PLANCKCHRON FLAGSHIP TECHNICAL SERIES

EMERGING COMPUTE / VERSION 1.0 / 09 AUGUST 2026

Aetherion Core

A Benchmark-First Architecture for Hybrid Quantum-Classical Compute Research

ABSTRACT

Aetherion Core is defined here as a proposed research architecture for decomposing workloads across classical processors, accelerators, simulators, and external quantum backends while preserving comparable baselines, resource accounting, verification, and reproducibility. The paper rejects raw device scale as a sufficient proxy for usefulness. It specifies a workload contract, solver registry, experiment compiler, execution adapters, verification harness, and evidence ledger. A quantum or post-silicon branch is justified only when the complete hybrid workflow is compared with the best available classical approach under a common quality and resource budget. Recent external quantum-computing research strengthens the need for built-in validation, but those external results are not Planckchron capability. Planckchron has not publicly disclosed a validated processor, physical or logical qubit count, gate-quality result, coherence result, or quantum advantage.

PLANCKCHRON RESEARCH

Self-published technical paper / not peer-reviewed / open access

planckchron.com/publications/aetherion-core-benchmark-first-hybrid-compute

TECHNICAL NOTE

Executive brief

Aetherion Core is defined here as a proposed research architecture for decomposing workloads across classical processors, accelerators, simulators, and external quantum backends while preserving comparable baselines, resource accounting, verification, and reproducibility. The paper rejects raw device scale as a sufficient proxy for usefulness. It specifies a workload contract, solver registry, experiment compiler, execution adapters, verification harness, and evidence ledger. A quantum or post-silicon branch is justified only when the complete hybrid workflow is compared with the best available classical approach under a common quality and resource budget. Recent external quantum-computing research strengthens the need for built-in validation, but those external results are not Planckchron capability. Planckchron has not publicly disclosed a validated processor, physical or logical qubit count, gate-quality result, coherence result, or quantum advantage.

EVIDENCE BOUNDARY

Research architecture only. Planckchron has no publicly disclosed validated processor, qubit count, gate-quality result, coherence result, fault-tolerant device, or quantum-advantage result.

Four operating conclusions

- The unit of comparison is the complete hybrid workflow, not a hardware headline.
- Every experiment needs a classical baseline, common budget, quality metric, uncertainty statement, and reproducibility package.
- Verification must be designed into the computation before the classically hard regime is reached.
- A solver earns routing priority from evidence on a workload class, not from its substrate label.

STATUS	**Concept architecture**
PUBLISHER	Planckchron Inc.
CITATION	*Planckchron Research (2026). Aetherion Core: A Benchmark-First Architecture for Hybrid Quantum-Classical Compute Research. Version 1.0.*

TECHNICAL NOTE

Reference architecture

The architecture below is a proposed control and evidence flow. It is a design artifact, not a representation of deployed capability.

01

DESCRIBE

Convert the problem into a typed workload contract with objective, constraints, data, quality metric, and resource budget.

02

DECOMPOSE

Separate classical preprocessing, candidate quantum or accelerator kernels, and classical postprocessing.

03

ROUTE

Select candidate solvers from evidence-backed capability descriptors rather than marketing specifications.

04

EXECUTE

Compile reproducible experiments through isolated simulator and hardware adapters.

05

VERIFY

Use baselines, tractable instances, consistency checks, uncertainty bounds, and process evidence.

06

COMPARE

Evaluate end-to-end quality, time, cost, energy where measurable, and operator effort under one contract.

TECHNICAL NOTE

1. Research objective

Emerging compute is not one homogeneous resource. A useful workflow may combine conventional processors, graphics or specialized accelerators, numerical simulators, and quantum devices. Each substrate has different input constraints, failure modes, queueing behavior, precision, observability, and cost. A credible architecture therefore begins with the workload and evidence contract rather than a preferred machine.

Aetherion Core is proposed as a control and evaluation layer for that environment. It does not imply ownership of a quantum processor or the existence of proprietary post-silicon hardware. Its near-term research value is to make solver selection, experiment construction, verification, and comparison repeatable across changing backends.

TECHNICAL NOTE

2. Workload contract

The workload contract defines the problem independently of any provider. It contains objective, constraints, input provenance, permissible approximations, output quality metric, confidence requirement, baseline methods, time and cost budgets, security classification, and reproducibility requirements. It also records whether the value lies in an exact answer, a sample, an expectation value, a feasible solution, or improved scientific understanding.

Without this contract, a system can select a backend based on an impressive but irrelevant metric. With it, every solver must demonstrate fitness for a named problem class under comparable conditions.

TECHNICAL NOTE

3. Reference architecture

A problem canonicalizer validates the workload contract and produces a stable intermediate form. A decomposition engine identifies candidate kernels and their data dependencies. The solver registry stores capability descriptors supported by dated evidence: problem class, scale range, precision, known failure modes, required preprocessing, expected queue and execution behavior, and validation method.

