FINTRACE Project Plan

Idea, Innovation, Business · A Science Challenge · Phase 1

David Allen Blubaugh | davidblubaugh2000@yahoo.com | November 7, 2026

Project title as posted. FINTRACE: Assured Agentic Financial Intelligence with Knowledge Graphs, Data Lakes, FPGA Acceleration, and Quantum Optimization

Public project post and submission record

0 Project summary

FINTRACE is a proposed software platform that helps financial analysts and model reviewers trace a research claim or simulated portfolio proposal back to the information available at the decision cutoff. The Python demonstrator will use a versioned data lake, a temporal knowledge graph and document retrieval to prepare evidence-linked reports for a small asset universe. Deterministic calculations, an independent audit worker and a separate release gate will check the evidence and constraints before an advisory report reaches a human reviewer. By April 23, 2027, I plan to deliver the demonstrator, a repeatable baseline comparison, and a review package showing a released report, a withheld report and recovery from an interrupted run. FPGA and quantum optimization will be evaluated separately if the classical baselines and access requirements are met.

1 Innovation 25%

What is new. FINTRACE proposes a release policy that binds a specific candidate and its audit verdict to a time-pinned data manifest. A reviewer can follow a claim to the source version and exact passage, or follow a number to its inputs, units and reference calculation. The same policy applies to outputs from the classical solver and optional experimental compute.

Closest alternatives. OpenBB provides financial data integration and research workflows [3]. FinQA benchmarks numerical reasoning over financial reports [4]. I will use local comparisons that isolate FINTRACE's design choices: deterministic retrieval and computation, one agent with document-only retrieval, one agent with graph-assisted retrieval, and the specialist workflow with the same graph, corpus and tool policies.

Expected improvement. The proposed targets are at least 10 percentage points more correctly supported material claims with graph-assisted retrieval and at least 20% lower median reviewer time with specialist orchestration. The pilot will report observed effects, uncertainty, cost and coverage, even when these targets are missed.

My original contribution. My contribution is the combination of time-pinned evidence, separate audit authority and candidate-bound release in one testable financial workflow, together with an experimental design that measures the marginal value of retrieval, specialization and optional compute. Dupouy and El Mofatiche provide background on financial agents and evaluation [2].

2 Technical plan 25%

Build and available resources. Start on a general-purpose computer using Python, Git, versioned Parquet files and DuckDB, JSON Schema or Pydantic, and SciPy. Use SQL tables for temporal relationships and a local document index first. Consider a dedicated graph service only if the relational prototype limits evaluation. The first milestone uses archived SEC filings and synthetic portfolio fixtures. SEC EDGAR and optional ALFRED vintages provide candidate historical inputs [5–6]. External LLM calls require approval of data handling terms and a spending cap. CPU execution is the required baseline. GPU, FPGA board, QPU time and paid feeds depend on confirmed access.

Smallest useful build. Use two issuers and one cutoff to reproduce a cited financial ratio, identify a filing amendment dated after the cutoff, produce a feasible synthetic portfolio and demonstrate HOLD when a required citation is missing. Scale to six issuer-linked securities over three sectors and six monthly historical cutoffs once versioning, identifiers and data rights pass the first gate.

Biggest unknowns. Test source availability and rights first, including amendments and corporate actions. Keep filing-based research and synthetic portfolio simulation viable if suitable price history is unavailable. Test retrieval and orchestration against paired baselines before increasing complexity. Inject stale data, conflicting claims, fabricated citations and service failures before enabling release. Use profiling and a matched solver benchmark to decide whether either compute experiment merits development.

Prototype and test sequence. Build ingestion, a frozen manifest, deterministic metrics and an independently checked solver before introducing agents. Add document-only retrieval, then graph-assisted retrieval, then the five-worker workflow. Use the same held-out questions, cutoffs, constraints and review rubric across variants. Exercise restart, access revocation and candidate tampering with fault-injection fixtures.

