

Reality Engineering for Governed Agent Systems: A Constitutional Runtime Architecture and Falsifiable Research Agenda

Muhammad Arif bin Fazil

arifOS Constitutional Kernel Project, Kuala Lumpur, Malaysia

github.com/ariffazil/arifOS

19 October 2026

Abstract

Agent systems can improve their prompts, tools, memory, and control code while remaining unable to determine whether their claims correspond to the external world or whether they are authorized to act. This position paper defines *Reality Engineering* as the design of interfaces that force machine claims and actions into auditable contact with evidence, authority, and observed consequences. It presents arifOS as a running case: a classical, model-agnostic constitutional runtime that places non-compensatory policy gates around agent execution. The architecture separates proposer, judge, executor, and witness roles; assigns every task an evidence floor; records predictions before outcomes; and retains falsifications rather than overwriting them. The paper does not claim artificial general intelligence (AGI), artificial superintelligence (ASI), quantum computation, or a physical law. Its defensible classification is an *agentic governance operating layer*: potentially useful for governing future AGI/ASI systems, but neither the intelligence substrate nor evidence that such systems have been achieved. Four falsifiable hypotheses and a preregisterable evaluation protocol are specified. A 19 October 2026 internal audit snapshot is disclosed as engineering telemetry, not efficacy evidence: 131 prediction records, 44 marked verified, 2 marked unverifiable, and 85 lacking either status in the exported aggregate. The central proposition is that increasingly self-modifying agents require a correctness and authority layer that they cannot silently rewrite.

Keywords: AI agents; governance kernel; agent operating system; evidence provenance; capability security; runtime assurance; human oversight; falsifiability

1 Introduction

The performance of an agent depends on more than model weights. A harness determines which context is shown, which tools are available, how failures are retried, what is remembered, and how outputs are checked. Recent systems optimize this surrounding code automatically. Self-Harness reports gains from a fixed model revising a bounded harness through weakness mining, proposal, and regression validation [1]. Meta-Harness treats harness code as a search object and gives a proposer access to prior source, scores, and execution traces [2]. Matched-budget evaluation, however, shows that harness evolution can fail to outperform simpler test-time scaling and may generalize weakly to held-out tasks [3]. These results expose a control problem: when the machinery that produces behavior can rewrite itself, what remains outside the optimization loop and who is authorized to change it?

This paper proposes a narrow answer: engineer explicit contact surfaces between agent claims and independently inspectable evidence, between proposed actions and authority, and between predictions and later outcomes. The term *Reality Engineering* has prior use in advertising research, where it described the blurring of commercial signification and popular culture [4]. No priority over the phrase is claimed. The present contribution is a distinct operational definition for agent infrastructure.

The paper makes three contributions. First, it gives a typed architecture for claims, evidence, authority, execution, outcomes, and falsification. Second, it resolves several apparent contradictions in the arifOS design by separating scientific, architectural, and metaphorical statements. Third, it converts the strongest proposed benefits into a falsifiable evaluation protocol rather than reporting unperformed experiments as results.

Scope. arifOS is analyzed as a running design and prototype, not as proof of a universal theory. The work is a position and systems-architecture paper. It reports a descriptive store snapshot but no controlled efficacy result, no independent replication, and no quantum computation.

2 Operational Definition

Definition 1 (Reality Engineering). Reality Engineering is the design and maintenance of mechanisms by which an agent’s claims and proposed actions are (i) bounded by their available evidence, (ii) checked against delegated authority before external effects, and (iii) corrected when witnessed outcomes contradict prior predictions.

The definition distinguishes six objects:

1. **Experience:** what an actor or observer underwent.
2. **Evidence:** a trace, measurement, document, or observation with provenance.
3. **Interpretation:** meaning assigned to evidence.
4. **Model:** a compressed account used to explain or predict.
5. **Prediction:** a time-stamped claim about an outcome not yet observed.
6. **Authority:** a revocable permission to mutate a specified resource under stated constraints.

Reality is not equated with the evidence store. Evidence is a fallible access channel. Consequently, “reality overrides the model” is implemented as a correction rule: sufficiently qualified counterevidence changes the system’s warranted claim status; it is not a claim of direct, infallible access to reality.

