

From Semantic OS Simulation to Interoperable Knowledge Infrastructure: Experimental Findings and the Semantic Knowledge Interchange (SKI) v0.1 Proposal

Hirofumi Inomata
Independent Researcher
https://github.com/digiponta/SEMOS_SIM

October 18, 2026 — Working Paper / Independent Proposal

Abstract

We investigate a semantic-centric computing architecture in which knowledge, context, intent, provenance, and dependency relationships become explicit objects of operating-system-level management. A sequence of prototype and synthetic simulation studies within the Quantum-like Hybrid Architecture (QHA), Semantic OS, and SEMOS_SIM projects examines semantic reuse, incremental execution, distributed semantic processing, and quantum-inspired candidate selection. Under specific reported simulation settings, semantic processing reduced a normalized cost from 1,000 to 660 (34%); distributed semantic processing reduced aggregate cost from 604.0 to 452.9 (25.0%) and makespan from 209.0 to 162.9 (22.1%); and incremental dependency traversal reduced recomputation from 60 to six nodes (90% fewer affected nodes processed). These observations are scenario-specific and do not establish unconditional superiority over functionally matched conventional systems. Quantum-inspired complex-amplitude methods and ideal/noisy quantum-state simulations exhibited phase-sensitive behavior but did not consistently outperform classical kernel-based search. Building on these findings, we propose *Semantic Knowledge Interchange* (SKI) v0.1, a model-independent exchange profile for assertions, canonical claims, applicability conditions, provenance, epistemic assessments, lifecycle events, and synchronization metadata. SKI is intended to complement RDF, JSON-LD, PROV-O, SHACL, and transport protocols rather than replace them. We define architectural principles, a preliminary data model, exchange contracts, conformance objectives, security requirements, and a staged implementation agenda. The central hypothesis is that portable, auditable knowledge may be a more generalizable outcome of Semantic OS research than quantum-inspired acceleration alone.

Keywords: Semantic OS; semantic memory; knowledge interoperability; Semantic Processing Unit; distributed knowledge; incremental computation; JSON-LD; provenance; SKI; quantum-inspired search.

1 Introduction

Operating systems typically expose processes, threads, address spaces, files, and devices as primary managed resources. AI-oriented systems add retrieval indexes, language models, vector stores, knowledge graphs, workflow graphs, and agent memory, yet many of these resources remain application-local and cannot safely travel between independently implemented systems. In the QHA/Semantic OS research program, a *semantic processing unit* (SPU) is conceptualized as a unit of work driven by semantic identity, context, intent, dependencies, and execution provenance rather than by a single fixed model representation.

Two distinct questions motivate this paper. First, can explicit semantic management support useful execution optimizations, such as safe reuse, selective recomputation, and distributed coordination? Second, can knowledge produced or validated by one AI system be exchanged with another without silently losing conditions, evidence, trust boundaries, or revision history? We address the first with a synthesis of reported simulation results and the second with the proposed Semantic Knowledge Interchange (SKI) v0.1 specification. The proposal is a *working draft*, not a W3C, ISO, or national standard.

Contributions. We (i) consolidate simulation results and clarify their evidential limits; (ii) separate semantic execution benefits from quantum-inspired heuristics; (iii) identify the missing representation contract

between heterogeneous semantic-memory implementations; (iv) specify an initial interoperable assertion and event model; and (v) define measurable conformance and benchmark criteria for future independent implementations.

2 Research Architecture and Experimental Program

2.1 Semantic execution model

A semantic assertion or request is represented conceptually by

$$K = (\text{id}, c, x, \kappa, p, e, \ell, v), \quad (1)$$

where c is a canonical claim, x context, κ applicability conditions, p provenance, e epistemic assessment(s), ℓ lifecycle state, and v revision. This tuple is a conceptual *proposal*, not a statement that all historical simulations shared an identical serialized record.

The Semantic OS control plane can route a request to an SPU, search a memory for relevant assertions, check dependencies and policies, reuse an earlier result, or dispatch execution to local/distributed workers. The long-term QHA plan includes orchestrated LLM_SEM instances on multiple VMs with shared Semantic Memory and staged internalization via a `/sleep`-like workflow. Shared external knowledge and locally internalized model weights must be treated as different states: synchronization of assertions does not imply synchronized neural parameters.