Success criterion	Proposed April acceptance target
Evidence and timing	Every released material claim has a resolvable citation or verified calculation, with zero cutoff or entitlement violations in the test suite. At least 95% of released material claims must be judged correct and supported on the labeled set.
Portfolio and calculation	Every released candidate passes the independent constraint checker. All critical reference fixtures pass declared unit and numeric tolerances. Seeded deterministic runs reproduce decisions and canonical outputs.
Security and recovery	Every mandatory fault case causes the expected HOLD or safe reference fallback. Zero prohibited tool calls, duplicate published artifacts or reuse of a verdict for changed bytes.
Release usefulness	Deliver supported answers for at least 80% of answerable held-out tasks. Score correct abstentions separately on unanswerable cases. Track review time, p50 and p95 latency, per-run cost and false HOLDs. Evaluate improvement targets separately from release safety.

The jury deliverable includes a Python prototype, architecture and interface specification, ontology and data dictionary, rights-cleared fixtures, evaluation harness, results, limitations and optional experiment reports or documented decisions to defer.

3 Execution plan 25%

3.1 Milestones

Period	Milestone	Done when
Nov 16–Dec 15, 2026	Prove a minimum evidence-to-report workflow.	Two issuers and one cutoff yield a cited ratio and feasible synthetic portfolio. A future amendment and missing citation exercise cutoff rejection and HOLD. Data rights and versioning rules are recorded.
Dec 16–Jan 15	Establish classical baselines.	The point-in-time corpus, CPU solver, deterministic report and single-agent document baseline run from one frozen configuration. Labeling rubric and held-out split are frozen. Prepare the two-month report for the organizer's reporting window, around January 16.
Jan 16–Feb 15	Measure retrieval effects.	Graph and document variants run on identical cases. Record citation support, answer coverage, latency and cost. A failed graph benefit test retains the simpler retrieval path.
Feb 16–Mar 15	Verify controlled orchestration.	Five typed roles, independent audit and hash-bound release pass negative and restart tests. Complete orchestration ablation and CPU profiling. Prepare the four-month report for the organizer's reporting window, around March 16, with optional compute decisions.
Mar 16–Apr 23, 2027	Deliver and defend a reproducible demonstration.	Reproduce the deterministic demo from a clean environment and package methods, results and limitations, including any optional experiment report or deferral. Demonstrate PASS, HOLD and interrupted-run recovery at the Apr 19–23 defense window.

3.2 Business outline

Target customer. Research or model-governance leads at small asset managers and financial research teams. University finance labs are plausible early evaluation users.

Value. Reduce the time spent tracing citations, checking what was knowable at the time and reconstructing simulated portfolio decisions. Provide a reviewable audit bundle and retain the analyst's approval step.

Rough economics. Proposed pilot price of $250–$750 per team per month. Illustrative monthly direct compute and hosting cost of $50–$150 per team excludes labor, data licenses, legal review and support. Pilot interviews and actual metering will determine whether the economics are viable.

Path to market. Seek three to five interviews with analysts, governance reviewers and instructors. Offer a synthetic or rights-cleared demonstration and seek one evaluation pilot after the defense. Test willingness to pay and compare review time with the team's existing workflow.

Proposed direct prototype budget is up to $300, excluding applicant time, licensed market data and hardware. Allocate up to $200 for approved model calls, $50 for storage or hosting and $50 for contingency. Track consumption per run and defer any paid resource outside that budget until separately arranged.

4 Capacity to deliver 15%

Background. I hold BS and MS degrees in electrical engineering from Wright State University and am a PhD student in Electrical and Systems Engineering at the University of Dayton. I bring Python, FPGA and digital engineering experience, including traceability work with SysML and knowledge graphs. I co-authored Intelligent Autonomous Drones by Cognitive Deep Learning (Apress, 2022).

Skill area	Who covers it	Role
Technical and engineering	David Allen Blubaugh	Own the architecture, ingestion, temporal model, agent contracts, benchmark code and profiling.
Testing, experiments and data	David Allen Blubaugh. Independent reviewer sought.	Build fixtures and lead verification. Seek a second reviewer to check claim labels and portfolio assumptions.
Business and customers	David Allen Blubaugh. Interviewees sought.	Test buyer needs, pricing and review workflow through discovery interviews.
Domain and specialist support	Finance, graph/data-rights, FPGA and quantum contributors to be sought as needed.	Review modeling and access assumptions or optional benchmark design. Core delivery uses CPU tools and the applicant's own work.