3 System Classification

The classification question is settled by necessary properties, not branding. Table 1 applies operational tests.

Table 1: Classification of arifOS under explicit criteria.

Candidate label	Required evidence and observed fit	Verdict
Quantum intelligence	Computation using controlled quantum states, quantum gates or measurements, with a specified quantum algorithm and hardware or simulator interface. arifOS currently supplies none of these. Quantum language in prior design texts is analogical.	VOID as a technical claim
AGI	Broad, transferable task competence across domains, validated by accepted tests and real-world performance. The kernel does not itself generate general competence.	VOID as an achieved capability
ASI	General competence exceeding humans across relevant cognitive domains, established by extraordinary evidence. No such evidence is presented.	VOID
AGI/ASI substrate	An execution or learning substrate that supplies the mechanisms from which general intelligence emerges. arifOS can wrap models but does not supply model weights, learning dynamics, or cognition.	HOLD / unsupported
AGI/ASI governance substrate	A model-agnostic layer for policy gates, authority, evidence, receipts, and human veto around capable agents. This matches the specified architecture, subject to implementation tests.	PLAUSIBLE / partial
Agentic OS	Runtime abstractions for agent identity, tasks, tools, memory, authority, audit, and lifecycle. arifOS implements or specifies several, but is not a hardware OS and lacks evidence of production maturity.	CLAIM as design; partial as runtime
OS for civilization	Multi-institutional standards, legitimacy, interoperability, adoption, and accountable operation at societal scale. These conditions are not established.	HYPOTHESIS / research direction

The scientifically conservative label is therefore **constitutional agent runtime** or **agentic governance OS**. “For civilization” states an intended deployment horizon, not a demonstrated property.

“Quantum intelligence” should not appear in the title, abstract, or claims unless a future implementation performs a specified computation through quantum information processing and reports comparative evidence.

4 Reality-Contact Architecture

Figure 1 shows the architecture as a sequence of typed gates. The language model is a proposer. It does not acquire authority from eloquence, confidence, benchmark score, access to a tool schema, or possession of evidence.

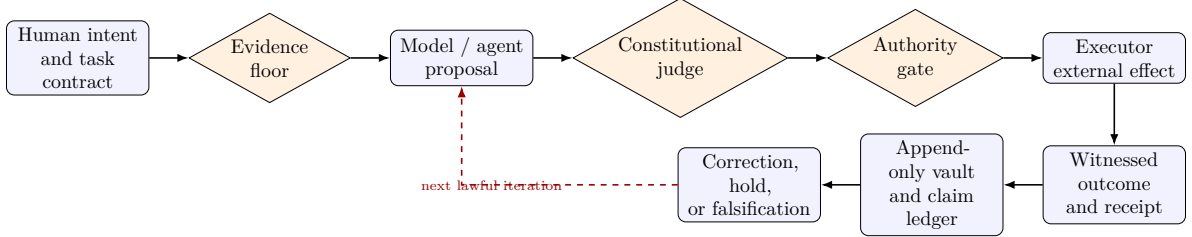


Figure 1: Reality-contact stack. Solid arrows are forward transitions; the dashed arrow is evidence-bearing correction. A failed gate produces no external mutation.

4.1 Four separated powers

The architecture separates four functions:

- **Proposer (Reflex):** generates candidate claims or actions.
- **Judge:** evaluates policy, evidence class, reversibility, and required authorization.
- **Executor:** performs an allowed effect through constrained tools.
- **Witness:** records what was requested, authorized, executed, and observed.

No single component should propose, authorize, execute, and certify the same consequential action. Separation does not guarantee truth: components may share a model family, data, or operator and therefore fail correlatively. It creates inspectable boundaries and permits stronger independence where risk warrants it.