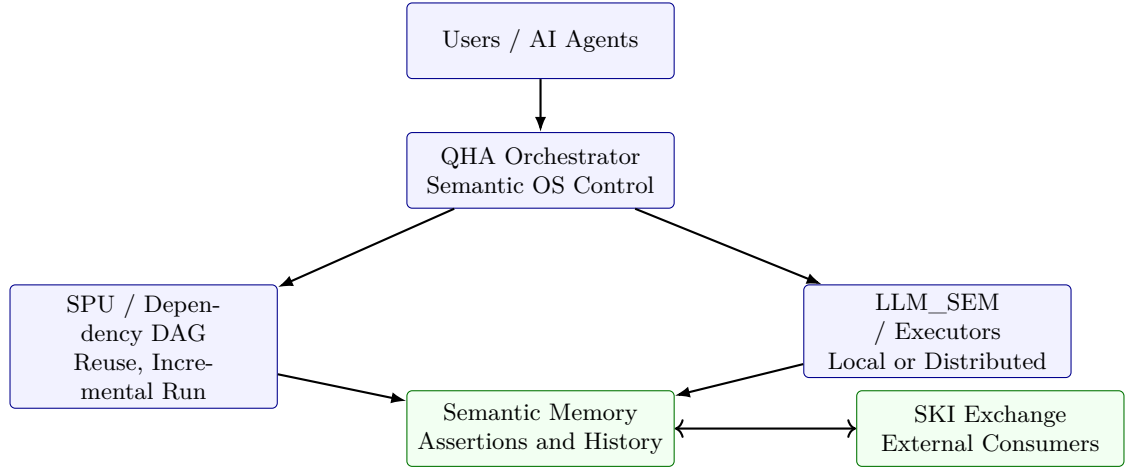


Figure 1: Conceptual architecture. SKI is a proposed interoperability boundary; existing prototype components vary in maturity and may simulate rather than execute actual virtual machines or LLM weight updates.

2.2 Experimental evidence categories

We distinguish (A) semantic execution cost simulations, (B) distributed coordination and incremental-recomputation simulations, (C) candidate selection studies on synthetic utility landscapes, and (D) ideal/noisy quantum-state emulation. These have different objectives and metrics, and their percentages should not be aggregated into a single system-level performance score.

3 Reported Experimental Results

3.1 Semantic reuse, SPU distribution, and incremental execution

Table 1 reports selected values in the research record. “Cost” is a simulation-defined proxy and not necessarily elapsed time, energy, or monetary expenditure. No public independently reproduced benchmark is claimed here.

The incremental scenario illustrates a direct opportunity: if an execution graph $G = (V, E)$ has a changed node set Δ , a correct invalidation algorithm executes only the affected closure

$$R(\Delta) = \Delta \cup \{v \in V \mid \exists u \in \Delta, u \rightsquigarrow v\}. \quad (2)$$

Table 1: Selected SEMOS_SIM measurements (scenario-specific, as reported).

Study / configuration	Reference outcome	Semantic or incremental outcome	Interpretation and boundary
SPU basic comparison (v0.7.2)	Normalized cost 1000	Cost 660	34% lower proxy cost; baseline feature parity and scaling require further audit.
Distributed (v0.7.4)	Aggregate cost 604.0	Aggregate cost 452.9	25.0% reduction within simulated workload.
Distributed (v0.7.4)	Makespan 209.0	Makespan 162.9	22.1% shorter simulated completion time.
Dependency recomputation (v0.7.9)	60 nodes reprocessed	6 nodes reprocessed	90% fewer recomputed nodes after the tested change; not a general time saving.
Distributed resilience (v0.7.x)	Multiple failure scenarios	Conditional savings reported	Recovery and consistency studies require explicit comparable failure distributions and complete measurement protocol.

The 60-to-six result characterizes the selected graph/change pair; it does not prove that semantic handling is uniquely necessary, since conventional build systems and incremental DAG engines also exploit such closure computations. A stronger benchmark must compare systems with *matched recognition and invalidation capabilities*.

3.2 Quantum-inspired search: negative and conditional findings

Initial interference-style candidate searches did not yield a stable gain over classical baselines. Later experiments added semantic-similarity-weighted complex amplitudes, independent synthetic holdouts, phase ablations, and routing controls. Some low-budget multimodal settings benefited from particular phase configurations, whereas Kernel-UCB often dominated when more evaluations were available. For instance, in the matched one-step v0.7.22 benchmark with 16 candidates, four common pilot observations, one additional oracle call, and 200 trials per landscape, Table 2 summarizes final optimum-discovery rates.