Time and support for Nov 16–Apr 23. Planning assumption of 8–10 hours per week, approximately 180–225 hours across the project window. Applicant to confirm availability. The core schedule uses local development and publicly accessible source material subject to its terms. Institutional, employer, advisor and specialist support require separate confirmation.

5 Real world impact 10%

Need and evidence. Financial research combines narrative disclosures, tables and numeric reasoning. FinQA documents the difficulty of that task [4]. EDGAR provides filing histories and ALFRED retains data vintages, giving FINTRACE testable sources for reconstructing what evidence was available [5–6]. Customer demand and time savings remain hypotheses for the pilot.

What it could lead to. Auditable company and sector research, constrained portfolio analysis, historical and hypothetical stress tests, and institutional memory that retains superseded and conflicting findings. Longer-term applications include analyst workflow support, governance and model audit, and appropriately governed insurance or credit research. The benchmark harness also supports research on heterogeneous computing.

Who benefits. Analysts and model reviewers gain traceable evidence and review records. Educators and students gain repeatable examples of cutoff leakage, source revisions and simulation. Broader financial use depends on domain validation, security, data rights and application-specific review.

6 Confirmation

"All work will follow applicable university, platform, consumer-safety, consent, and privacy requirements."

Applicant name David Allen Blubaugh | Prepared November 7, 2026

Applicant review and confirmation are required before submission. The supplied template specifies November 6, 2026 at 11:59 p.m. ET for Phase 1. Confirm acceptance before forwarding this plan. The supplied posting lists "Synthetic Biology" under "AI Smart Technology." Confirm the appropriate category with the organizer.

Appendix A Architecture and authority

The three sheets of FINTRACE FT-ARCH-001, Revision B (November 7, 2026) [1] specify the proposed system architecture, authority contracts and verification lifecycle. The following design retains their identifiers and turns their logical connections into testable implementation requirements.

Element	Proposed implementation and authority
D01–D02 Sources and ingestion	Ingest filings, statements, eligible prices and corporate actions, optional news and macro releases, and controlled experiment data. Treat external content as untrusted. Validate identity, units, schema, timestamp, rights and quality. Quarantine ineligible data with a reason code.
D03–D04 Lake and projections	Keep content-addressed source payloads and accepted normalized versions within retention rights. Build temporal graph and vector views from the same frozen snapshot. Only ingestion and projection services can publish data and views.
C01 and A01–A05	The controller freezes run context, dispatches typed tasks and enforces retries, deadlines and compute budgets. Workers can submit proposed artifacts through validated services. Base stores remain read-only to workers.
S01 and S02	The knowledge gateway enforces entitlement and cutoff on graph, index and lake reads. The compute broker accepts allowlisted typed tools and validated results. CPU is the reference path. LLMs receive scoped context and have no raw-store credentials, shell or unrestricted outbound tools.
L01 and C02	L01 keeps append-only run events and versioned outputs within policy. C02 rechecks release conditions and binds PASS to the exact candidate and manifest. A05 has a separate read role. It can issue a verdict but cannot modify the candidate or release it.
U01 and H01–H02	The human sets the research question and approves assumptions before the run, then reviews the released advisory report or HOLD notice. Optional compute sits behind S02. Human approval is a review action. The demonstrator has no order-routing interface.

Specialist worker contracts

Role	Typed inputs and outputs
A01 Data Quality	SnapshotManifest and QualityPolicy → CoverageReport with rights, time and identity exceptions. Required gaps stop dispatch downstream.
A02 Evidence and Research	ResearchQuestion, permitted snapshot and accepted coverage → EvidenceSet with passage offsets, date, source version, contrary evidence and gaps.
A03 Financial Analysis	EvidenceSet and MetricSpecification → ClaimSet and CalculationTrace with units, assumptions, evidence references and uncertainty.
A04 Risk and Optimization	ScenarioSet, portfolio constraints and prior weights → PortfolioCandidate with objective, exposures, residuals and solver status. Reference verifier recomputes feasibility.
A05 Independent Audit	Read-only candidate, evidence, manifest and tool traces → AuditVerdict with reasons, failing checks, reviewer notes and exact candidate hash.

