. 1036–1060, Oct. 2004, doi: 10.1037/0033-295X.111.4.1036.
[60] J. E. Laird, A. Newell, and P. S. Rosenbloom, “SOAR: An architecture for general intelligence,” *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, Sep. 1987, doi: 10.1016/0004-3702(87)90050-6.
[61] J. E. Laird, C. Lebiere, and P. S. Rosenbloom, “A Standard Model of the Mind: Toward a Common Computational Framework Across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics,” *AI Mag.*, vol. 38, no. 4, pp. 13–26, Dec. 2017, doi: 10.1609/aimag.v38i4.2744.
[62] R. Sun, P. Slusarz, and C. Terry, “The Interaction of the Explicit and the Implicit in Skill Learning: A Dual-Process Approach,” *Psychol. Rev.*, vol. 112, no. 1, pp. 159–192, Jan. 2005, doi: 10.1037/0033-295X.112.1.159.
[63] J. S. B. T. Evans, “Dual-Processing Accounts of Reasoning, Judgment, and Social Cognition,” *Annu. Rev. Psychol.*, vol. 59, no. Volume 59, 2008, pp. 255–278, Jan. 2008, doi: 10.1146/annurev.psych.59.103006.093629.
[64] M. Chen, “Trust, understanding, and machine translation: the task of translation and the responsibility of the translator,” *AI Soc.*, vol. 39, no. 5, pp. 2307–2319, Oct. 2024, doi: 10.1007/s00146-023-01681-6.
[65] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš, “From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, 2020, pp. 4483–4499. doi: 10.18653/v1/2020.emnlp-main.363.
[66] N. Evans and S. C. Levinson, “The myth of language universals: Language diversity and its importance for cognitive science,” *Behav. Brain Sci.*, vol. 32, no. 5, pp. 429–448, Oct. 2009, doi: 10.1017/S0140525X0999094X.
[67] Ball B. ,. Koliousis, A. ,. Mohanan, A. and M. Peacey, “Computational Philosophy: Reflections on the PolyGraphs Project. Humanities and Social Sciences Communications, 11(1), 1–12. doi:10.1057/s41599-024-02619-z https://arxiv.org/abs/2403.09014,” *Humanit. Soc. Sci. Commun.*, vol. 11, no. 2024, 2024, doi: 10.1057/s41599-024-02619-z.
[68] B. Hutchinson, “Modeling the Sacred: Considerations when Using Religious Texts in Natural Language Processing,” in *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 1029–1043. doi: 10.18653/v1/2024.findings-naacl.65.
[69] Kim J. M. ,. Choi, B. I. ,. Shin, H. P. and H.-J. Kim, “A Methodology for Constructing of Philosophy Ontology Based on Philosophical Texts. Computer Standards & Interfaces, 29(3), 328–337. doi:10.1016/j.csi.2006.05.002,” vol. 29, no. 2007, 2007, doi: 10.1016/j.csi.2006.05.002.
[70] C. Mayo-Wilson and K. J. S. Zollman, “The computational philosophy: simulation as a core philosophical method,” *Synthese*, vol. 199, no. 1, pp. 3647–3673, Dec. 2021, doi: 10.1007/s11229-020-02950-3.
[71] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, “OWL 2: The next step for OWL,” *J. Web Semant.*, vol. 6, no. 4, pp. 309–322, Nov. 2008, doi: 10.1016/j.websem.2008.05.001.
[72] Shvaiko P. and J. Euzenat, “Ontology Matching: State of the Art and Future Challenges,” vol. 25, no. 2013, 2013, doi: 10.1109/TKDE.2011.253.
[73] L. Boroditsky, “Does Language Shape Thought?: Mandarin and English Speakers’ Conceptions of Time,” *Cognit. Psychol.*, vol. 43, no. 1, pp. 1–22, Aug. 2001, doi: 10.1006/cogp.2001.0748.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[74] B. L. Worf, “Thought, and Reality: Selected Writings of Benjamin Lee Whorf,” MIT Press, 1956.
[75] E. Sapir, “The Status of Linguistics as a Science,” *Language*, vol. 5, no. 4-Part1, pp. 207–214, Dec. 1929, doi: 10.2307/409588.