4.2 Non-compensatory gates

A weighted score can conceal a catastrophic failure: excellent usefulness can numerically offset absent authority. arifOS therefore applies hard vetoes before any soft aggregation. Let a proposed transition be τ , hard predicates be $h_i(\tau) \in \{0, 1\}$, and soft dimensions be $z_j(\tau) \in (0, 1]$. The transition score is defined only when every hard predicate passes:

$$D(\tau) = \begin{cases} \left(\prod_{j=1}^m z_j(\tau)^{w_j} \right)^{1/\sum_j w_j}, & \text{if } \prod_i h_i(\tau) = 1, \\ \perp, & \text{otherwise.} \end{cases} \quad (1)$$

The geometric mean is retained as a design choice for soft dimensions because it penalizes low components and is scale-symmetric under the stated normalization. No uniqueness theorem is claimed here: such a theorem requires explicit axioms and proof. The previously used “Grand Equation” is therefore demoted from scientific law to configurable governance functional.

4.3 Verdict semantics

The runtime uses a restrictive lattice rather than a scalar approval probability:

$$\text{VOID} \prec 888\text{-HOLD} \prec \text{HOLD} \prec \text{SABAR} \prec \text{SEAL}.$$

VOID blocks a prohibited transition. 888-HOLD requires explicit human confirmation for irreversible or high-impact action. HOLD pauses for missing evidence or authority. SABAR permits continued bounded investigation without external commitment. SEAL authorizes only the scoped transition recorded in the receipt. A verdict is not a claim that the output is true; it states the transition’s governance status.

4.4 Evidence layers

Each task declares a required evidence floor and a current evidence layer. Figure 2 prevents inferred text from silently becoming verified state.

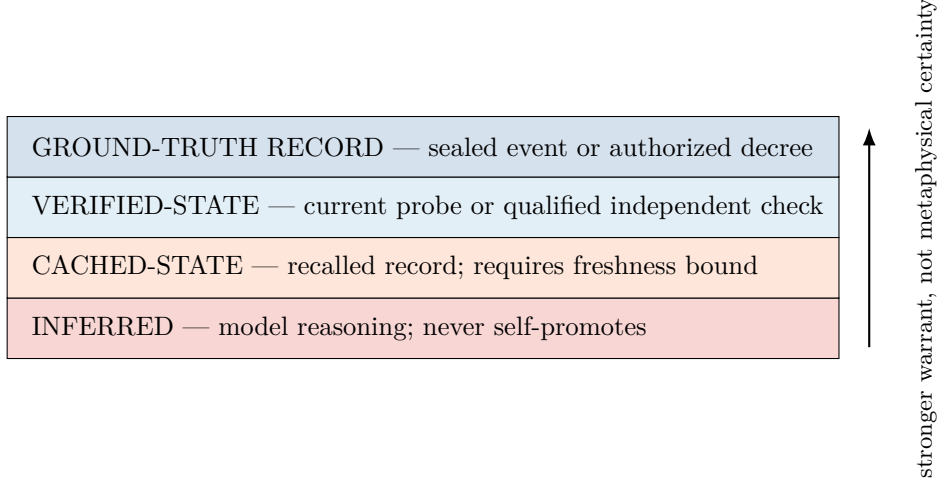


Figure 2: Evidence ladder. The top layer is an institutional record with chain integrity, not reality itself.

4.5 Reality Graph and temporal loop

The Reality Graph is a typed, versioned graph over identity, relationship, authority, consequence, and observed-world records. Claims may have **falsified-by** and **awaiting-witness** edges. Falsified claims remain addressable; deletion would erase learning and make calibration unverifiable. Authority edges are checked before mutation, not appended afterward as decorative audit metadata.

A temporal service freezes a prediction before its outcome window, then binds later observations to that prediction. The minimal record is

$$R = \langle id, t_p, C, E_p, W, t_o, O, Q, V \rangle, \quad (2)$$

where t_p is prediction time, C the claim, E_p evidence available at prediction time, W the outcome window, t_o observation time, O the observation, Q witness qualifications, and V the adjudicated result. Changes create successor records; they do not rewrite R .

5 Relation to Prior Work

The contribution is not a claim that governance, evaluation, provenance, or capability security is new. It is a synthesis that places them in a single pre-commit and post-outcome loop. Table 2 distinguishes nearby systems.

Table 2: Selected 2026 systems and the boundary addressed by this paper. Numbers are author-reported and not directly comparable across benchmarks.