Appendix B Interfaces and run recovery

Interface	Contract
I01 D01 → D02	Payload, source ID, rights, publication or availability, effective and ingestion times. Process source text as data.
I02 D02 → D03	Accepted normalized records, validation results and raw references. Failed inputs stay in quarantine and remain ineligible.
I03 D03 → D04	Idempotent projection keyed by dataset, ontology and index versions. Publish manifest atomically after coverage and consistency checks.
I04 S01 ↔ D03 and D04	Read lake evidence and derived views by pinned manifest. D04 is the shared read port shown on schematic Sheet 1.
I05 C01 ↔ A01–A05	Typed tasks and results with bounded permissions, deadlines and escalation status.
I06 Workers ↔ S01	Scoped retrieval returns dated evidence and exact source versions after entitlement and time checks.
I07 Workers ↔ S02	Typed calculation or model request and validated output. Record tool, model, code and seed identifiers.
I08 C01 → C02	Candidate hash and complete audit bundle, including A05 verdict. Gate checks artifact bytes, policy, manifest and verdict identity before publication.
I09 All stages → L01	Log input and output hashes, schemas, manifest, software and model versions, seeds, usage, timing, errors and verdicts. Protect evidence and secrets in telemetry.
I10 U01 ↔ C01 and C02	Reviewer's request and approved assumptions enter C01. C02 sends the exact released advisory report or HOLD disposition.
I11 S02 ↔ H01	Array shape, units, precision and device/kernel versions. Return numerical outputs and parity and performance results. Use CPU reference or HOLD on failure.
I12 S02 ↔ H02	Objective, constraints, encoding and solver settings. Return samples, decoded candidates and costs. Verify feasibility classically.

Run lifecycle and failure handling

Configure → Pin Data → Research → Compute → Audit → Release. Freeze the question, universe, horizon, constraints, cutoff, data policy, thresholds and tool versions before work begins. Pin validated data and matching views, retrieve eligible evidence, compute reference metrics and solve, then audit and release the exact candidate for human review. An expired entitlement, missing or stale required input, unresolved material contradiction, infeasible candidate or unavailable required service triggers HOLD. Optional services may fall back to the validated reference path. Nonmaterial gaps remain visible as qualifications.

R01 restart rule. Record job identity and idempotency key before dispatch. On restart reconcile registry state with external job status before retrying. Use a unique publication key tied to run and candidate and reconcile an interrupted publish before issuing another. Preserve failures and rerun downstream work under a new version when inputs change. A changed candidate, manifest or policy requires a fresh verdict. Final gate decisions carry machine-readable reasons.

Appendix C Data and portfolio specification

Temporal data and financial ontology

Select six securities from three sectors using membership known at the first cutoff and keep the research universe fixed for comparison. Record stable internal issuer and instrument IDs, SEC CIK where applicable, historical symbol mappings and listing intervals. Include fixture cases for splits, dividends, ticker changes, delistings and restatements. Use six prespecified month-end cutoffs and a one-month research horizon. For price-based tests, require documented rights and contemporaneous prices, actions and volumes. If those are unavailable, use explicitly synthetic prices, liquidity and returns for portfolio and risk fixtures.

Entities	Minimum relationships and attributes
Issuer, Security, Industry	Issuer issues Security. Issuer classification and identifier mappings have validity intervals. An issuer can have multiple securities.
Source, SourceVersion, Filing, Event	SourceVersion has a hash, provenance, rights and timestamps. Filing identifies accession, form and period. Event links affected instruments. Amendments and events retain predecessor and effective-time links.
Metric, Observation, Claim	An observation links metric, issuer or security, reporting period, unit, currency and source version. A claim cites an exact passage or calculation and may contradict or supersede another claim.
Constraint, Scenario, PortfolioCandidate	A candidate uses versioned constraints, holdings, scenario data and objective. Store units, assumptions, solver status and feasibility residuals.
Run, Experiment, Model, Decision	A run pins the manifest, tools and policies. An experiment compares runs. A release decision binds candidate, audit result, checks and human review status.

Publication or first verifiable availability establishes when evidence could be known. Effective time identifies the period or event the record describes. Ingestion time identifies when FINTRACE acquired it. Revision history records what changed, when it became available and what it superseded. Historical-availability tests require archived contemporaneous versions available by cutoff. Operational replay also requires ingestion by the simulated decision time. Use UTC cutoff instants with source timezone retained. Exclude uncertain same-day release timing or conservatively admit it on the next verified day. Never use an effective date alone as evidence of availability.