Table 2: Matched one-step final success probability (v0.7.22); higher is better.

Synthetic landscape	Random	Kernel-UCB	Kernel-Softmax	Phase + diffusion ($r = 1$)
Smooth	0.290	0.495	0.340	0.345
Shuffled	0.375	0.375	0.350	0.385
Multimodal	0.390	0.385	0.395	0.360

For smooth inputs, the phase/diffusion method was 15.0 percentage points below Kernel-UCB; for shuffled inputs it was one point above, with uncertainty spanning zero in the exploratory paired bootstrap. Phase-only gates changed amplitudes’ phases but not Born measurement probabilities, as expected. The quantum-inspired search ran in a *classical* simulator and, in the smooth case, incurred approximately 0.0446 ms median selection time versus 0.0231 ms for Kernel-UCB. Equal candidate oracle counts do not imply equal CPU time. These results do not establish physical or algorithmic quantum advantage.

3.3 Routing and noisy quantum-state studies

An adaptive-search experiment (v0.7.20) classified three synthetic landscape types using four observed pilot candidates. Of 360 independent holdout cases (120 per type), its confusion matrix was

$$C = \begin{bmatrix} 85 & 30 & 5 \\ 42 & 67 & 11 \\ 62 & 49 & 9 \end{bmatrix}, \quad \text{accuracy} = \frac{85 + 67 + 9}{360} \approx 44.7\%. \quad (3)$$

The multimodal recall was only $9/120 = 7.5\%$, motivating caution about any router that depends on unobserved problem-type labels.

A later density-matrix simulator (v0.7.23) introduced phase- and amplitude-damping channels through Kraus operators,

$$\rho' = \sum_k E_k \rho E_k^\dagger, \quad \text{Tr}(\rho) = 1, \quad \text{purity}(\rho) = \text{Tr}(\rho^2). \quad (4)$$

Under full phase damping applied *after* the interference operation, the average purity declined from approximately 1 to 0.09, while the immediate computational-basis measurement success remained unchanged; this follows because the channel suppresses off-diagonal coherence without changing diagonal populations. Under full amplitude damping, all population moved toward the ground state; when that candidate had already been observed, the conditional distribution over unseen candidates became undefined. A transparent classical fallback then occurred in 24% of 100 trials per landscape. These observations validate specific numerical behaviors and boundary handling; they are not evidence of faster semantic computation.

4 Why an Interoperability Standard?

The experimental program indicates that reliable reuse and update semantics offer a plausible system-level research direction, whereas search speedups attributable to quantum-inspired transformations are conditional. However, reusable knowledge is only portable if independent systems agree on what is being transferred. A bare embedding does not specify an assertion’s interpretation, causal/conditional scope, source, contradiction handling, correction policy, or the semantics of deletion and internalization. Even when two systems store identical vector coordinates, their model spaces or decoding conventions may differ.

Existing standards already cover important parts: RDF expresses linked statements and resources [2]; JSON-LD 1.1 provides JSON-compatible Linked Data serialization [1]; PROV-O describes provenance [3]; SHACL constrains RDF graphs [4]; and MCP can expose URI-addressed resources to clients [5]. The intended contribution of SKI is therefore *an interoperable profile and lifecycle contract for AI knowledge memories*, built on rather than in opposition to these standards. Overlap with existing knowledge-graph, ontology-alignment, and event-sourcing approaches remains a research threat and must be mapped systematically in a later related-work study.

5 Semantic Knowledge Interchange (SKI) v0.1

5.1 Status and scope

SKI v0.1 is an *independent editor’s draft* documented within the SEMOS_SIM project [6]. It defines a proposed interchange layer for producers and consumers of semantic assertions. The draft is neither ratified by a standards development organization nor supported by demonstrated third-party implementations at the time of writing. The example vocabulary namespace <https://example.org/ski/v0.1#> is intentionally illustrative and not a deployed vocabulary service.

5.2 Normative core record and knowledge status

The SKI-Core proposal describes an assertion a as

$$a = (id, claim, issuer, time, version, lifecycle, context, conditions, evidence, assessments, policy). \quad (5)$$

A structured claim contains at least a subject, predicate, and object. A receiving system *MUST* validate the presence and types of required fields before promotion; *SHOULD* preserve provenance and original assessments; and *MUST NOT* interpret receipt as proof, authorization, or implicit permission to train model weights. Here *MUST*, *SHOULD*, and *MAY* express intended draft conformance requirements, not currently certified standard behavior.