Work	Reported contribution	Evidence source	reported by	Remaining boundary
Self-Harness [1]	Fixed models iteratively modify bounded harness surfaces	Improvement on all nine model-benchmark combinations; up to 132% relative gain		Evaluator and task split define “better”; external-world correctness remains separate
Meta-Harness [2]	Proposer searches code using prior sources, scores, and traces	7.7-point gain in online classification with 4× fewer context tokens		Search objective and authority remain externally supplied
Rethinking Harness Evaluation [3]	Matched-budget baselines and held-out transfer	No consistent Terminal-Bench advantage over simpler scaling; stronger results in adaptive games		Shows why optimization receipts do not themselves establish generalization
Agent libOS [5]	Capability-controlled runtime; visible affordance does not imply resource authority	27 deterministic tasks; all modelled unauthorized effects prevented; 7.0% conservative denial of allowed attempts		Deterministic artifact does not establish open-world semantic correctness
AgentEval [6]	Contract-bound assessment record across stateful execution	Real tool-world evaluation over permission, transaction, and clarification families		Assessment validity still depends on contracts and witness quality
EIO-Agents [7]	Typed evidence semantics and portable evaluation records	Open specification with evidence contracts, claims, witnesses, and proof status		Semantic interoperability does not grant action authority
Agent OS [8]	Derives candidate abstractions from classical and cloud operating systems	Position paper arguing for stable, portable agent abstractions		Portability semantics and constitutional legitimacy are distinct layers
Agent memory survey [9]	Taxonomy of internal/external substrate, cognitive mechanism, and subject	Survey accepted by TMLR with survey certification		Memory utility does not make remembered content true

6 Disclosed Internal Data

The original draft mentioned an earlier circulation of “27 correct, 6 wrong, 4 mixed” outcomes. A 19 October 2026 aggregate probe instead returned 131 prediction records, 44 marked verified, and 2 marked unverifiable. The circulated tuple does not reconcile with that aggregate and is withdrawn as an efficacy result.

The residual category is computed only as $131 - 44 - 2 = 85$. It may combine pending, unevaluated, differently labelled, or malformed records; without the row-level export it must not be interpreted as incorrect. Figure 3 reports counts, not accuracy. The included CSV records the provenance and restriction.

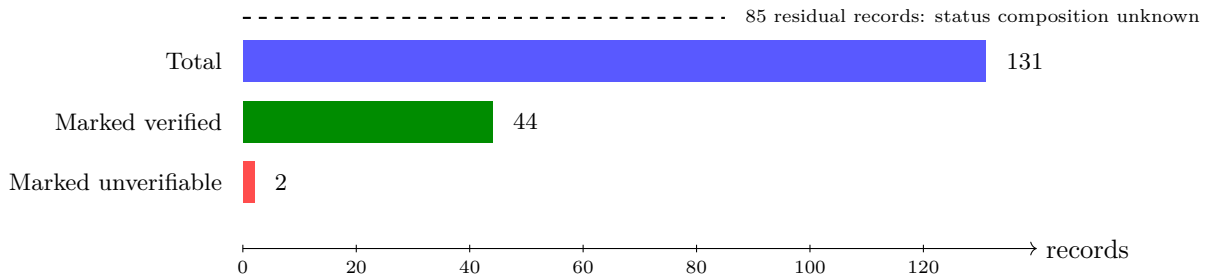


Figure 3: Internal CHRON aggregate snapshot, 19 October 2026. This is operational telemetry from a single system, not an independently verified performance result.

Table 3: Permitted and prohibited interpretations of the snapshot.

Status	Interpretation
Permitted	A store probe reported the three aggregate values at one time; 44/131 equals 33.6% of records marked verified, but this is a workflow coverage ratio, not predictive accuracy.
Prohibited	“44 correct out of 131,” “98.5% valid,” or any other accuracy/reliability claim without record-level denominators, adjudication rules, and independent witnesses.
Required next artifact	Immutable row-level export containing prediction time, outcome window, evidence-at-prediction, observation, witness, adjudication rule, status, and content hash.

7 Paradoxes and Dispositions

The following are not all logical paradoxes. Most are design tensions produced by collapsing different layers. Their dispositions are explicit.

Table 4: Resolution register. “Resolved” means the architecture removes a contradiction; it does not mean empirical success has been shown.