Keep the lake authoritative. Build indexes only from the permitted snapshot and atomically publish a validated manifest after graph/vector alignment checks. Rebuild affected views when correcting or removing content. Deny current access to revoked content even for an old manifest. Purge or restrict raw content, indexes, caches and derived artifacts as terms require, retaining permitted tombstones and audit metadata. Benchmarks must document any replay lost to rights changes. Release only permitted benchmark fixtures and metadata.

Portfolio baseline and risk method

Use long-only weights in simulated monthly allocations. As a starting fixture, require weights summing to one, a 30% security cap, a 50% sector cap and one-way turnover of at most 25% from recorded starting weights. Add liquidity bounds using permitted lagged volume and simulated notional, or declared synthetic limits. Optimize a linear trade-off between assumed return and mean downside shortfall over frozen scenarios, with turnover costs where specified. Express absolute trades and scenario shortfall with linear auxiliary variables. Use SciPy/HiGHS on CPU, compare with feasible equal-weight and incumbent baselines, and verify small reference cases analytically or by enumeration [7]. Report objective, feasibility, concentration and sensitivity to risk weights, scenarios and costs. Examine historical or hypothetical correlated shocks and regime shifts. Any realized-return evaluation is descriptive and requires independently rights-cleared forward data.

Appendix D Evaluation and release gates

Evaluation protocol and decision gates

Every task/result envelope includes run ID, task ID, input hashes, UTC cutoff, schema version, status, evidence references, uncertainty and errors. Apply per-role allowlists, validated schemas, configurable time and token budgets, and at most one retry for transient errors in the pilot. Freeze an initial ceiling of five minutes and $2 in metered service charges per advisory run. Route persistent service errors to HOLD unless the declared reference fallback meets the full release contract. Deterministic services own arithmetic, constraint checks, entitlements and release policy. Agent agreement is insufficient evidence for release.

Preregister roughly 60 labeled question–cutoff cases across the six months and a separate suite of at least 30 negative or recovery fixtures. Freeze a chronological development and held-out split, with final cutoffs kept out of tuning. Annotate claims, acceptable evidence, formulas and expected abstentions. Seek independent adjudication of ambiguous labels and report achieved review coverage. Keep the same models, tools, corpus, cutoffs, token ceilings and tasks across relevant paired comparisons. Because a pretrained model may already know later events, score only source-supported answers from the pinned corpus, disable browsing and add synthetic and future-event traps. Use cutoff-aware joins and separate forward outcomes from feature and prompt inputs. Record uncertainty with paired intervals grouped by cutoff where feasible and treat the small pilot as exploratory.

Gate	Evidence and decision rule
G1 Data and classical baseline	Prove cutoff, rights and version rules on fixtures. Rerun identical deterministic outputs using canonical serialization, pinned dependencies and seeds. Compare critical dimensionless metrics at absolute tolerance 1e-8 or relative tolerance 1e-6. Define explicit unit-specific exceptions before testing. Verify normalized constraint residuals to 1e-8.
G2 Retrieval and orchestration	Score supported material claims, valid and relevant citations, answer coverage, false HOLDs and blinded reviewer minutes where feasible. Keep the simpler default if the graph or specialists miss their effect targets or increase p95 latency or cost by over 25% without documented review benefit.
G3 Release assurance	Pass fabricated citation, amendment leakage, unit/currency mismatch, entitlement revocation, prompt injection, candidate mutation, infeasible portfolio and retry/restart fixtures. Check that authorized tool calls remain within policy. Exact replay applies to stored artifacts and deterministic steps. Fresh LLM responses may differ [8–9].
G4 Optional compute	Freeze instance set, reference, precision, timing scope, resource budgets and success criteria before optional work. Publish a benchmark result, a negative result or the reason for deferral.

Report calibration separately from accuracy. Use Brier score and reliability bins only for explicit confidence values evaluated against correctness labels. Otherwise report categorical uncertainty, abstention rates and coverage, including sample counts. Freeze run cost, p50 and p95 latency measurement rules and separate cold starts from steady state. The audit bundle carries exact source versions and spans, artifact and input hashes, configurations, model/tool identities, seeds, budgets, outcomes, unresolved issues and release rationale. Test candidate substitution and replay independently of any audit model.