Importantly, the model separates three independent axes:

- **Lifecycle:** CANDIDATE, APPROVED, SUPERSEDED, RETRACTED.
- **Epistemic assessment:** TRUE, FALSE, UNVERIFIED, CONTESTED, OUTDATED, always associated with assessor, time, method, and optional evidence.
- **Memory realization:** EXTERNAL, CACHED, INTERNALIZED, UNKNOWN, local to a particular system and model instance.

An “approved” record need not be true, a true assessment by one assessor need not be accepted by another, and internalized weights cannot be assumed to be deleted merely because an external assertion was retracted.

5.3 Illustrative JSON-LD exchange payload

The following abbreviated JSON-LD payload is a *proposal example*, not yet a validated instance of an official SKI ontology or SHACL shape.

Listing 1: Illustrative SKI v0.1 record; all example.org terms are placeholders.

```
{
  "@context": {
    "ski": "https://example.org/ski/v0.1#",
    "prov": "http://www.w3.org/ns/prov#",
    "issuer": {"@id": "ski:issuer", "@type": "@id"}
  },
  "@id": "urn:uuid:2ff02d96-7377-4e65-9108-7c1593f6b90a",
  "@type": "ski:KnowledgeAssertion",
  "issuer": "https://example.org/agents/producer-A",
  "ski:recordedAt": "2026-10-18T10:00:00+09:00",
  "ski:version": 1,
  "ski:lifecycle": "CANDIDATE",
  "ski:claim": {
    "@type": "ski:StructuredClaim",
    "ski:subject": {"@id": "https://example.org/concepts/gpu"},
    "ski:predicate": {"@id": "https://example.org/relations/accelerates"},
    "ski:object": {"@id": "https://example.org/concepts/matrix-computation"}
  },
  "ski:conditions": [{"ski:description": "with appropriate GPU implementation"}],
  "ski:assessments": [{"ski:epistemicStatus": "UNVERIFIED"}],
  "prov:wasDerivedFrom": {"@id": "https://example.org/evidence/benchmark-1"}
}
```

The assertion ID identifies a specific issued record, not an eternal, context-free equivalence class of all statements with similar wording. Canonicalization must distinguish assertion identity, graph syntax equivalence, entity linkage, and entailment; embedding cosine similarity alone is insufficient.

5.4 Exchange events and distributed synchronization

SKI-Exchange is a proposed optional event profile with operations **ASSERT**, **ASSESS**, **CORRECT**, and **RETRACT**. A minimal event contains an event identifier, assertion identifier, issuer, time, operation, and previous/new versions. Receivers should enforce idempotence for duplicate identifiers, detect version gaps, retain conflicting histories pending resolution, and propagate correction/retraction notices to known dependent copies. Conflicts must not be hidden by arbitrary last-writer-wins semantics when epistemic claims disagree. Signature schemes, trust federation, causal ordering, and full replication guarantees remain to be specified.

5.5 Security, privacy, and implementation boundaries

Knowledge payloads can contain malicious instructions, confidential data, or references to hostile resources. Implementations therefore require provenance authentication, schema validation, access policy enforcement, content sanitization, controlled outbound URI resolution, audit trails, and deletion/retention governance. In particular, an external textual claim must never automatically become a privileged instruction for an agent or LLM. Remote JSON-LD contexts should be pinned, cached, or disallowed according to a strict policy to limit network abuse and semantic substitution attacks. Model internalization should be a separately authorized operation with an explicit, testable rollback claim (or a documented inability to guarantee full removal).

6 Conformance and Evaluation Plan

SKI standardization should be driven by independent implementations and falsifiable acceptance criteria. Table 3 proposes a staged roadmap, rather than portraying missing features as achieved.

We propose five primary outcome metrics: (i) structural preservation of claims, conditions, and provenance; (ii) semantic correctness of reconstructed propositions in independent consumers; (iii) event convergence and conflict preservation after disconnection/reordering; (iv) refusal of unauthorized or malicious transfers; and (v) operating costs of serialization, validation, indexing, and synchronization. A first experiment could export 100 curated LLM_SEM assertions, import them into LLM_TRY and an independent reference consumer, round-trip them, and test corrections, contradictory assessments, missing evidence, and retractions. These are *future experiments*; no SKI interoperability percentage is claimed in this paper.

7 Threats to Validity

1. **Synthetic workloads.** Many reported execution measurements use controlled DAGs and proxy costs; real application traces, heavy-tail distributions, and adversarial updates are missing.