Tension	Failure from conflation	Disposition	Status
Reality vs. evidence	Treating a stored observation as reality itself	Evidence is a provenance-bearing, defeasible access channel; contradictory qualified evidence triggers re-adjudication	Conceptually resolved
Self-improvement vs. self-governance	An optimizer rewrites its own acceptance criteria	Methods may evolve; constitutional root, judge policy, identity, and append-only history require separately authorized change	Architecturally resolved; tests needed
Capability vs. authority	A visible or executable tool becomes permission	Resource authority is checked at the effect boundary through scoped, revocable grants	Architecturally resolved
Author vs. certifier	A system declares its own output valid	Separate processes and records; disclose shared-model or shared-data dependence; require external witness for high-risk claims	Mitigated, never absolute
Held-out evaluation vs. adaptive search	Repeated optimization leaks the test set	Three partitions: development, promotion, and private audit; precommit budgets and rotate private audit only after sealing a release	Protocol resolved; leakage remains a risk
Memory vs. truth	Recalled text silently becomes verified state	Memory is cached claim material with freshness and provenance, never self-authenticating truth	Resolved by typing
Geometric score vs. hard rights	High utility offsets absent consent or safety	Apply hard vetoes before Equation 1; compute soft score only on lawful candidates	Resolved
Human sovereignty vs. attention	Every decision waits for one person	Delegate reversible low-risk scopes, batch reviews, rate-limit escalation, and reserve 888-HOLD for irreversible or high-impact actions	Open systems problem
Transparency vs. attack surface	Full disclosure reveals operational secrets	Publish invariants, schemas, and testable semantics; protect credentials, live policies, and exploitable details under least privilege	Risk-tiered resolution
Classical code vs. quantum language	Metaphors are mistaken for quantum computation	Remove physical claims; use “quantum-inspired” only when a mapped algorithmic contribution is specified, and never as evidence of quantum advantage	Resolved by rejection
Governance layer vs. AGI	Safety architecture is marketed as intelligence	Treat model capability and governance as orthogonal axes; arifOS can govern an AGI without being one	Resolved by classification
Civilizational intent vs. evidence	Aspirational scale is reported as achieved scope	Treat civilizational deployment as a research horizon requiring multi-institutional adoption and legitimacy evidence	Open

No finite architecture resolves the philosophical problem of access to reality, correlated failures among supposedly independent judges, or the legitimacy of governance at civilizational scale. The correct resolution is not rhetorical closure but typed uncertainty, precommitted tests, and retained counterevidence.

8 Falsifiable Research Programme

Four hypotheses translate the architecture into experiments.

H1: Silent-drift reduction. On matched long-horizon tasks with injected state changes, a stack with frozen predictions, outcome binding, and retained falsifications will have a lower rate of false states that persist unchallenged than a score-only agent stack.

H2: Recovery latency. After injection of a false belief into memory, a provenance-typed stack will require fewer action steps and less wall-clock time to restore the correct operational state than an untyped conversational-memory baseline.

H3: Authority containment. Under adversarial tool descriptions and prompt injection, a pre-commit capability gate will produce fewer unauthorized external effects than wrapper-only tool controls, while false denials are separately reported.

H4: Goodhart displacement. Evidence floors will reduce unsupported high-confidence claims but may increase evidence fabrication attempts. Confirmation requires a shift in the measured failure distribution, not merely a higher aggregate score.

8.1 Preregisterable protocol

A minimum controlled study should use multiple model families, blinded adjudication, and at least three seeds per condition. Tasks should include software environments, structured enterprise workflows, and one non-digital domain with delayed observations. Conditions are: (A) ordinary harness, (B) harness with post-hoc logging, (C) pre-commit authority and evidence gates, and (D) full prediction–outcome loop. All conditions receive matched inference, tool, and human-review budgets.

Primary endpoints are silent-drift rate, unauthorized-effect rate, false-denial rate, median correction latency, calibration error, and receipt completeness. Secondary endpoints are task success, cost, latency, escalation burden, and evidence-fabrication attempts. Promotion data and the final audit set must be isolated; a candidate is evaluated once on the private audit set after design freeze. The protocol should publish schemas, hashes, exclusions, and negative results.