Appendix E Optional compute experiments

H01 FPGA scenario kernel

Profile the CPU workflow before development. Proceed to a device experiment only if the scenario kernel consumes at least 30% of representative runtime, matching hardware is accessible, and the reference passes. Start with correlated scenario transformation and portfolio P&L over fixed input shocks. Feed identical arrays to a NumPy float64 oracle and a fixed-point or floating-point FPGA implementation. Test absolute P&L error ≤1e-5 of portfolio notional for the declared input envelope and inspect worst cases and tails. Measure host-to-device transfer, execution and return, end-to-end p50/p95, throughput, total host/device energy when measurable, and hardware and deployment cost. A proposed continue criterion is ≥20% lower end-to-end median latency or energy at passing parity and no material regression in the other measured dimension. If hardware or power metering is unavailable, label the result simulation or timing only. Derivative pricing is a later profiling candidate. LLM inference stays on conventional compute.

H02 Quantum constrained selection

Evaluate a separate equal-weight cardinality-constrained binary subset of six to twelve assets using the same return/risk coefficients and selection and exposure constraints across methods. Minimize a frozen combination of portfolio variance and negative assumed return over equal-weight selections, with fixed cardinality and sector exposure bounds. Encode constraints with prespecified penalties where required and evaluate decoded samples against the original constrained objective. Freeze instances, encoding, shot counts, seeds and feasibility decoder. Compare a simulator and any accessible QPU against exact enumeration and a classical discrete solver, using a MILP linearization for binary products where needed [7, 10]. Score feasibility before and after any repair, optimality gap on decoded feasible candidates, time to a verified feasible result and total cost including encoding, classical optimization, execution, sampling, queues and postprocessing. Treat simulator outcomes as algorithm evidence. Advance only if resource-normalized results on matched instances merit further tests. No profitability or compute advantage is presumed.

Appendix F Source reconciliation and references

The challenge template supplies the scoring weights, milestone periods, reporting cadence, defense window and confirmation language. The supplied public-post text provides the exact posted title and contact, the application areas, fourteen open questions and ten requested implementation topics. The schematic supplies architecture identifiers, interfaces, lifecycle and acceptance evidence. Together they describe a conceptual research baseline.

Source issue	Treatment in this plan
Posting metadata	The title and email follow the supplied record. “AI Smart Technology” is the listed technology. “Synthetic Biology” is the supplied category and requires organizer confirmation. Supplied tags are Proposal, Data, Analysis and #IdeaInnovationBusiness. The live posting and current submission status could not be independently confirmed.
Graph port and logging on Sheet 1	Sheet 2 defines I04 as a pinned read across D03 and D04, with D04 drawn as the shared port. I09 logging spans all stages. Sheet 1 omits the human request input path for clarity. The contracts in Appendix B govern implementation.
Assurance and readiness	A05 gives an independent verdict. C02 is the technical release authority. U01 reviews the outcome. Proposed experiments, budget, thresholds, data rights and staffing remain planning assumptions until measured or confirmed.
Open questions and technical scope	Sections 1–2 and Appendices C–E resolve provisional choices for scope, universe, ontology, synchronization, baselines, attack tests, profiling, quantum formulation, thresholds and fallback. Section 4 distinguishes the applicant’s work from contributors still to be sought.
Date and attestation	The supplied template sets the deadline and reporting windows. The November 7 preparation date records this proposal revision. Applicant confirmation and acceptance of a late submission require follow-up.

References

[1] FINTRACE Engineering Schematic, FT-ARCH-001, Revision B, November 7, 2026. Companion schematic, sheets 1–3.

[2] Dupouy and El Mofatiche. Building AI Agents for Finance. Packt, 2026.

[3] OpenBB documentation and Open Data Platform.

[4] Chen et al. FinQA. EMNLP 2021.

[5] SEC EDGAR Application Programming Interfaces.

[6] Federal Reserve Bank of St. Louis. ALFRED and real-time periods.

[7] SciPy documentation. Mixed-integer linear programming.