The experiment compiler emits a versioned execution package containing source, parameters, seeds where applicable, environment, backend request, mitigation configuration, baseline configuration, and analysis plan. Backend adapters isolate provider-specific interfaces. The verification harness runs classical baselines, tractable subproblems, invariants, cross-platform repetitions, or process-level checks appropriate to the experiment. The evidence ledger records outputs, uncertainty, failures, resource use, and review decisions.

TECHNICAL NOTE

4. Benchmark contract

Benchmarking should compare systems on the same useful task and budget. Device-level metrics may help diagnose hardware but do not by themselves establish application value. The benchmark report therefore includes the full hybrid path: data preparation, encoding, circuit or kernel construction, execution, error handling, decoding, postprocessing, and validation.

The classical comparison must be strong enough to challenge the premise. It should name the algorithm, implementation, hardware, optimization effort, approximation, stopping condition, and confidence interval. If new classical work overturns an advantage candidate, the record should change. The purpose is discovery, not defending a substrate.

- Task fidelity: both paths solve the same stated problem.
- Quality: accuracy, objective value, distributional similarity, physical consistency, or another domain-relevant measure.
- Resources: wall time, compute time, queue time, cost, memory, energy where measured, and operator effort.
- Uncertainty: statistical error, device noise, model error, approximation error, and run-to-run variation.
- Reproducibility: complete configuration, code, data lineage, and instructions sufficient for an informed rerun.
- Failure disclosure: timeouts, discarded runs, post-selection, mitigation assumptions, and negative results.

TECHNICAL NOTE

5. Verification beyond exact classical answers

As experiments enter regimes that are difficult to simulate exactly, verification must move from a single external answer to a set of trust-building methods. These may include classically tractable reference instances, embedded consistency checks, physical invariants, cross-platform agreement, independent estimators, calibrated error mitigation, logical or process-level syndrome evidence, and statistical lower or upper bounds.

External research reported in 2026 illustrates this direction by integrating validation into hard quantum computations and by comparing independent mitigation methods and hardware platforms. Those studies are cited because they inform the architecture. They are not Planckchron results, and their claims remain open to continuing classical challenge and replication.

TECHNICAL NOTE

6. Evaluation program

The first evaluation suite should be simulator-first and include optimization, sampling, linear-algebra, and physical-model workloads small enough for strong classical verification. It should measure routing correctness, reproducibility, failure containment, baseline quality, and evidence completeness before attempting claims about speed or quality.

A later phase may connect to external hardware providers using the same contract. The acceptance gate is not a favorable result; it is whether the system can reproduce, explain, and fairly compare the result. Independent reruns should precede any material public performance statement.

TECHNICAL NOTE

7. Research roadmap and stop criteria

Phase one builds the workload schema, solver registry, and simulator adapters. Phase two adds benchmark automation and the verification ledger. Phase three studies selected external backends. Phase four invites independent review of methods and results. Hardware research, if pursued, requires a separate evidence program and must not inherit maturity from the orchestration layer.

Work should pause or narrow when a classical method closes the claimed gap; when verification cannot support the target regime; when resource accounting is incomplete; when backend variation overwhelms the effect; or when the workflow depends on undisclosed post-selection or manual intervention.

TECHNICAL NOTE

References and source notes

References are provided for the external standards, public guidance, and primary research that informed this paper. A citation does not imply endorsement, partnership, certification, or validation of Planckchron capability.

01 **IBM Quantum Research, 2026**

Quantum Advantage Through Trusted Quantum Computation

`https://www.ibm.com/quantum/blog/quantum-advantage`
External company-reported research linking primary papers; no Planckchron result or affiliation is implied.

02 **Google Quantum AI**

Quantum Error Correction Below the Surface Code Threshold

`https://quantumai.google/research`
External peer-reviewed research listing; no Planckchron result is implied.

03 **NIST Research**

Randomized Benchmarking of Quantum Gates

`https://tf.nist.gov/general/pdf/2315.pdf`
External public research reference on benchmarking.

04 **NIST Research**

The Complexity and Verification of Quantum Random Circuit Sampling

`https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927073`
External public research reference on verification.

05 **IBM and Pasqal, 2025**

The Dawn of Quantum Advantage

`https://www.ibm.com/quantum/blog/quantum-advantage-era`
External position and validation framework; not independent proof of Planckchron capability.

PUBLICATION DISCLOSURE

These self-published technical papers are not peer-reviewed, audited, or independent validation. They describe research methods, proposed architectures, and evaluation plans. Product capability exists only where a dated evidence record expressly reports a result. Research architecture only. Planckchron has no publicly disclosed validated processor, qubit count, gate-quality result, coherence result, fault-tolerant device, or quantum-advantage result.