[76] E. M. Ponti *et al.*, “Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing,” *Comput. Linguist.*, vol. 45, no. 3, pp. 559–601, Sep. 2019, doi: 10.1162/coli_a_00357.
[77] J. Rawls, *A theory of justice*, Rev. ed., 5.-6. printing. Cambridge, Mass: Belknap Press of Harvard Univ. Press, 2003.
[78] I. Kant, *Groundwork of the Metaphysics of Morals. (Translated by M. Gregor & J. Timmermann, Cambridge University Press, 2012.)*. Cambridge University Press, 1785. [Online]. Available: https://www.cs.ucdavis.edu/~koehl/Teaching/ECS088/PDF_files/AI_morality_Kant.pdf
[79] L. Confucius (c. 479 BCE). The Analects (Lunyu). Standard edition: Legge James (trans.)., *The Analects (Lunyu)*. Clarendon Press, Oxford, 1861. [Online]. Available: https://ctext.org/analects
[80] Hall D. L. and R. T. Ames, *Thinking Through Confucius. State University of New York Press, Albany*. Albany: State University of New York Press, 1987.
[81] D. Keown, *The Nature of Buddhist Ethics*. London: Palgrave Macmillan UK, 1992. doi: 10.1007/978-1-349-22092-2.
[82] W. B. Hallaq, “An Introduction to Islamic Law,” Cambridge Aspire website. Accessed: Jun. 22, 2026. [Online]. Available: https://www.cambridge.org/highereducation/books/an-introduction-to-islamiclaw/EFE6192467D2302522998BA33C34233D
[83] “Toward an African Moral Theory* - Metz - 2007 - Journal of Political Philosophy - Wiley Online Library.” Accessed: Jun. 15, 2026. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9760.2007.00280.x
[84] P. O. with the editorial assistance of S. Olivelle, *Manu’s Code of Law: A Critical Edition and Translation of the M-anava-Dharmaś-astra*. in South Asia Research. Oxford, New York: Oxford University Press, 2004.
[85] J. L. Garfield and G. Priest, “Nagarjuna and the Limits of Thought,” *Philos. East West*, vol. 53, no. 1, pp. 1–21, Jan. 2003, doi: 10.1353/pew.2003.0004.
[86] Herbert Fingarette, *Confucius: The Secular as Sacred*. 1998.
[87] P. Manakul, A. Liusie, and M. Gales, “SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore: Association for Computational Linguistics, 2023, pp. 9004–9017. doi: 10.18653/v1/2023.emnlp-main.557.
[88] Crosby S. A. and D. S. Wallach, “Efficient Data Structures for Tamper-Evident Logging,” in *the 18th USENIX Security Symposium (USENIX Security 2009), pp*, 2009, pp. 317–334.
[89] B. Laurie, A. Langley, and E. Kasper, “Certificate Transparency,” Internet Engineering Task Force, Request for Comments RFC 6962, Jun. 2013. doi: 10.17487/RFC6962.
[90] “ISA/IEC 62443 Series of Standards - ISA,” isa.org. Accessed: Jun. 22, 2026. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards
[91] A. Biere, “PicoSAT Essentials,” *J. Satisf. Boolean Model. Comput.*, vol. 4, no. 2–4, pp. 75–97, May 2008, doi: 10.3233/SAT190039.
[92] D. Nurvitadhi E. ,. Sim, J. ,. Sheffield and others, “Accelerating Recurrent Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC,” in *FPL 2016*, 2016.
[93] M. Blott *et al.*, “FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks,” *ACM Trans. Reconfigurable Technol. Syst. TRETS*, vol. 11, no. 3, p. 16:1-16:23, Dec. 2018, doi: 10.1145/3242897.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD

[94] Y. Umuroglu *et al.*, “FINN: A Framework for Fast, Scalable Binarized Neural Network Inference,” in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, in FPGA ’17. New York, NY, USA: Association for Computing Machinery, Feb. 2017, pp. 65–74. doi: 10.1145/3020078.3021744.

Received date: June 28 2026
Revised date: Sept. 14 2026
Accepted date: Sept. 17 2026

DOI: https://10.63211/j.p.26.204505
ISSN 2760-8646 (Online) ICAITD