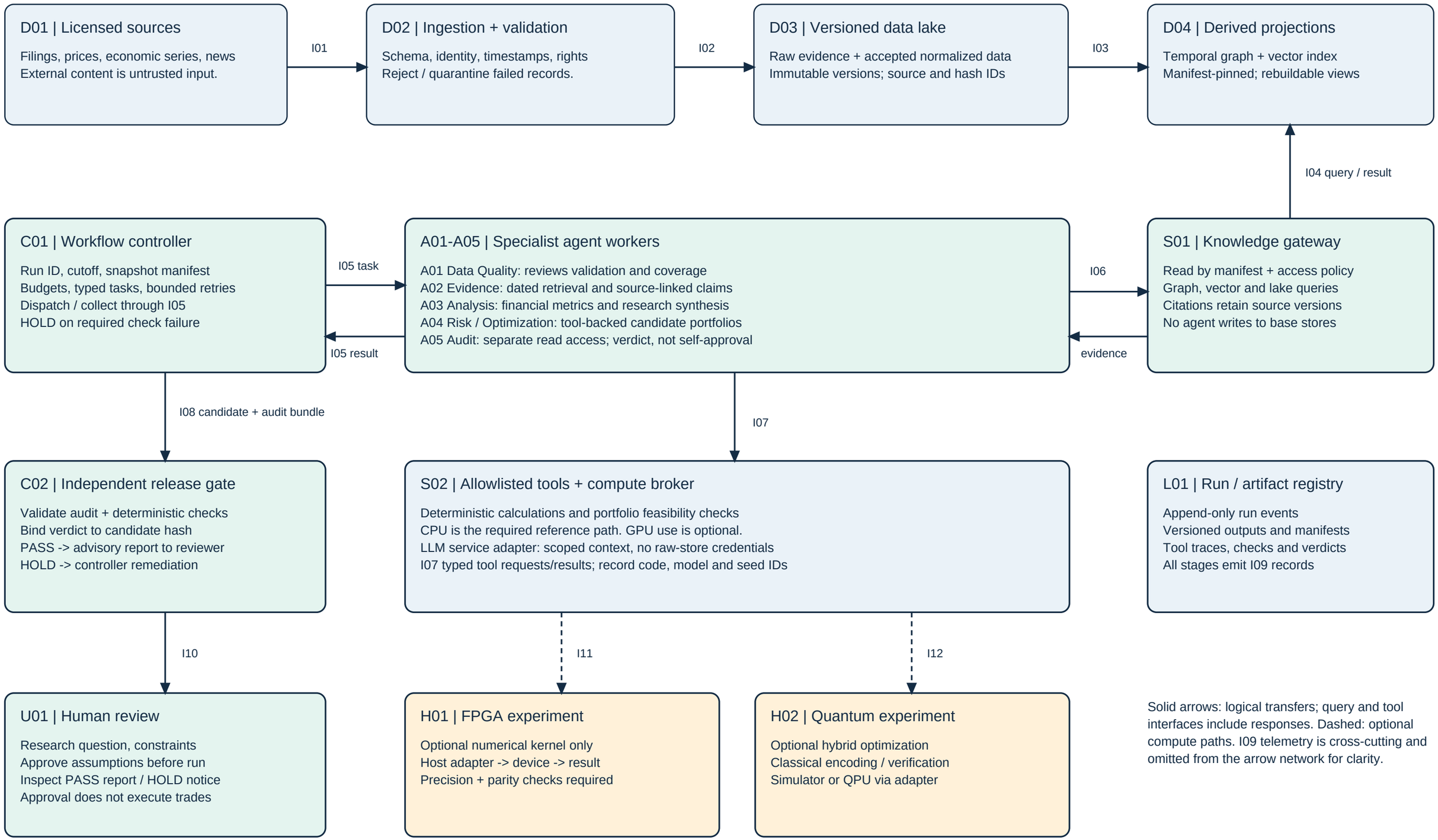
[8] NIST AI 600-1. Generative AI Profile.

[9] OWASP GenAI Security Project. Prompt injection.

[10] Qiskit Finance. Portfolio optimization tutorial.