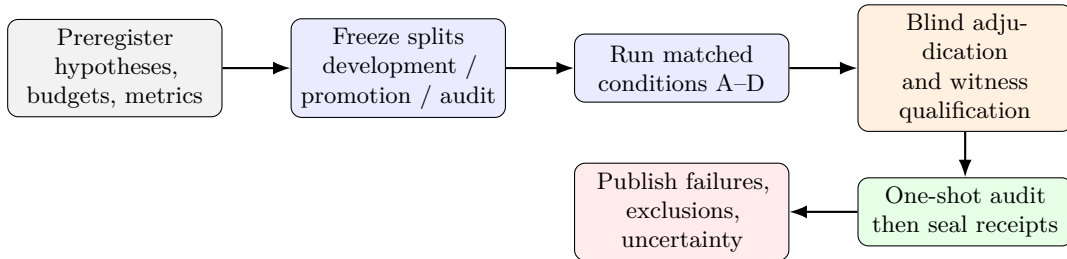


Figure 4: Evaluation flow designed to separate system development from the evidence used to claim generalization.

9 Threats to Validity

The architecture is derived from one project and reflects its terminology. Independent implementations may expose hidden coupling or unnecessary mechanisms. The telemetry snapshot is not independently verified and lacks row-level evidence in this submission. The proposed experiments have not been run. A human veto can fail through overload, capture, misunderstanding, or coercion; therefore “human in the loop” is not by itself a safety proof. Cryptographic integrity can show that a record was not changed after sealing, but cannot show that the original observation was accurate. Finally, formal policy checks prove only properties encoded in the policy and implementation, not every socially relevant value.

10 Implementation and Reproducibility

A reproducible release should include: (i) machine-readable task and receipt schemas; (ii) policy fixtures for every hard gate; (iii) property-based tests proving that failed authority checks cannot reach executor primitives; (iv) an append-only prediction export with hashes; (v) a claim ledger mapping every paper

statement to code, test, observation, derivation, or hypothesis; and (vi) a public threat model. Code and data links must resolve before submission.

The companion package for this manuscript includes the TeX source and two small CSV files: the disclosed aggregate snapshot and a claim-status ledger. These files document what the paper can currently support; they do not substitute for the row-level experimental artifact required by the protocol.

11 Conclusion

The defensible identity of arifOS is neither quantum intelligence nor demonstrated AGI/ASI. It is a classical constitutional runtime for governing agent claims and actions, with a plausible role as an AGI/ASI governance substrate if its enforcement properties survive independent testing. Its civilizational scope remains an ambition requiring institutional legitimacy, interoperability, and deployment evidence.

Reality Engineering names the narrower engineering problem: make claims answerable to evidence, actions answerable to authority, and predictions answerable to outcomes. The scientific contribution will stand or fall not on the vocabulary, equations, or manifesto, but on preregistered comparisons, independently inspectable receipts, and the system’s willingness to preserve evidence that falsifies it.

Ethics and Governance Statement

The architecture reserves irreversible or high-impact actions for explicit human authorization and treats human dignity, lawful authority, and reversibility as hard constraints. This does not transfer responsibility from deployers to software. Institutions adopting such a kernel remain responsible for lawful policy, representative governance, security, monitoring, and remedies for affected persons.

Data Availability

The aggregate CSV included with this manuscript contains no personal data. A row-level CHRON export was not available for this revision and is therefore not claimed. The cited external results are available in their respective papers and repositories.

Author Contributions

Muhammad Arif bin Fazil: conceptualization, system architecture, methodology, software direction, investigation, and writing. AI systems assisted with drafting, literature retrieval, consistency checking, and typesetting under human direction; they are not authors and hold no decision authority.