Table 3: Proposed SKI development and evaluation milestones (not completed results).

Stage	Deliverable	Acceptance evidence
v0.1	Editor’s draft and example exchange records	Published vocabulary proposal and reviewable requirements
v0.2	Stable vocabulary IRIs, JSON-LD contexts, SHACL shapes	Positive/negative fixture validation and canonicalization tests
v0.3	Python producer/consumer and semantic-memory adapters	LLM_SEM $\rightarrow$ LLM_TRY round-trip of 100 assertions
v0.4	Synchronization, conflicts, and retraction propagation	Replay, duplication, out-of-order, partition/re-join scenarios
v0.5	Third-party implementation and security review	Independent producer–consumer interoperability report
v1.0 candi-date	Governance, versioning, compatibility, conformance suite	Published change process, stable test vectors, external feedback

2. **Baseline equivalence.** A semantic engine must be compared with a conventional system possessing the same recognition, caching, dependency tracking, and consistency features.
3. **Limited statistical evidence.** Exploratory quantum studies use finite independent holdouts and multiple comparisons, with no fully pre-registered, multiplicity-controlled confirmatory study.
4. **Execution-system fidelity.** Early QHA experiments employ simulated VMs, mock semantic engines, and simulated internalization markers rather than production VM/LLM-weight integrations.
5. **Quantum interpretation.** Ideal state vectors and density matrices on classical CPUs are valid numerical models for small systems but not evidence of a hardware quantum advantage.
6. **Specification maturity.** SKI v0.1 lacks finalized vocabulary ownership, authoritative ontology, formal SHACL suite, independent implementers, and inter-organization security governance.

8 Discussion and Conclusion

The SEMOS_SIM evidence suggests that explicit semantic identity, dependency awareness, and reusable knowledge can guide useful system-level optimizations under identifiable conditions. By contrast, quantum-inspired amplitude methods did not consistently outperform strong classical search baselines; noise studies primarily validated physical-model consistency and exposed runtime boundary cases. A pragmatic architecture should therefore treat quantum-inspired selection as an optional, measured heuristic, not a defining source of acceleration.

We propose SKI as a complementary and potentially more widely transferable outcome: a portable, provenance-aware knowledge contract that distinguishes what a system asserts, who assessed it, when it is applicable, how it changes, and where it is stored or internalized. Its value will depend on proving *interoperability rather than simply inventing another serialization*. The next decisive milestone is an independently reproducible SKI producer/consumer conformance experiment coupled with a matched, real-data Semantic OS performance benchmark. Until such results exist, SKI remains a research proposal and semantic execution gains remain conditional observations.

Availability and Reproducibility

Project repository: https://github.com/digiponta/SEMOS_SIM. The v0.7.x scripts and reported experiment names include `quantum_classical_matched_v0722.py`, `pilot_profile_router_v0720.py`, and `quantum_density_noise_v0723.py`. The SKI v0.1 draft is stored at `docs/SKI_SPECIFICATION_v0.1_ja.md`. The figures and percentages reproduced here are from user-provided experimental logs and earlier reports; they have not been independently rerun or audited during preparation of this manuscript. This distinction is essential for external review.

References

- [1] W3C, “JSON-LD 1.1: A JSON-based Serialization for Linked Data,” W3C Recommendation, July 2020. <https://www.w3.org/TR/json-ld11/>.

- [2] W3C, “RDF 1.1 Concepts and Abstract Syntax,” W3C Recommendation, 2014. <https://www.w3.org/TR/rdf11-concepts/>.
- [3] W3C, “PROV-O: The PROV Ontology,” W3C Recommendation, 2013. <https://www.w3.org/TR/prov-o/>.
- [4] W3C, “Shapes Constraint Language (SHACL),” W3C Recommendation, 2017. <https://www.w3.org/TR/shacl/>.
- [5] Model Context Protocol, “Resources,” Specification release July 28, 2026. <https://modelcontextprotocol.io/specification/2026-07-28/server/resources>.
- [6] H. Inomata, “Semantic Knowledge Interchange (SKI) Specification v0.1,” independent editor’s draft, October 2026. https://github.com/digiponta/SEMOS_SIM/blob/v0.7.0/docs/SKI_SPECIFICATION_v0.1_ja.md.
- [7] H. Inomata, “SEMOS_SIM: Semantic OS Simulation Research,” source repository and synthetic benchmark scripts, 2026. https://github.com/digiponta/SEMOS_SIM.