FINTRACE | System architecture

LOGICAL ENGINEERING SCHEMATIC / REV B / 07 NOV 2026 / PROPOSED RESEARCH ARCHITECTURE



FINTRACE | Interface and authority contracts

LOGICAL ENGINEERING SCHEMATIC / REV B / 07 NOV 2026 / PROPOSED RESEARCH ARCHITECTURE

I01	D01 -> D02	Source payload, source ID, rights, publication/effective/ingestion times. Never treat source text as instructions.
I02	D02 -> D03	Accepted normalized records + raw references + validation results. Failed inputs remain quarantined and ineligible.
I03	D03 -> D04	Idempotent projection jobs keyed by dataset version, ontology version and index build. Publish manifest only after validation.
I04	S01 <-> D03/D04	Pinned manifest reads of lake evidence and derived views. In Sheet 1, D04 port represents this shared read interface.
I05	C01 <-> A01-A05	Task/result envelope: run_id, task_id, input hashes, cutoff, schema version, status, evidence refs, uncertainty and errors.
I06	Agents <-> S01	Scoped queries and dated evidence sets. Enforce entitlement, timestamp filters and source-version resolution.
I07	Agents <-> S02	Typed calculation / model requests and validated results. Deny arbitrary shell execution and unrestricted outbound tools.
I08	C01 -> C02	Candidate hash + complete audit bundle; A05 verdict obtained by C01. Gate rechecks deterministic release conditions.
I09	All stages -> L01	Run/task events, input/output hashes, manifests, software/model versions, seeds, timings, errors and gate decisions.
I10	U01 <-> C01/C02	Human request and assumptions enter C01; C02 returns released advisory report or HOLD. Input path omitted on Sheet 1.
I11	S02 <-> H01	Numeric arrays, units, precision, kernel/device version; outputs + error/performance metrics. CPU reference or explicit HOLD on failure.
I12	S02 <-> H02	Objective, encoding, constraints and solver settings; samples/candidates + costs. Classical feasibility validation required.

Authority boundary: Only ingestion/projection services publish accepted data and views. Agents may propose derived research artifacts through validated services. A05 cannot modify the candidate it audits; C02 cannot release a different candidate under an old verdict. Human review is downstream of technical release, and no component is authorized to send live orders.

FINTRACE | Run lifecycle and verification gates

LOGICAL ENGINEERING SCHEMATIC / REV B / 07 NOV 2026 / PROPOSED RESEARCH ARCHITECTURE

01 | CONFIGURE

Freeze question, universe, constraints, decision cutoff, availability policy, model/tool versions and release thresholds.

02 | PIN DATA

Select accepted data versions and matching graph/index manifest. Validate rights, freshness, entity mapping and completeness.

03 | RESEARCH

Controller dispatches scoped tasks. Claims resolve to source versions; unknowns and conflicts remain explicit.

04 | COMPUTE

Run deterministic reference calculations and solver. Optional hardware results must meet declared parity and feasibility limits.

05 | AUDIT

A05 independently reads candidate + evidence + traces. Deterministic checks and evidence review yield PASS or HOLD.

06 | RELEASE

C02 binds PASS to exact candidate hash and run manifest, then publishes an advisory report for human review.

R01 | HOLD / restart semantics

Missing evidence, failed checks, stale data, infeasible output or unavailable required service -> HOLD.

Preserve failure records; do not silently relax constraints. Correct inputs or configuration under a new version and rerun affected downstream tasks.

On restart, reconcile recorded tasks and external jobs before retrying. Use idempotency keys; never assume an interrupted task failed.

A changed candidate invalidates its prior audit and release verdict.

V01 | Required acceptance evidence

Temporal: no evidence later than permitted cutoff.
Data: explicit corporate-action and revision handling.
Evidence: material claims resolve to actual sources.
Math: reference tolerances and portfolio constraints pass.
Security: injected document instructions cannot grant tools.
Recovery: interrupted run resumes without duplicate effects.
Reproducibility: rerunnable artifacts; exact LLM replay is not assumed.
Hardware: end-to-end cost and accuracy versus classical baseline; no assumed quantum advantage.

Point-in-time rule: Use an explicit evaluation policy. Historical-availability studies require source publication/availability by cutoff and archived contemporaneous versions. Operational replay also requires ingestion by the simulated decision time. Effective dates alone do not prove information availability.

Design status: Proposed research architecture. See FINTRACE Project Plan Appendices C-E for the proposed asset universe, data policy, numerical tolerances, acceptance thresholds and optional compute criteria. Implementation and verification are planned. This original design is based on the FINTRACE project description. The referenced book provides background.