References

- [1] H. Zhang, S. Zhang, K. Li, C. Zhang, Y. Chen, Y. Zhang, L. Bai, and S. Hu, “Self-Harness: Harnesses That Improve Themselves,” *arXiv preprint arXiv:2606.09498*, 2026. <https://arxiv.org/abs/2606.09498>
- [2] Y. Lee, R. Nair, Q. Zhang, K. Lee, O. Khattab, and C. Finn, “Meta-Harness: End-to-End Optimization of Model Harnesses,” *arXiv preprint arXiv:2603.28052*, 2026. <https://arxiv.org/abs/2603.28052>
- [3] Y. Wang, H. Zhu, Z. Hu, Y. Yuan, Z. Chen, S. Senthil, H. Hajishirzi, Y. Tsvetkov, P. Dasigi, and T. Xiao, “Rethinking the Evaluation of Harness Evolution for Agents,” *arXiv preprint arXiv:2607.12227*, 2026. <https://arxiv.org/abs/2607.12227>
- [4] M. R. Solomon and B. G. Englis, “Reality Engineering: Blurring the Boundaries between Commercial Signification and Popular Culture,” *Journal of Current Issues & Research in Advertising*, vol. 16, no. 2, pp. 1–17, 1994. doi:10.1080/10641734.1994.10505015.
- [5] Y. Zhang, “Agent libOS: A Runtime Substrate for Capability-Controlled Self-Evolving LLM Agents,” *arXiv preprint arXiv:2606.03895*, 2026. <https://arxiv.org/abs/2606.03895>

- [6] R. Gao, “AgentEval: Evidence-Bounded Assessment of Stateful Agents,” *arXiv preprint arXiv:2610.21177*, 2026. <https://arxiv.org/abs/2610.21177>
- [7] F. Bousetouane, “EIO-Agents: The Missing Semantic Layer for AI Agent Evaluation,” *arXiv preprint arXiv:2610.07675*, 2026. <https://arxiv.org/abs/2610.07675>
- [8] G. Steinder and H. Franke, “Towards an Agent Operating System—Lessons from Classical and Cloud OS,” *arXiv preprint arXiv:2607.25076*, 2026. <https://arxiv.org/abs/2607.25076>
- [9] W.-C. Huang et al., “A Survey of Agent Memory in the Second Half: Towards Self-Evolving and Long-Horizon Agents,” *Transactions on Machine Learning Research*, 2026. <https://arxiv.org/abs/2602.06052>
- [10] J. H. Saltzer and M. D. Schroeder, “The Protection of Information in Computer Systems,” *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.
- [11] B. W. Lampson, “Protection,” *ACM SIGOPS Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.
- [12] C. A. E. Goodhart, “Problems of Monetary Management: The U.K. Experience,” in *Monetary Theory and Practice*, Macmillan, 1984.
- [13] G. W. Brier, “Verification of Forecasts Expressed in Terms of Probability,” *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.
- [14] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, 2023. doi:10.6028/NIST.AI.100-1.
- [15] National Institute of Standards and Technology, *Secure Software Development Framework (SSDF) Version 1.1*, NIST SP 800-218, 2022. doi:10.6028/NIST.SP.800-218.
- [16] J. Biamonte et al., “Quantum Machine Learning,” *Nature*, vol. 549, pp. 195–202, 2017. doi:10.1038/nature23474.

A Claim Status Ledger

Claim	Status	Required upgrade path
arifOS is a constitutional agent runtime	CLAIM as architecture	Public interfaces, test coverage, versioned releases, independent reproduction
arifOS reduces silent drift	HYPOTHESIS	Controlled H1 experiment
arifOS prevents unauthorized effects	HYPOTHESIS for open-world use	Property tests, adversarial evaluation, independent security audit
arifOS is quantum intelligence	VOID	New quantum algorithm and execution evidence would define a different claim
arifOS is AGI/ASI	VOID	Broad capability evidence; governance alone is insufficient
arifOS can govern AGI/ASI	PLAUSIBLE	Scale tests, threat model, model-independent enforcement, institutional evaluation
arifOS is an OS for civilization	HYPOTHESIS	Multi-institutional legitimacy, standards adoption, reliability and remedy evidence
Reality Engineering improves correctability	HYPOTHESIS	H1–H4 preregistered and independently replicated

B Submission Checklist

The manuscript uses a named author, complete references, single-spaced 10-point text, one-inch margins, vector figures, and machine-readable text. Before arXiv upload, the author should verify all repository URLs, add an ORCID if desired, select the appropriate license, compile using arXiv’s TeX Live environment, inspect the generated PDF, and remove unused files. The TeX source—not a TeX-generated PDF alone—should be uploaded with the included data files.